COMBINATORICA
Akadémiai Kiadó — Springer-Verlag

COMBINATORICA 9 (3) (1989) 289—313

CONSTRUCTING DISJOINT PATHS ON EXPANDER GRAPHS

D. PELEG[1] and E. UPFAL

Received March 10, 1987
Revised July 20, 1988

In a typical parallel or distributed computation model processors are connected by a spars interconnection network. To establish open-line communication between pairs of processors that wish to communicate interactively, a set of disjoint paths has to be constructed on the network. Since communication needs vary in time, paths have to be dynamically constructed and destroyed.

We study the complexity of constructing disjoint paths between given pairs of vertices on expander interconnection graphs. These graphs have been shown before to possess desirable properties for other communication tasks.

We present a sufficient condition for the existence of $K \leqq n^{\varrho}$ edge-disjoint paths connecting any set of K pairs of vertices on an expander graph, where n is the number of vertices and $\varrho < 1$ is some constant. We then show that the computational problem of constructing these paths lies in the classes Deterministic-$\mathcal{P}$ and Random-$\mathcal{NC}$.

Furthermore, we show that the set of paths can be constructed in probabilistic polylog time in the parallel-distributed model of computation, in which the n participating processors reside in the nodes of the communication graph and all communication is done through edges of the graph. Thus, the disjoint paths are constructed in the very computation model that uses them.

Finally, we show how to apply variants of our parallel algorithms to find sets of ***vertex-disjoint*** paths when certain conditions are satisfied.

1. Introduction

Information exchange between processors is an essential component of any parallel or distributed computation. In most applications, data transfer between individual processors has to be done through a relatively sparse communication network. Processors can communicate directly with only a few neighbors, and most communication has to traverse intermediate nodes. We model such a computation environment by a bounded-degree graph in which nodes represent processors and edges represent communication lines.

There are two basic methods for establishing communication in such models: packet routing and open-line communication. In packet routing, when processor x wishes to send a message to processor y, x generates a packet with the message and sends it to its destination. Many packets can traverse the network simultaneously, and their routes may overlap. In case of congestion, packets may have to wait in queues of intermediate processors. In open-line communication, on the other hand, a disjoint communication path is reserved for each pair of processors that wish to

[1] Supported in part by a Weizmann fellowship and by contract ONR N00014-85-C-0731.
AMS subject classification (1980): 68 R 10, 05 C 38

communicate. The edges of the path are dedicated to this purpose, and are released only when the communication terminates.

Packet routing is advantageous when transmitting a large volume of small size messages. Open-line communication on the other hand, is more efficient in handling long or variable length messages, and bidirectional on-line interactive communication.

In most instances of parallel and distributed computation, the communication pattern changes through the execution of the algorithm and cannot be predicted in advance. Furthermore, processors may communicate with different partners at different stages of the execution. Therefore, in a sparse network, one cannot reserve in advance a communication path for each pair of processors that might wish to communicate sometime during the execution of the algorithm. Instead, communication paths have to be constructed and destroyed dynamically according to the varied communication requests of the algorithm. This dynamic process of constructing disjoint paths on a sparse communication network motivates the work reported in this paper.

While the problem of packet routing has been extensively studied [18, 2, 17, 12] we are not aware of any theoretical study of the algorithmic aspects of open-line communication in bounded-degree networks. (For certain dense networks the problem was recently studied, cf. [16].) This lack may probably have to do with the proximity of this type of problem to NP-complete problems. The problem of deciding whether there are K edge- or vertex-disjoint paths in a given graph connecting K given pairs of vertices is known to be $\mathcal{NP}$-complete. (The case of fixed K was recently shown to be in $\mathcal{P}$ [14] for any K.) The negative results can be circumvented in our case. In most applications there is some control on the pattern of the communication graph, so the input graph on which one is required to find the set of paths is not arbitrary. More importantly, in these applications it is required to find a large number of disjoint paths, but not necessary the maximum number possible.

In this paper we concentrate our efforts on the important class of regular expander graphs. (A graph $G=(V,E)$ is an (α,β,γ)-expander if for every set of vertices U s.t. $|U|\leqq\beta|V|$, we have $|\Gamma(U)-U|\geqq\alpha|U|$, and for every set of vertices U s.t. $|U|\leqq|V|/2$, we have $|\Gamma(U)-U|\geqq\gamma|U|$, where $\Gamma(U)$ denotes the set of neighbors of nodes in U.) These graphs have been shown to possess some properties which make them especially useful for applications in communication networks [1, 4, 13]. For example, it has been shown in [13] that expansion is a necessary condition for the optimal solution of general packet routing problems, and that the probabilistic packet routing algorithm of Valiant [18] can be implemented efficiently on any expander network. The study of expander graphs enables us to obtain results for a large class of networks rather than for one particular communication pattern. Furthermore, the rapid progress in the explicit construction of strong, low-degree expanders [10] suggests that these graph are very likely to become a feasible pattern for future communication networks. Here we explore the implementation of open-line communication on expander graphs.

The major part of this paper is concerned with the problem of finding edge-disjoint paths, which is the usual requirement in communication applications. We first prove an existence result, giving sufficient conditions for the existence of K edge-disjoint paths connecting a_i to b_i for any set $\mathcal{AB}=\{(a_i,b_i)\}$ of $K\leqq n^\varrho$ pairs of vertices, where n is the number of vertices and $\varrho<1$ is some constant.

The existence proof involves special structures that consists of sets of edge-disjoint trees satisfying some special conditions. The existence of such structures in an expander graph implies the existence of a related set of disjoint paths. We demonstrate the existence of these structures by extending a construction technique developed in [5].

The existence proof does not yield directly efficient algorithms for constructing the paths, mainly because we cannot construct efficiently the tree-structure used in the proof. Instead, we show that the problem of constructing a set of disjoint paths on expander graphs, when such paths exist, can be reduced efficiently to a flow problem. Together with the existence result, we derive an $O(n^2K)$ algorithm for constructing K edge-disjoint paths for $K \leqq n^{\varrho}$.

We then turn to the question of parallel computation of the paths, beginning with the PRAM model. While the flow part of the sequential algorithm is in Random-$\mathcal{NC}$, the rest of the algorithm appears to be inherently sequential, since it uses the outcome of the flow phase to construct the paths one after the other. To overcome this problem we introduce a new technique, based on the properties of random walks on expander graphs. We use the algebraic characterization of expander graphs in terms of the eigenvalues of their adjacency matrices for the analysis of this phase of the algorithm. Thus, we obtain a probabilistic PRAM algorithm that uses $O((dn)^3)$ processors and runs in $O(\log^2 n)$ parallel steps.

The PRAM model is too powerful for our typical applications, and is used mainly as an intermediate step towards our final goal, which is to construct the disjoint paths within the very same model of computation that has to use them for the open-line communication. In this model, which we term the *parallel-distributed* model, there are only n processors that reside in the n nodes of the communication graph, and all the communication has to go through edges of the graph, one message per edge per step. It has been shown before that any n processor PRAM program can be simulated efficiently by n processors working in the parallel-distributed model [7]. Our main difficulty is to reduce the number of processors used by the PRAM algorithm. The $O((dn)^3)$ processors are needed in order to solve a flow problem on a graph of n nodes. We reduce the number of processors needed by restricting the flow problem to a subgraph of $n^{1/6}$ nodes. This gives us a probabilistic algorithm that runs in $O(\log^4 n)$ parallel steps on the parallel-distributed model.

The following theorem summarizes our contribution in the edge-disjoint case.

Theorem 1.1. *There is a constant ϱ depending only on α, β, γ and d, such that for any n-vertex d-regular (α, β, γ)-expander G with sufficiently large n and $\alpha \geqq 2$, and for any set $\mathcal{AB} = \{(a_i, b_i)\}$ of K disjoint pairs of vertices, if $K \leqq n^{\varrho}$ then*

1. *The graph G contains K edge-disjoint paths connecting a_i to b_i, $i = 1, \ldots, K$;*
2. *The K paths can be constructed in $O(Kn^2)$ steps;*
3. *The K paths can be constructed by a probabilistic PRAM algorithm that uses $O((dn)^3)$ processors and terminates in $O(\log^2 n)$ parallel steps;*
4. *The K paths can be constructed by a probabilistic algorithm on the parallel-distributed model in $O(\log^4 n)$ parallel steps.* ▮

The techniques developed for the existence proof of edge-disjoint paths and for the sequential and parallel algorithms, can be extended to the vertex-disjoint case. The number of disjoint paths that can be constructed in this case is naturally significantly smaller.

Theorem 1.2. 1. *For every integer $d>2$ and for every $\alpha>0$ there exist a d-regular (α, β)-expander and a set of $K=\frac{\alpha}{\alpha+1}\frac{d+1}{2}+1$ pairs of vertices that cannot be connected by vertex-disjoint paths.*

2. *For every d-regular graph there exists a set of $K=\frac{d}{2}+1$ pairs of vertices that cannot be connected by vertex-disjoint paths.*

3. *For every n-vertex d-regular (α, β, γ)-expander with sufficiently large n and $\alpha\geqq 2$, every set of $K\leqq\frac{\alpha-3}{2}$ pairs of vertices can be connected by vertex-disjoint paths. This set can be constructed by probabilistic algorithms, in $O(\log^2 n)$ parallel steps on an $O((dn)^3)$ processor PRAM and in $O(\log^4 n)$ parallel steps on the parallel-distributed model.* ∎

2. Preliminaries

The following notations and definitions are used throughout the paper.

Let $G=(V, E)$ be our interconnection graph. We denote the set of vertices occurring in a subgraph G' by $V(G')$ ($V(p)$ is used in a similar way w.r.t. a path p).

For every set of vertices $W\subseteqq V$, let $\Gamma(W)=\{u|(u, v)\in E \text{ and } v\in W\}$ and $\hat{\Gamma}(W)=\Gamma(W)-W$. Thus $\Gamma(W)$ denotes the set of neighbors of nodes in W and $\hat{\Gamma}(W)$ denotes the set of neighbors of W that are outside W.

Define $N(i, W)$, the *i-neighborhoods* of a set of vertices W, as follows: $N(0, W)=W$ and $N(i+1, W)=N(i, W)\cup\Gamma(N(i, W))$.

Fact 2.1. *In a d-regular graph $|N(i, W)|$ is bounded above by $d^{i+1}|W|$.*

Proof. $|N(i, W)|\leqq(1+d+d(d-1)+\ldots+d(d-1)^{i-1})|W|\leqq\left(1+d\frac{(d-1)^i-1}{d-2}\right)|W|\leqq$
$\leqq\left(\left(1-\frac{d}{d-2}\right)+d^{i+1}\right)|W|\leqq d^{i+1}|W|$. ∎

For two vertices u, w let $\text{dist}(u, w)$ denote the distance between them, i.e. the length of the shortest path connecting them. For two sets of vertices U, W, let $\text{dist}(U, W)=\min\{\text{dist}(u, w)|u\in U, w\in W\}$.

An *(α, β, γ)-expander* is a graph with the property that for every set U s.t. $|U|\leqq\beta n$, $|\hat{\Gamma}(U)|\geqq\alpha|U|$, and for every set U s.t. $|U|\leqq\frac{n}{2}$, $|\hat{\Gamma}(U)|\geqq\gamma|U|$.

We concentrate in this work on d-regular (α, β, γ)-expanders, with $\alpha\geqq 2$ and $0<\gamma<1$. Throughout, we define $l=\lfloor\alpha\rfloor$ as the *spanning factor* of the expander.

The input for our problem is given as a set $\mathcal{AB}=\{(a_i, b_i)\}$ of K disjoint pairs of vertices in G. Our goal is to find K disjoint paths connecting a_i to b_i, $i=1, \ldots, K$. Let $A=\bigcup_{i=1}^{K} a_i$ and $B=\bigcup_{i=1}^{K} b_i$. We refer to the set $A\cup B$ as the set of *end-points*.

All logarithms mentioned in the paper are to the base 2 unless explicitly noted otherwise.

3. Existence proof for the edge-disjoint case

A two phase strategy is used in the existence proofs and later in the construction algorithm. We first outline the strategy and the motivation for using it.

Suppose that we are given a set C of $2K$ vertices in G, all very far (e.g., $O(\log n)$) apart. If G is an expander, then by a counting argument we can prove that no matter how C is partitioned into pairs, these pairs can always be connected by vertex-disjoint paths. In order to use this result for an arbitrary set of pairs $\mathcal{AB}$, we introduce an initial phase in which we choose a set C of $2K$ vertices, $O(\log n)$ apart from each other, and then connect the vertices of the set $A\cup B$ to the vertices of the set C by a set of disjoint paths. In the second phase we connect the points in C according to the specification of $\mathcal{AB}$.

The vertices in $A\cup B$ may be very close to each other, and moreover, they may be arranged in an inconvenient and highly congested interconnection. The advantage of the two phase approach is that in the first phase, the paths from $A\cup B$ out do not have to obey any pairing requirements; they merely have to connect two sets of nodes according to some arbitrary matching, which is a considerably easier task. By the second phase, when the pairing specifications enter the picture, we start from points that are already far apart.

There are stil two problems that have to be addressed in order to utilize the above strategy. First, we have to prove that there are disjoint paths between the set $A\cup B$ and the set C. Secondly, we have to prove that even when the paths connecting $A\cup B$ to C are removed from the expander graph, the graph stil has the property that any pairing of the vertices in C can be connected by disjoint paths. To guarantee this we need to add some restrictions on the paths connecting $A\cup B$ vertices to C vertices. In particular, these paths must be short and a path to $c_i\in C$ must not use vertices from a certain vicinity of any other $c_j\in C$, $i\neq j$.

These requirements make the first phase harder than may seem at first glance. We cannot use a cut-set argument (Menger Theorem), since the paths constructed by this argument are not necessarily short. Instead, we prove the existence of a set of short paths by a two level construction of tree structures with special properties, centering around the nodes of $A>B$ as internal nodes. In what follows we describe these basic constructions and establish some preliminary lemmas leading to the existence of the first-phase paths (Theorem 3.10). The second phase is handled in Theorem 3.13.

We first show that for every set of vertices U of size at most βn there exists a special subgraph which we call *(U, l)-forest*. This subgraph is a collection of directed trees, in which the nodes of U are internal. We now give a more precise definition.

Definition 3.1. For any given set $U\subset V$ and $l\geqq 2$, a (U, l)**-forest** is a directed acyclic subgraph $G_F=(V_F, E_F)$ with the following properties:

(F1) $U\subseteq V_F\subseteq V$.

(F2) $|V_F|\leqq(l+1)|U|$.

(F3) $E_F\subseteq E$ (looking at the underlying undirected edges).

(F4) For every $v\in V_F$, *indegree* $(v)\leqq 1$.

(F5) For every $v\in U$, *outdegree* $(v)=l$.

Theorem 3.1. *Let $G=(V, E)$ be d-regular (α, β, γ)-expander s.t. $\alpha\geqq 2$, and let $l=\lfloor\alpha\rfloor$. Then for every set $U\subset V$ s.t. $|U|\leqq\beta n$ there exists a (U, l)-forest.*

Proof. Extending techniques that were introduced in [5], we give an iterative algorithm for constructing the (U, l)-forest $G_F=(V_F, E_F)$. We start with an empty G_F and a copy $\bar{G}$ of G, and repeatedly add edges to G_F while deleting parts of $\bar{G}$. Throughout the process we let $\bar{U}$ denote the collection of nodes from U whose outdegree in the forest is still less than l.

We use the following terminology. For every set $W \subseteq V$, define the assets of W as $A(W)=|\hat{\Gamma}(W)|$. (Recall that $|\hat{\Gamma}(W)|$ is the set of neighbors of W in G that are outside W.) In every moment during the construction and for every $v \in U$, define the *commitments* or liabilities of v, $C(v)$, as the number of children that v still needs to get. Clearly $0 \leq C(v) \leq l$. For every set $W \subseteq U$, let $C(W)=\sum_{v \in W} C(v)$. Define the *balance* of W as $B(W)=A(W)-C(W)$. A set $W \subseteq U$ is *good* if $B(W) \geq 0$, *critical* if $B(W)=0$ and *bankrupt* if $B(W)<0$. As the copy $\bar{G}$ of G changes along the process, the definitions of assets and balance are used with regard to the modified version $\bar{G}$. We say that the situation is *good w.r.t.* $\bar{U}$ if every set $W \subseteq \bar{U}$ is good.

The algorithm is based on repeated applications of a procedure *INSERT*(u, v). The input to this procedure consists of two vertices $u \in \bar{U}$ and $v \in V$ such that v's indegree in the forest is still 0 and $(u, v) \in E$. The procedure adds v as a child of u in the constructed forest, eliminates v from the graph $\bar{G}$ and updates the counter $C(u)$ and the set $\bar{U}$ accordingly. The entire iterative process is required to maintain the situation good w.r.t. the set $\bar{U}$.

Procedure *INSERT*(u, v).
1. $V_F \leftarrow V_F \cup \{u, v\}$.
2. $E_F \leftarrow E_F \cup \{(u, v)\}$.
3. Remove v and all its incident edges from $\bar{G}$.
4. $C(u) \leftarrow C(u)-1$.
5. If $C(u)=0$ then $\bar{U} \leftarrow \bar{U} \setminus \{u\}$.

The Construction Algorithm

(Children in the forest.) Initially $\bar{U}=U$, $\bar{G}=G$, $V_F=E_F=\emptyset$ and $C(v)=l$ for every $v \in U$.

While $\bar{U} \neq \emptyset$ **do:**
1. Find a vertex $v \in \hat{\Gamma}(\bar{U})$ in $\bar{G}$.
2. Find a vertex $u \in \bar{U}$ such that $(u, v) \in E$ and applying *INSERT*(u, v) maintains the situation good w.r.t. $\bar{U}$.
3. Apply *INSERT*(u, v).

The main step on the way to proving Thm. 3.1 is to show that the iterative process never gets stuck as long as $\bar{U} \neq \emptyset$.

Lemma 3.2. *In the beginning of the construction the situation is good w.r.t. U.*

Proof. Consider any set $W \subseteq U$. Then

$$A(W) = |\hat{\Gamma}(W)| \geq \alpha |W| \geq l|W| = C(W),$$

so W is good. ∎

Lemma 3.3. *If $\bar{U}$ is nonempty and the situation is good w.r.t. $\bar{U}$ then step* 1 *can be executed.*

Proof. We need to show that under the assumptions of the lemma, $\hat{\Gamma}(\bar{U})$ is nonempty. For every $u \in \bar{U}$, $C(u) > 0$. Therefore $C(\bar{U}) \geqq |\bar{U}| > 0$. Since the situation is good w.r.t. $\bar{U}$, $|\hat{\Gamma}(\bar{U}) = A(\bar{U}) \geqq C(\bar{U}) > 0$. ∎

Lemma 3.4. *If $\bar{U}$ is nonempty and the situation is good w.r.t. $\bar{U}$ then step* 2 *can be executed.*

Proof. Consider the situation after step 1, i.e., after choosing a node $v \in \hat{\Gamma}(\bar{U})$. The node v has neighbors in $\bar{U}$, and it was not inserted into the forest as a child in previous iterations (otherwise step 3 of procedure *INSERT* would remove v from $\hat{\Gamma}(\bar{U})$). We need to prove that under the assumptions of the lemma there is at least one neighbor $u \in \bar{U}$ of v such that the situation remains good w.r.t. $\bar{U}$ after applying $INSERT(u, v)$. Let $u_1, \ldots, u_k$, $k \geqq 1$, be v's neighbors in $\bar{U}$, through the edges $e_i = (u_i, v)$, for $1 \leqq i \leqq k$. We prove the claim by contradiction. Thus, assume that the choice of v as a child of any u_i will cause the situation to become bad, i.e., there are sets $W_1, \ldots, W_k \subseteqq \bar{U}$ such that choosing v as a child of u_i will make W_i bankrupt.

Claim 3.4.1. *For every $1 \leqq i \leqq k$, $u_i \notin W_i$.*

Proof. Assume that $u_i \in W_i$. Then by choosing the edge (u_i, v), both $A(W_i)$ and $C(W_i)$ are reduced by 1, so $B(W_i)$ does not change. Hence W_i has to be bankrupt already; a contradiction. ∎

Claim 3.4.2. *For every $1 \leqq i \leqq k$, $v \in \hat{\Gamma}(W_i)$.*

Proof. Assume that $v \notin \hat{\Gamma}(W_i)$. Then by choosing the edge (u_i, v), $A(W_i)$ stays the same, so $B(W_i)$ too does not change. Hence W_i has to be bankrupt already; a contradiction. ∎

At this point observe that if $k=1$ then we can already complete the proof of the lemma by deriving a contradiction. This follows immediately from the fact that (1) v has a single neighbor in $\bar{U}$, namely u_1, (2) v has a neighbor in W_1 (Claim 3.4.2), and (3) $u_1 \notin W_1$ (Claim 3.4.1). Therefore we carry on assuming that $k \geqq 2$.

Claim 3.4.3. *For every $1 \leqq i \leqq k$, W_i, is critical.*

Proof. After choosing the edge $e = (u_i, v)$, $B(W_i)$ becomes negative. But $C(W_i)$ does not change by choosing e, and $A(W_i)$ is reduced by at most 1. Hence $B(W_i)$ has to be 0. ∎

Claim 3.4.4. *For every sets A, B, C, D, if* (1) $A \cup B \subseteqq C \cup D$ *and* (2) $A \cap B \subseteqq C \cap D$, *then* $|A| + |B| \leqq |C| + |D|$.

Proof. $|A| + |B| = |A \cup B| + |A \cap B| \leqq |C \cup D| + |C \cap D| = |C| + |D|$. ∎

Claim 3.4.5. *For every $X, Y \subseteqq \bar{U}$, $B(X \cup Y) + B(X \cap Y) \leqq B(X) + B(Y)$.*

Proof. It is clear that $C(X \cup Y) + C(X \cap Y) \leqq C(X) + C(Y)$. We need to show that $|\hat{\Gamma}(X \cup Y)| + |\hat{\Gamma}(X \cap Y)| \leqq |\hat{\Gamma}(X)| + |\hat{\Gamma}(Y)|$. By the previous claim this reduces to showing that (1) $\hat{\Gamma}(X \cup Y) \cup \hat{\Gamma}(X \cap Y) \subseteqq \hat{\Gamma}(X) \cup \hat{\Gamma}(Y)$, and (2) $\hat{\Gamma}(X \cup Y) \cap$

$\cap \hat{\Gamma}(X \cap Y) \subseteq \hat{\Gamma}(X) \cap \hat{\Gamma}(Y)$. These can be shown by a straightforward case analysis. ▮

Corollary 3.4.6. *For every* $X_1, \ldots, X_k \subseteq \bar{U}$, $B(\bigcup_{i<k} X_i) \leqq \sum_{i \leqq k} B(X_i)$. ▮

Denote $W^* = \bigcup_{i \leqq k} W_i$. Let us temporarily remove v and the edges $e_1, \ldots, e_k$ from the graph $\bar{G}$ and re-compute the balances of our sets, denoting the new values by A', C' and B'. For every set W_i we have $A'(W_i) = A(W_i) - 1$ and $C'(W_i) = C(W_i)$, hence also $B'(W_i) = B(W_i) - 1$, and since W_i is critical, $B'(W_i) = -1$. By Cor. 3.4.6,

$$B'(W^*) = B'(\bigcup_{i \leqq k} W_i) \leqq \sum_{i \leqq k} B'(W_i) = -k.$$

Since $k \geqq 2$, we have $B'(W^*) \leqq -2$. On the other hand it is clear that $A'(W^*) = A(W^*) - 1$ and $C'(W^*) = C(W^*)$, hence also $B'(W^*) = B(W^*) - 1$, so $B(W^*) \leqq -1$. Hence W^* is bankrupt, contradicting the assumption that the situation is good w.r.t. $\bar{U}$.

This completes the proof of Lemma 3.4. ▮

The proof of Theorem 3.1 is now completed upon noting that when the process is terminated, i.e., $\bar{U} = \emptyset$, the resulting subgraph G_F indeed satisfies properties (F1) through (F5) in the definition of a (U, l)-forest. In particular the size of V_F is bounded by the fact that each $v \in U$ has exactly l children while each $v \in V_F \setminus U$ is a leaf. ▮

We now proceed, using the existence of (U, l)-forests, to prove the existence of more complex structures in the graph.

Definition 3.2. Let $W \subset V$, $W = \{w_1, \ldots, w_\eta\}$ and let $l \geqq 2$, $m \geqq 1$. A (W, l, m)-**structure** is a subgraph $G_S = (V_S, E_S)$ composed of a collection of η pairs $(T_i, p(w_i, r_i))$, one for each node $w_i \in W$, where T_i is a directed subtree rooted at a node r_i and $p(w_i, r_i)$ is a path connecting w_i to r_i. The collection satisfies the following properties:

(S1) Each T_i is a full directed subtree of degree l and depth m, hence it has l^m leaves.
(S2) The subtrees are vertex-disjoint from each other and from the paths.
(S3) The paths are edge-disjoint and their length is at most $\lceil \log_l \eta \rceil + 1$.
(S4) $V_S \subset N(m + \lceil \log_l \eta \rceil + 1, W)$ and $|V_S| \leqq d^{m + \lceil \log_l \eta \rceil + 2} \eta$.

(In fact, (S4) follows directly from the rest of the definition, but it is useful to state it explicitly.)

Theorem 3.5. *Let* $G = (V, E)$ *be a d-regular* (α, β, γ)*-expander, where* $\alpha \geqq 2$, *and let* $l = \lfloor \alpha \rfloor$. *For every set* $W \subset V$ *and for every* $m, q, \eta \geqq 1$ *such that* $|W| = \eta$, $q = m + \lceil \log_l \eta \rceil + 1$ *and* $d^q \eta \leqq \beta n$ *there exists a* (W, l, m)*-structure.*

Proof. Let $U = N(q-1, W)$. By Fact 2.1, $|U| \leqq d^q \eta \leqq \beta n$, and by Theorem 3.1 there exists a (U, l)-forest $G_F = V_F, E_F)$ of size $|V_F| \leqq (l+1) d^q \eta \leqq d^{q+1} \eta$. We construct the (W, l, m)-structure $G_S = (V_S, E_S)$ as a subgraph of the (U, l)-forest. For every node $v \in V_F$, denote by S_v the subtree rooted at v in the (U, l)-forest. We say that S_v is *clean* if it contains no node from W (in particular $v \notin W$). In case S_v is full up to depth m, we call it a *large* subtree, and denote by S_v^m the tree obtained by truncating S_v at depth m.

Let us now describe a process of associating a pair $(T_i, p(w_i, r_i))$ with each node $w_i \in W$. Each vertex v in V_F has to "suggest" a candidate pair (p, T) to its parent.

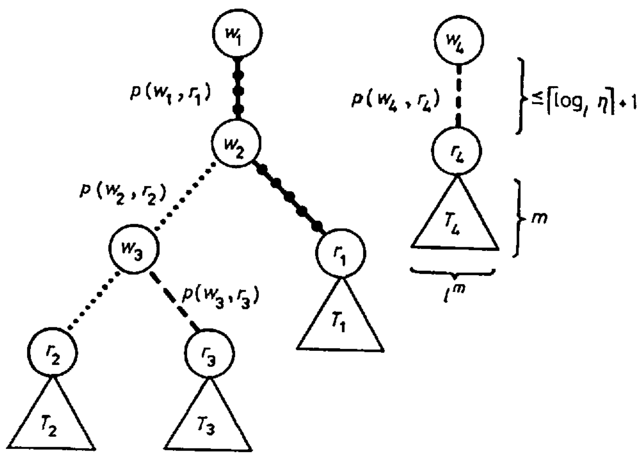

Fig. 1. A possible (W, l, m)-structure, $\eta = 4$

These suggestions are made according to the following rules. Assume v's parent is u, and let e be the edge connecting u to v.

1. If $v \notin W$ and S_v is clean and large, then v suggests (e, S_v^m).
2. If S_v is not clean and some (at least one) of v's children have made suggestions $(p_1, T_j'), \ldots, (p_j, T_j')$ to v, then v selects the pair (p_i, T_i') with the shortest path p_i, and suggests (p', T_i') to its parent, where p' is $e \cdot p_i$, the concatenation of e and p_i.

In addition, if $v \in W$ then v selects the pair with the second shortest path for itself.

We first need to guarantee that every node in W gets to select a pair.

Lemma 3.6. *Consider a node $v \in V_F$ such that S_v is not clean. The number of suggestions v gets from its children is at least one, and is l if $v \in W$.*

Proof. We prove the Lemma by structural induction on the (U, l)-forest G_F. That is, we first prove the claim for the leaves of G_F, and then we proceed to prove that if the claim holds for every vertex in $S_v - \{v\}$ for some $v \in V_F$ then it must also hold for v. Clearly, the claim holds vacuously for every $v \in V_F$ such that S_v is clean, and in particular for the leaves (since vertices of W cannot be leaves in G_F). The only interesting base case is that of a node $v \in V_F$ such that S_v is not clean but for every lower node $u \in S_v$, $u \neq v$, S_u is clean. This necessarily means that $v \in W$ but every lower node in S_v is not in W. Denote the children of v in G_F by $v_1, \ldots, v_l$, and let e_i be the edge connecting v to v_i. Then for each child v_i of v, $1 \leq i \leq l$, S_{v_i} is clean and large, since S_v is full up to depth at least q. Therefore each v_i suggests $(e_i, S_{v_i}^m)$ to v, so v gets l suggestions. Now consider the case of a node v for which there is a lower node $u \in S_v$, $u \neq v$, such that S_u is not clean. In particular this means that S_{v_i} is not clean for some child v_i of v, By the inductive hypothesis, v_i gets at least one suggestion from its children, so it passes on a suggestion to v. Finally, if $v \in W$, then every child v_i with an unclean subtree S_{v_i} gives v a suggestion by the inductive hypothesis, and for every child v_i with a clean subtree, this subtree is also large (for similar considerations as in the base case), so it too makes a suggestion. So again v gets l suggestions. ∎

As a result of this lemma it is clear that every node $w_i \in W$ gets to select a pair $(T_i, p(w_i, r_i))$ for itself. We next have to verify that the paths in selected pairs are not too long.

Lemma 3.7. *For every node $w_i \in W$, the pair $(T_i, p(w_i, r_i))$ selected by w_i satisfies $|p(w_i, r_i)| \leqq \lceil \log_l \eta \rceil + 1$.*

Proof. S_{w_i} is full up to depth $q = m + \lceil \log_l \eta \rceil + 1$. Consider the set of vertices of the $(\lceil \log_l \eta \rceil + 1)$st level of this tree. There are at least η such vertices in every subtree S_{v_j}, $1 \leqq j \leqq l$, where v_1 to v_l are the children of w_i. Therefore in every subtree S_{v_j} there is at least one node z_j at the $(\lceil \log_l \eta \rceil)$th level such that S_{z_j} is clean. Also S_{z_j} is large. Let p_{z_j} denote the path from v to z_j. Clearly, either (p_{z_j}, S_{z_j}) reaches v as a suggestion from v_j or v gets from v_j a suggestion with a shorter path. In either case, the suggestions v gets from each v_j include a path of length at most $\lceil \log_l \eta \rceil + 1$. Therefore the second shortest suggestion (which gets to be selected) satisfies the requirement of the lemma. ∎

This completes the proof of Theorem 3.5, as the rest of the conditions in the definition of an (W, l, m)-structure are easy to verify. In particular, the various disjointness requirements are taken care of by the selection process of the pairs. ∎

We will also need the following technical lemma.

Lemma 3.8. *Let $G = (V, E)$ be a d-regular (α, β, γ)-expander, let $u, v \in V$, $X, Y \subset V$ and let μ_1, μ_2 be integers such that $1 \leqq \mu_1 < \mu_2 \leqq \lfloor \log_d n \rfloor - 1$, $\mathrm{dist}(\{u, v\}, X) > \mu_1$, $\mathrm{dist}(\{u, v\}, Y) > \mu_2$, $|X| < \lfloor \gamma(\gamma+1)^{\mu_1} \rfloor$ and $|Y| < \gamma(\gamma+1)^{\mu_2 - \mu_1}$. Then there exists a path of length at most $\tau = 2\left\lceil \frac{\log n}{\log(\gamma+1)} + \mu_2 \right\rceil$ connecting u and v and not going through X or Y.*

Proof. Let G' be the graph obtained by removing from G the vertices of $X \cup Y$ along with their edges. We now obtain some lower bounds on the size of i-neighborhoods of u in G', denoted $N'(i, u)$.

Claim 3.8.1. *For $1 \leqq i \leqq \mu_1$, $|N'(i, u)| \geqq (\gamma+1)^i$.*

Proof. Note that for $i \leqq \mu_1$, $|N'(i, u)| = |N(i, u)|$. The claim is proved by induction on i. The case of $i = 1$ is immediate since $\gamma < d$. Now assume the claim for $i < \mu_1$ and consider $i+1$. Since $i < \mu_1 \leqq \lfloor \log_d n \rfloor - 1$, by Fact 2.1 $|N(i, u) \leqq n/2$, hence the expansion property applies to $N(i, u)$ and we get the inequality

$$|N(i+1, u)| = |N(i, u)| + \left|\Gamma(N(i, u))\right| \geqq (\gamma+1)|N(i, u)|.$$

This inequality, combined with the inductive hypothesis, yields the desired claim. ∎

In particular, letting $s_1 = |N'(\mu_1, u)|$, we get $s_1 \geqq (\gamma+1)^{\mu_1}$, and by the assumption of the lemma , $\lfloor \gamma s_1 \rfloor > |X|$.

Claim 3.8.2. *For $\mu_1 \leqq i \leqq \mu_2$, $|N'(i, u)| \geqq s_1 - 1 + (\gamma+1)^{i - \mu_1}$.*

Proof. By induction on i. The case of $i=\mu_1$ is immediate, and the general case $(i+1)$ is again based on fact that when $i<\mu_2\leqq\lfloor\log_d n\rfloor-1$, the expansion property applies to $N(i,u)$ and

$$|N'(i+1,u)| \geqq |N'(i,u)|+\left|\Gamma(N'(i,u))\right|-|X| \geqq (\gamma+1)|N'(i,u)|-|X|.$$

This inequality, coupled with the inductive hypothesis, yields

$$\begin{aligned}|N'(i+1,u)| &\geqq (\gamma+1)\left(s_1-1+(\gamma+1)^{i-\mu_1}\right)-|X|\\ &= s_1-1+(\gamma+1)^{i+1-\mu_1}+(s_1-1)\gamma-|X|\\ &\geqq s_1-1+(\gamma+1)^{i+1-\mu_1},\end{aligned}$$

since $(s_1-1)\gamma\geqq|X|$. ∎

In particular, letting $s_2=|N'(\mu_2,u)|$, we get $s_2\geqq s_1-1+(\gamma+1)^{\mu_2-\mu_1}$, and by the assumption of the lemma, $s_2>\frac{|X\cup Y|}{\gamma}$.

Claim 3.8.3. *For* $i\geqq\mu_2$, $|N'(i,u)|\geqq\min\{\lfloor n/2\rfloor+1,\ s_2-1+(\gamma+1)^{i-\mu_2}\}$.

Proof. Again by induction on i. For the general case $(i+1)$ observe that if $|N'(i,u)|\geqq\geqq\lfloor n/2\rfloor+1$ then so is $|N'(i+1,u)|$, and otherwise the expansion property applies and we get

$$|N'(i+1,u)| \geqq |N'(i,u)|+\left|\Gamma(N'(i,u))\right|-|X\cup Y| \geqq (\gamma+1)|N'(i,u)|-|X\cup Y|$$

and proceed as in the previous claim using the fact that $(s_2-1)\gamma\geqq|X\cup Y|$. ∎

Overall we get that for every i,

$$\begin{aligned}|N'(i,u)| &\geqq \min\{\lfloor n/2\rfloor+1,\ (\gamma+1)^{\mu_1}+(\gamma+1)^{\mu_2+\mu_1}+(\gamma+1)^{i-\mu_2}-2\}\\ &\geqq \min\{\lfloor n/2\rfloor+1,\ (\gamma+1)^{i-\mu_2}\}.\end{aligned}$$

A similar result holds for v. It is therefore clear that for $i_0=\left\lceil\frac{\log n}{\log(\gamma+1)}+\mu_2\right\rceil$, both $|N'(i_0,u)|\geqq\lfloor n/2\rfloor+1$ and $|N'(i_0,v)|\geqq\lfloor n/2\rfloor+1$. Therefore they have a common vertex w in distance at most i_0 from each of them in G', hence the shortest path connecting them is of length at most τ and satisfies the requirements of the lemma. ∎

Let us now define some parameters to be used throughout the rest of the paper and mention some of the relations they satisfy. The motivation for these definitions will become clear along the proofs. It is important however to notice that the relations between these constants are fixed simultaneously for all the proofs that follow.

Our constructions depend on a constant $0<\varphi<1$. Essentially the set U for which we build a (U,l)-forest has to be kept of size at most n^{φ}. For the existence proof and most of our algorithms we may use any $\varphi<1$, but for our distributed algorithm (Section 4.3) we need to fix φ to a smaller value.

We fix the following set of parameters.

$$0 \leqq \varphi \leqq 1,$$
$$l = \lfloor \alpha \rfloor,$$
$$\theta = \log_l d,$$
$$\delta = \frac{\varphi}{2 \log d},$$
$$\psi_1 = \frac{\delta \log (1+\gamma)}{6 \log^2 d},$$
$$\psi_2 = \frac{\delta}{3 \log d},$$
$$\mu_1 = \lfloor \psi_1 \log n \rfloor,$$
$$\mu_2 = \mu_1 + \lfloor \psi_2 \log n \rfloor.$$

Lemma 3.9. *There exists a constant* $\varrho = \varrho(\alpha, \beta, \gamma, \varphi, d)$, $0 < \varrho < 1$, *that simultaneously satisfies the following five requirements:*

(R1) $\varrho < \dfrac{\varphi - \delta\theta}{\theta + 1}$,

(R2) $\varrho < \psi_1 \log (1+\gamma)$,

(R3) $\varrho + \psi_1 \log d < \psi_2 \log (1+\gamma)$,

(R4) $\varrho + \psi_1 \log d < \dfrac{\delta}{3}$,

(R5) $\varrho + (2\psi_1 + \psi_2) \log d < \dfrac{2\delta}{3}$.

Proof. We need to verify that $\varphi - \delta\theta$, $\psi_2 \log (1+\gamma) - \psi_1 \log d$, $\dfrac{\delta}{3} - \psi_1 \log d$ and $\dfrac{2\delta}{3} - (2\psi_1 + \psi_2) \log d$ are all positive, which follows immediately from the definitions and the fact that $1+\gamma < d$. ∎

We are now ready to formulate the requirements on the set of paths generated in the first phase of our strategy.

Definition 3.3. Let $A \cup B$, C be two sets of $2K$ vertices each in G, such that $K \leqq n^\varrho$ for some constant ϱ satisfying requirements (R1) through (R5) of Lemma 3.9. Set

$$m = 3 \left\lceil \frac{\delta}{3} \log_l n \right\rceil,$$
$$\tau_1 = \lceil \log_l (2K) \rceil + m + 1,$$
$$\tau_2 = 2 \left\lceil \frac{\log n}{\log (1+\gamma)} + \mu_2 \right\rceil.$$

We say that C is *well-distributed* with respect to G and $A\cup B$ if

(WD1) $\text{dist}(A\cup B, C) > \tau_1+\mu_1+\mu_2$.

(WD2) For every $c_i, c_j \in C$, $\text{dist}(c_i, c_j) > \mu_1+\mu_2$.

A set of edge-disjoint paths connecting $A\cup B$ to C is said to be *well-distributed* with respect to $A\cup B$ and C, if

(WD3) The sum of the lengths of the paths is bounded by $2K(\tau_1+\tau_2)$.

(WD4) For every $c_i, c_j \in C$ the path connected to c_i does not use any of the vertices in $N(\mu_1, c_j)$.

Theorem 3. 10. *For any n-vertex d-regular (α, β, γ)-expander G with sufficiently large n and $\alpha \geqq 2$, and for any two sets of vertices $A\cup B$ and C such that $|A\cup B|=|C|=2K\leqq 2n^{\varrho}$ (for some constant ϱ as in Lemma 3.9) and C is well-distributed w.r.t. $A\cup B$, there exists a set of well-distributed edge-disjoint paths connecting $A\cup B$ to C.*

Proof. Let $A\cup B=\{v_1, \ldots, v_{2K}\}$ and $C=\{c_1, \ldots, c_{2K}\}$, and assume that C is well-distributed w.r.t. $A\cup B$. Note that by requirement (R1) of Lemma 3.9, $d^{\tau_1}(2K)\leqq \beta n$ for sufficiently large n. Therefore by Theorem 3.5 there exists an $(A\cup B, l, m)$-structure $G_S=(V_S, E_S)$. By property (S4) of Definition 3.2 the size of the structure satisfies $|V_S|\leqq d^{\tau_1+1}(2K)$, which by requirement (R1) of Lemma 3.9 again, is bounded above by n^{φ} for sufficiently large n.

Next we find a set of "scattered" vertices $D=\{d_1, \ldots, d_{2K}\}$. Each vertex $d_i\in D$ is a leaf in a distinct subtree T_i in the structure G_S. The $(A\cup B, l, m)$-structure immediately induces a set P of edge-disjoint paths $p(v_i, d_i)$ connecting each $v_i \in A\cup B$ with the corresponding leaf d_i. Each such path $p(v_i, d_i)$ is obtained by concatenating $p(v_i, r_i)$ with the unique path $p(r_i, d_i)$ connecting the root r_i to d_i in T_i. Note that by properties (S1) and (S3) of the $(A\cup B, l, m)$-structure (Definition 3.2), each of these paths is of length at most τ_1. In addition, the set of vertices of D and the associated set of paths P are required to satisfy the following properties:

(D1) For every two nodes $d_i, d_j \in D$ s.t. $i\neq j$, $\text{dist}(d_i, d_j) > \mu_1+\mu_2$.

(D2) For every two nodes $d_i, d_j \in D$ s.t. $i\neq j$, $\text{dist}(d_j, V(p(v_i, d_i))) > \mu_1$.

(Recall that $V(p)$ denotes the set of vertices in the path p.)

Lemma 3.11. *There exist such sets D and P.*

Proof. We construct these sets inductively by choosing the vertices $d_i\in D$ one by one, and fixing the paths $p(r_i, d_i)$ leading to them in a way consistent with the previous choices. (The path $p(v_i, d_i)$ is then obtained by concatenating the segment $p(v_i, r_i)$ supplied by the structure G_S with the chosen segment.)

Suppose we already chose $i-1$ vertices and paths and placed them in the sets D and P respectively. We choose the new vertex d_i and the path leading to it, $p(r_i, d_i)$ in such a way that d_i satisfies

(1) $\text{dist}(d_i, D) > \mu_1+\mu_2$,

(2) For every $j>i$, $\text{dist}(d_i, V(p(v_j, r_j))) > \mu_1$,

(3) For every $j<i$, $\text{dist}(d_i, V(p(v_j, d_j))) > \mu_1$,

and that $p(r_i, d_i)$ satisfies

(4) $\operatorname{dist}(D, V(p(r_i, d_i))) > \mu_1$.

It is easy to verify that if each of our choices satisfies requirements (1) through (4) then the final sets satisfy the desired properties (D1) and (D2). However, in order to make the induction go through we need to add one more requirement, namely,

(5) For every $j > i$, $\operatorname{dist}(d_i, V(T_j^{m/3})) > \mu_1$.

(Recall that T^i denotes the tree T truncated at depth i.)
Define the following sets:

$$X_1' = N(\mu_1+\mu_2, D),$$

$$X_2' = \bigcup_{j>i} V(p(v_j, r_j)),$$

$$X_3' = \bigcup_{j<i} V(p(v_j, d_j)).$$

$$X_4 = N(\mu_1, D),$$

$$X_5' = \bigcup_{j>i} V(T_j^{m/3}),$$

$$X_{235} = N(\mu_1, X_2' \cup X_3' \cup X_5').$$

D contains $(i-1)$ vertices, so by Fact 2.1

$$|X_4| \leqq 2Kd^{\mu_1+1} \leqq 2dn^{\varrho+\psi_1 \log d}$$

and

$$|X_1| \leqq 2Kd^{\mu_1+\mu_2+1} \leqq 2dn^{\varrho+(2\psi_1+\psi_2)\log d}.$$

Similarly, $X_2' \cup X_3' \leqq 2K\tau_1 \leqq 2\tau_1 n^{\varrho}$ and $|X_5'| \leqq 2Kl^{m/3+1} \leqq 2ln^{\varrho+\delta/3}$, so

$$|X_{235}| \leqq 2d\tau_1 n^{\varrho+\psi_1 \log d} + 2dln^{\varrho+\delta/3+\psi_1 \log d}.$$

Eliminate the vertices of $X_1 \cup X_{235}$ and all their adjacent edges from the graph for the rest of step i. Consider the tree T_i. This tree has $l^{m/3} = n^{\delta/3}$ nodes at depth $m/3$. By requirement (R4) of Lemma 3.9 this number is larger than $|X_4|$ for sufficiently large n. Therefore there must be some vertex u at depth $m/3$ in T_i such that the subtree of T_i rooted at u, S_u, contains no vertex of X_4. A simple counting argument shows that even after erasing the vertices of $X_1 \cup X_{235}$ there is still at least one leaf in S_v which is not erased. This is true because S_v has $l^{2m/3} = n^{2\delta/3}$ leaves whereas by the fact that $\tau_1 \in O(\log n)$ and by requirements (R4) and (R5) of Lemma 3.9, $|X_1 \cup X_{235}| < n^{2\delta/3}$ for sufficiently large n. We now claim that this leaf and the path connecting it with r_i satisfy properties (1) to (5) and therefore can be chosen as d_i and $p(r_i, d_i)$. Clearly, choosing the leaf after the elimination of $X_1 \cup X_{235}$ guarantees that it satisfies properties (1), (2), (3) and (5). It remains to show that the newly constructed path $p(r_i, d_i)$ satisfies property (4). This path is combined of two segments, $p(r_i, u)$ and $p(u, d_i)$. For any $d_j \in D$, $j > i$ we have $\operatorname{dist}(d_j, V(p(r_i, u))) > \mu_1$ by property (5) in the inductive hypothesis on d_j, and $\operatorname{dist}(d_j, V(p(u, d_i))) > \mu_1$ since S_u contains no vertices of X_4. ∎

Once the vertices in $A \cup B$ are connected to vertices in D we can use Lemma 3.8 iteratively to prove the existence of paths connecting the vertices in D to vertices in C. This is done as follows:

Lemma 3.12. *There exist $2K$ vertex-disjoint paths $p(d_i, c_i)$, $1 \leqq i \leqq 2K$, each of length at most τ_2, with the property that for every i, $p(d_i, c_i)$ connects $d_i \in D$ to $c_i \in C$ and goes through no vertices from the μ_1-neighborhood of any vertex of $D \cup C$ (except its endpoints d_i and c_i) and through no vertices from the paths $p(v_j, d_j)$, $1 \leqq j \leqq 2K$, $j \neq i$.*

Proof. We show that it is possible to choose the paths one by one. Assume we already chose $i-1$ paths $p(d_j, c_j)$, $1 \leqq j \leqq i-1 < 2K$, and now we have to connect d_i and c_i. Let

$$X_1 = \bigcup_{j=1}^{i-1} V(p(d_j, c_j)), \quad X_2 = \bigcup_{1 \leqq j \leqq 2K, j \neq i} V(p(v_j, d_j)), \quad X = X_1 \cup X_2$$

and

$$Y = N(\mu_1, D \cup C + \{d_i, c_i\}).$$

D and X_2 are contained in the structure G_S which by property (S4) of Definition 3.2 is contained in $N(\tau_1, A \cup B)$, and since C is well-distributed w.r.t. $A \cup B$, by property (WD1) of definition 3.3 dist $(C, D) > \mu_1 + \mu_2$ and dist $(C, X_2) > \mu_1 + \mu_2$. Combined with properties (D1) and (D2) of D, property (WD2) of Definition 3.3 and the inductive hypothesis we get that dist $(\{d_i, c_i\}, X) > \mu_1$ and dist $(\{d_i, c_i\}, Y) > \mu_2$. Also, the length of paths $p(v_i, d_i)$ is bounded by τ_1, and by the inductive hypothesis the paths $p(d_j, c_j)$ constructed so far are of length at most τ_2, so

$$|X| \leqq (2K-1)\tau_1 + (i-1)\tau_2 < 2(\tau_1 + \tau_2) n^{\varrho}.$$

Both $\tau_1, \tau_2 \in O(\log n)$, so by requirement (R2) of Lemma 3.9, for sufficiently large n

$$|X| < \left[\frac{\gamma}{1+\gamma} n^{\psi_1 \log(1+\gamma)}\right] \leqq [\gamma(1+\gamma)^{\mu_1}].$$

Finally, using Fact 2.1 we get

$$|Y| \leqq (2K-2) d^{\mu_1+1} \leqq 2dn^{\varrho + \psi_1 \log d},$$

so by requirement (R3) of Lemma 3.9, for sufficiently large n

$$|Y| < \frac{\gamma}{1+\gamma} n^{\psi_2 \log(1+\gamma)} \leqq \gamma(1+\gamma)^{\mu_2 - \mu_1}.$$

Hence all the premises of Lemma 3.8 are satisfied, and we now deduce the existence of a path of length at most τ_2 connecting d_i and c_i and going through no vertices of $X \cup Y$, as required. ▮

Combining the paths $p(v_i, d_i)$ and $p(d_i, c_i)$, $1 \leqq i \leqq 2K$ asserted in Lemmas 3.11 and 3.12, we get a set of well-distributed edge-disjoint paths $p(v_i, c_i)$, $1 \leqq i \leqq 2K$ connecting $A \cup B$ to C. This, at last, completes the proof of Theorem 3.10. ▮

For the second phase of our strategy, namely, connecting the nodes of C according to the pairing requirements of $\mathscr{AB}$, we need

Theorem 3.13. *Let G be an n-vertex d-regular (α, β, γ)-expander with sufficiently large n and $\alpha \geqq 2$, and let $A \cup B$ and C be two sets of vertices such that $|A \cup B| = |C| = 2K \leqq 2n^{\varrho}$ and C is well-distributed with respect to $A \cup B$. Let P be a set of edge-disjoint, well-distributed paths connecting the vertices in $A \cup B$ to the vertices in C. Then for any partition of C into pairs there are D vertex-disjoint paths connecting these pairs such that the path connecting c_i to c_j uses no vertices of paths from P except $p(v_i, c_i)$ and $p(v_j, c_j)$.*

Proof. Assuming a partition of C into k pairs, we construct the paths connecting the pairs one by one. The choice of each path is done using Lemma 3.8 with appropriately defined sets X and Y, in a way similar to Lemma 3.12. Details are omitted. ▮

Combining the two phases together, using Theorems 3.10 and 3.13, we have

Theorem 3.14. *Let $G=(V, E)$ be an n-vertex d-regular (α, β, γ)-expander with sufficiently large n and $\alpha \geqq 2$. There exists a constant $\varrho = \varrho(\alpha, \beta, \gamma, \varphi, d)$ s.t. for every set of vertex-disjoint pairs $\mathscr{A}\mathscr{B} = \{(a_i, b_i) | a_i, b_i \in V,\ 1 \leqq i \leqq K\}$, $K \leqq n^{\varrho}$, there exist K edge-disjoint paths in G connecting the pairs.* ▮

Note. The set C is actually not needed for the existence proof itself. In fact, the proof can be simplified as follows. Once we prove the existence of the set D of scattered leaves in the $(A \cup B, l, m)$-structure G_S, we can use the nodes of D themselves as starting points for the second phase. Thus, we can identify pairs $d_i', d_i'' \in D$ as the mates of pairs $(a_i, b_i) \in \mathscr{A}\mathscr{B}$, and these pairs in D can be shown to have connecting paths by an argument as in Lemma 3.12. However, the more complex proof procedure has to be followed in the algorithms discussed in the next section. The reason is that we cannot compute the structure G_S efficiently, and without it we cannot find a set D with the required properties, whereas a set C with the right properties can be found without knowing the structure G_S, simply by choosing its elements far enough from $A \cup B$.

4. Algorithms for the edge-disjoint case

In this section we give several algorithms in various levels for constructing edge-disjoint paths. In Section 4.1 we derive a polynomial time algorithm based on the existence proof of the last section, and in Sections 4.2 and 4.3 we describe an $\mathscr{R}\mathscr{N}\mathscr{C}$ algorithm and a distributed algorithm, respectively.

4.1. Polynomial time algorithm

The existence proof given in the previous section does not translate directly to an efficient sequential algorithm. In particular we know no way to identify an $(A \cup B, l, m)$-structure in polynomial time, thus we cannot construct directly a set of well-distributed paths connecting the set $A \cup B$ to a set C of well-distributed vertices.

The algorithmic solution is based on the following idea. While we cannot actually construct the $(A \cup B, l, m)$-structure, we can identify its boundaries by marking a τ_1-neighborhood around the set $A \cup B$. We then choose the vertices of C

deterministically as some arbitrary nodes outside $N(\tau_1+\mu_1+\mu_2+1, A\cup B)$ satisfying the distance requirements. Then we rely on the fact that a set of well-distributed paths connecting $A\cup B$ to C exist, and compute such paths by defining a min-cost max-flow problem on a modified graph $\tilde{G}$ with the following property: the paths employed by the solution to the flow problem in $\tilde{G}$ correspond to well-distributed paths in G, whenever such paths exist.

Note that because the τ_1-neighborhood of $A\cup B$ has to be of size at most n^φ, we have the constraint that $d^{\tau_1+\mu_1+\mu_2+1}K\leqq n^\varphi$. This forces us to strengthen requirement (R1) made on ϱ in Lemma 3.9 and replace it with

$$\text{(R1')}\qquad 0<\varrho<\frac{\varphi-\delta\theta-(\psi_1+\psi_2)\log d}{\theta+1}.$$

This new requirement can be met by some ϱ values as long as $0<\delta\theta+ +(\psi_1+\psi_2)\log d<\varphi$, which indeed holds for the values of δ, ψ_1 and ψ_2 chosen earlier.

Algorithm 1

Step 1: Construct the graph $\tilde{G}$ from G as follows:

1.1. Add a source vertex s and a target vertex t.

1.2. Connect s by a directed edge to each vertex in $A\cup B$.

1.3. For each vertex $v\in C$ do the following. Replace $N(\mu_1, v)$ by two vertices v_1, v_2, connect every vertex in $\hat{\Gamma}(N(\mu_1, v))$ by a directed edge to v_1, connect v_2 by a directed edge to every vertex in $\hat{\Gamma}(N(\mu_1, v))$ and add a directed edge from v_1 to v_2 and from v_2 to t.

1.4. Give each edge in $\tilde{G}$ capacity 1 and cost 1.

Step 2: Apply a min-cost max-flow algorithm on $\tilde{G}$ to find precisely $2K$ edge-disjoint paths with minimum total length connecting s to t. Translate these paths back to G, to form a well-distributed set of paths connecting $A\cup B$ to C.

Step 3: For $i=1$ to K do:

Let $C(a_i)$ denote the vertex in C connected to $a_i\in A\cup B$ by the flow algorithm.

3.1. Remove from the graph all vertices belonging to existing paths and to the μ_1-neighborhoods of $C(a_j)$ and $C(b_j)$, $j\neq i$, along with their edges.

3.2. Mark a shortest path connecting $C(a_i)$ to $C(b_i)$.

Theorem 3.13 guarantees the existence of K paths connecting the pairs of vertices in C. Thus we prove:

Theorem 4.1. *There exists a constant $\varrho=\varrho(\alpha, \beta, \gamma, \varphi, d)$ s.t. for any n-vertex d-regular (α, β, γ)-expander with sufficiently large n and $\alpha\geqq 2$ and for any set of $K\leqq n^\varrho$ pairs, the pairs can be connected by K edge- disjoint paths in $O(Kn^2)$ steps.* ∎

4.2. Random-$\mathcal{NC}$ algorithm

The main-cost max-flow problem phase of the sequential algorithm is reducible to a problem of weighted matching with bounded weights in a dn-vertex graph [9, p. 187]. Thus, the flow phase is in Random-$\mathcal{NC}$. The second phase in which nodes of C are connected according to the input requirements appears to be inherently sequential, and we replace it by a probabilistic phase based on random walks.

Algorithm 2

Step 1: Choose a set C of $2K$ random vertices (each $2K$-subset of V is chosen with equal probability).

Step 2: As in the sequential algorithm use a min-cost max-flow parallel algorithm to construct $2K$ disjoint paths of minimum length connecting the vertices in $A \cup B$ to the vertices in C.

Step 3: For each vertex $C(a_i)$, $i=1, ..., k$, mark $K^2 \log^3 n$ random walks of length $\xi \log n$ starting at $C(a_i)$. (The constant ξ is fixed in the proof.) Starting at each vertex $C(b_i)$ mark one random walk of length $\xi \log n$.

Step 4: Let $D(b_i)$ denote the end-point of the random walk starting at $C(b_i)$. Let $\mathcal{D}(a_i)$ denote the set of end-points of the random walks starting at $C(a_i)$. Choose a vertex $D(a_i)$ in $\mathcal{D}(a_i)$ with a shortest path to $D(b_i)$, and connect $D(a_i)$ to $D(b_i)$ by a shortest path.

Lemma 4.2. *If $K \leq n^{\varrho'}$ then the minimum distance between any vertex $x \in A \cup B \cup C$ and any vertex $y \in C$, $x \neq y$, is $\mu_3 = (1-2\varrho'-\varepsilon)\log_d n$ with probability $1-8d\frac{1}{n^\varepsilon}$.*

Proof. The probability that a particular pair (x, y) violates the requirement of the lemma is bounded above (by Fact 2.1) by $\frac{d^{\mu_3+1}}{n} = dn^{-2\varrho'-\varepsilon}$. Hence over all $8K^2$ pairs the probability of violation is bounded by $8dn^{-\varepsilon}$. ∎

Lemma 4.3. *Using the probabilistic parallel weighted matching algorithm of* [11], *Step 2 uses $O(\log^2 n)$ time and $O((dn)^3)$ processors and with high probability finds a set of well-distributed edge-disjoint paths connecting each vertex in $A \cup B$ to a distinct vertices in C.* ∎

The core of this section is the analysis of steps 3 and 4. The proof is based on the algebraic characterization of expander graphs and its implication to the analysis of random walks on expanders.

The following lemma is a simple corollary of Theorem 2.5 in [3]:

Lemma 4.4. *If G is a d-regular (α, β, γ)-expander, $A(G)$ the adjacency matrix of G' and λ the second largest eigenvalue in absolute value of $A(G)$, then $\frac{\lambda}{d} < 1$.* ∎

Perron—Frobenius theory of non-negative matrices provides us with an estimate on the rate at which the distribution of a random walk on G convergences to its limit distribution.

Lemma 4.5. [15] *Let $P^j_{x,y}$ denote the probability that a random walk starting at node x reaches node y in the j'th transition. Then $P^j_{x,y} \leq \left(\frac{\lambda}{d}\right)^j + \frac{1}{n}$.* ∎

We now fix $\xi = 2\left(\log \frac{d}{\lambda}\right)^{-1}$.

Lemma 4.6. *If* $6\varrho' \leqq \psi_1 \log \frac{d}{\lambda} - \varepsilon$ *for* $0 < \varepsilon < \frac{1}{3}$ *then with probability* $1 - n^{-\varepsilon}$, *all the random walks starting at any processor x are vertex-disjoint from the random walks starting at any vertex* $y, x \neq y$, *and from the set of well-distributed paths connecting* $A \cup B$ *to* $C \setminus \{x\}$.

Proof. Since $x, y \in C$, the shortest path between x and y has at least length $(1-2\varrho'-\varepsilon)\log_d n$ (Lemma 4.2). Let $P(x) = x_1, x_2, \ldots$ and $P(y) = y_1, y_2, \ldots$ denote two random walks starting at x and y respectively. We say that $P(x)$ hits $P(y)$ if $x_i = y_j$ and $j \leqq i$ (i.e. $P(x)$ chose a vertex that was already on $P(y)$ or they chose the same vertex simultaneously). Because of the distance between x and y, $P(x)$ cannot hit $P(y)$ at its first $k = \frac{1}{2}(1-2\varrho'-\varepsilon)\log_d n$ random choices. The probability that $P(x)$ hits $P(y)$ at all is bounded, using Lemma 4.5, by $(\xi \log n)^2 \left(\frac{\lambda}{d}\right)^k$. Overall there are $X = K + K^3 \log^3 n$ random walks. Thus the probability that $P(x)$ hits any of the other $X-1$ random walks is bounded by

$$(\xi \log n)^2 \left(\frac{\lambda}{d}\right)^k (X-1).$$

Similarly, since the paths connecting $A \cup B$ to C are well-distributed, the distance between x and vertices of any path leading from $A \cup B$ to another vertex y in C is at least μ_1. Thus, the probability of a random walk starting at x hitting any of the well-distributed paths connecting $A \cup B$ to $C - \{x\}$ is bounded by

$$2K(\tau_1+\tau_2)\xi \log n \left(\frac{\lambda}{d}\right)^{\mu_1}.$$

Altogether, the probability that any of the X random walks intersect with any other path is bounded by

$$X\left(2K\left(\tau_1\tau_2\xi \log n \left(\frac{\lambda}{d}\right)^{\mu_1} + (\xi \log n)^2 \left(\frac{\lambda}{d}\right)^k\right)(X-1)\right) \leqq n^{-\varepsilon}.$$

Lemma 4.7. *With probability* $1 - \frac{1}{\log n}$ *the algorithm constructs 2K vertex-disjoint paths connecting* $C(a_i)$ *to* $C(b_i)$, $i = 1, \ldots, K$.

Proof. It remains to show that the shortest path connecting $D(a_i)$ to $D(b_i)$ does not use a vertex of any other path. We first observe that given i and a particular element $x \in \mathcal{D}(a_i)$ (endpoint of a random walk), the probability that x is not among the $\frac{n}{K^2 \log^2 n}$ nearest vertices to $D(b_i)$ is at most $\left(1 - \frac{1}{K^2 \log^2 n}\right)\left(1 + \frac{1}{n}\right)$. This follows from the fact that the random walks are long enough to ensure, using Lemma 4.5, that the endpoints are almost random locations, namely, the probability to end in an arbitrary vertex y is bounded above by $\frac{1}{n} + \left(\frac{\lambda}{d}\right)^{\xi \log n} \leqq \frac{1}{n} + \frac{1}{n^2}$. Hence the probability that $D(a_i)$ is outside this set is at most $\left(\left(1 - \frac{1}{K^2 \log^2 n}\right)\left(1 - \frac{1}{n}\right)\right)^{K^2 \log^3 n} \leqq$

$\leqq e^{-\log n} < \frac{1}{n}$. Therefore with high probability, for every i $D(a_i)$ is among the $\frac{n}{K^2 \log^2 n}$ nearest vertices to $D(b_i)$. On the other hand, for each i the vertex $D(b_i)$ was chosen independently of the rest of the paths. Therefore with high probability, for every i, the $\frac{n}{K^2 \log^2 n}$ nearest vertices to $D(b_i)$ contain no vertex of paths leading to any other vertices. (The number of nodes on such paths is at most $O(K \log n)$ so the probability of interference is $O\left(K \log n \frac{1}{K^2 \log^2 n}\right)$ for any particular i, and at most $O\left(\frac{1}{\log n}\right)$ over all pairs.) Since the path connecting $D(a_i)$ to $D(b_i)$ lies entirely within the set of $\frac{n}{K^2 \log^2 n}$ nearest vertices to $D(b_i)$, with high probability, for any i, the shortest path connecting $D(a_i)$ and $D(b_i)$ does not use vertices of any other path. ∎

Successive runs of the algorithm, even on the same input, are probabilistically independent. Thus we can improve the failure probability of the algorithm at the expense of increasing its run-time, and get

Theorem 4.8. *There exists a constant $\varrho'(\alpha, \beta, \gamma, \varphi, d)$ s.t. for every n-vertex d-regular (α, β, γ)-expander with sufficiently large n and $\alpha \geqq 2$, and every set $\mathscr{AB}$ of $K \leqq n^{\varrho'}$ pairs, the pairs can be connected by edge-disjoint paths by a parallel probabilistic algorithm in $O(r \log^2 n)$ parallel steps, using $O((dn)^3)$ processors with success probability $1-e^{-r \log\log n}$.* ∎

4.3. Parallel-distributed algorithm

The ultimate goal of this work is an algorithm that computes disjoint paths in an interconnection network using only n processors which reside in nodes of the network, and communicating by messages through links of the network. Messages have no more than $O(\log n)$ bits and at most one message can traverse an edge in one communication step. In this section we present a version of the parallel algorithm that works in this model.

There are two main difficulties in obtaining such an algorithm. We first have to reduce the number of processors used by the Random-$\mathscr{NC}$ algorithm from $O((dn)^3)$ to $O(n)$. Then we have to show that the PRAM algorithm can be simulated in polylog time in the parallel-distributed model.

We first present an $O(n)$ processor PRAM algorithm. The only part of the previous algorithm that actually needs more than n processors is the flow phase. While we cannot eliminate this phase, we can show that for $K=n^{\varrho''}$, $\varrho''<\varrho'$, the flow phase can be restricted to a subset of the graph with no more then $O(n^{1/6})$ vertices. This flow task can be solved in polylog time using only $O(n)$ processors.

Given a set $\mathscr{AB}$, denote by H_{AB} the set of vertices in the $\log \frac{n^{1/6}}{2K}$-neighborhood of the set $\mathscr{AB}$. We will prove the existence of a set of well-distributed paths inside the graph defined by the set of vertices H_{AB}, $|H_{AB}| \leqq n^{1/6}$. While we can prove the existence of well-distributed paths, we can not predict in advance the set of end-

points C of the paths. Instead we will choose a random set M such that with high probability
1. Every $2K$-subset of M is well-distributed.
2. M contains a set C that can be connected by well-distributed paths to $\mathcal{AB}$.

Let $F_{AB}=\{v|\text{dist}(v, A\cup B)\leqq m\}$. By Fact 2.1, $|F_{AB}|\leqq 2Kd^{m+1}$. To select the set M, each element in F_{AB} starts a random walk of length $R=\log_d \frac{n^{1/6}}{2K}-m$. The set M is the set of end-points of these F_{AB} random walks. For the proofs in this section we fix $\varphi=\frac{1}{24}\log\frac{\lambda}{d}-\varepsilon$ where λ is the second eigenvalue of the adjacency matrix of the graph.

Lemma 4.9. *With high probability every 2K-subset C of the set M is well-distributed w.r.t.* $A\cup B$.

Proof. Let P_v denote the probability that a vertex v is in M. Since P_v is defined by random walks on an expander graph, By Lemma 4.5 $P_v\leqq\left(\left(\frac{\lambda}{d}\right)^R+\frac{1}{n}\right)|F_{AB}|$. Thus, the probability that property (WD1) of Definition 3.3 is violated, i.e., that $\text{dist}(A\cup B, C)<\tau_1+\mu_1+\mu_2+1$, is bounded by

$$2Kd^{\tau_1+\mu_1+\mu_2+1}\cdot 2Kd^{m+1}\cdot\left(\left(\frac{\lambda}{d}\right)^R+\frac{1}{n}\right)\leqq n^{-\varepsilon}.$$

Similarly, the probability that property (WD2) of Definition 3.3 is violated, i.e. that the minimum distance inside C is smaller than $\mu_1+\mu_2+1$ is bounded by

$$2Kd^m d^{\mu_1+\mu_2+1}\cdot 2Kd^{m+1}\cdot\left(\left(\frac{\lambda}{d}\right)^R+\frac{1}{n}\right)\leqq n^{-\varepsilon}. \quad \blacksquare$$

Theorem 4.10. *There exists a constant* $\varrho''=\varrho''(\alpha,\beta,\gamma)$ *such that for any n-vertex d-regular* (α,β,γ)*-expander G with sufficiently large n and* $\alpha\geqq 2$, *and for any set of vertices* $A\cup B$ *such that* $|A\cup B|=2K\leqq 2n^{\varrho''}$, *with high probability there exists a set of vertices* $C\subset M$, $|C|=2K$, *s.t. C is well-distributed w.r.t.* $A\cup B$ *and is connected to* $A\cup B$ *by a set of well-distributed paths using only vertices of* H_{AB}, *and going through no* μ_1*-neighborhood of any vertex in M, except their endpoint.*

Proof. The proof follows the lines of Section 3 but calls for some major changes, since we need to prove the existence of paths residing *inside* the subgraph H_{AB}, which is not an expander. Also, the set C cannot be chosen in advance but is computed dynamically. This computation cannot use an identical variant of Lemma 3.8, because the proof of that lemma argues about large (size $n/2$) sets, and therefore cannot be repeated inside H_{AB}. Instead we need to employ a "random-walk" argument.

The proof goes as follows. We identify an $(A\cup B, l, m)$-structure G_S, a set of vertices $D=\{d_1, \ldots, d_{2K}\}$ and a set P of edge-disjoint paths $p(v_i, d_i)$, $1\leqq i\leqq 2K$ as in Theorem 3.10. Thus we connected the vertices in $A\cup B$ with the vertices of D. We now attempt to find a subset C, $C\subset M\subset F_{AB}$, $|C|=2K$, and paths $p(d_i, c_i)$, $1\leqq i\leqq K$, connecting D to C as required.

For the purpose of the (non-constructive) proof, we assume that together with the set M we are given the set of random walks $\tilde{M}$, that defined M. Since $D \subseteq F_{AB}$, there is a path in $\tilde{M}$, for each $d \in D$, connecting it to a vertex in M. We use this set of paths and define C to be the $2K$ end-points of the walks starting in vertices of D.

Lemma 4.11. *With high probability the set of paths connecting $A \cup B$ to C is well-distributed.*

Proof. Each path contains two segments. The length of the segment inside the structure G_S is bounded by τ_1, and the length of the rest of the path is bounded by $1/6 \log n < \tau_2$. Thus, the total length of the $2K$ paths is bounded by $2K(\tau_1+\tau_2)$, and requirement (WD3) of Definition 3.3 is satisfied.

To prove that the paths are edge-disjoint we observe that the segments of the paths connecting D to C are independent random walks. Furthermore, the choice of D guarantees that the startpoints of the random walks are at least $\mu_1+\mu_2+1$ apart from each other. Hence, using an argument similar to that of Lemma 4.6 we prove that the probability that any of the random walk segments intersects with any other path (either from $A \cup B$ to D or from D to C) is bounded by

$$K^2(\tau_1+\tau_2)\log n \left(\frac{\lambda}{d}\right)^{\mu_1} \leqq n^{-\varepsilon}.$$

Using similar argument, the probability that any of the paths uses a vertex in the μ_1-neighborhood of the end-points of any other path is bounded by $4K^2 \log n d^{\mu_1}\left(\frac{\lambda}{d}\right)^R \leqq n^{-\varepsilon}$. Thus, requirement (WD4) of Definition 3.3 holds with high probability. ▮

This completes the proof of Theorem 4.10. ▮

Algorithm 3

Step 1: Mark the set $F_{AB}=\{v | \mathrm{dist}(v, A\cup B) \leqq m\}$ and the set $H_{AB}=\{v|\mathrm{dist}(v, A\cup B) \leqq$ $\leqq \frac{1}{6}\log_d n\}$.

Step 2: [Run $2Kd^m$ random walks in parallel. Each $v \in F_{AB}$ is a start point of d^m walks. The variable $COUNT(v, i)$ holds the number of walks visiting v at step i.]

All processors v do in parallel:

Step 2.1: $COUNT(v, 1)=d^m$ if $v \in F_{AB}$, 0 otherwise;

Step 2.2: For $i=1$ to $R=\frac{1}{6}\log_d \frac{n}{2K}-m$ do:

2.2.1: Choose randomly a partition of $COUNT(v, i)$ into d non-negative integers, with all possible partitions having equal probabilities. Let $v_1, \ldots, v_d$ be the d neighbors of v. Assign the chosen integers into variables $Z_{v,v_1,i}, \ldots, Z_{v,v_d,i}$.

2.2.2: For $j=1, \ldots, d$ do:

Send the value $Z_{v,v_j,i}$ to v_j,
Receive the value $Z_{v_j,v,i}$ from v_j.
[$Z_{v,v_j,i}$ paths leave vertex v to vertex v_j in their i-th transition].

2.2.3: $COUNT(v, i+1)=\sum_{j=1}^{d} Z_{v_j,v,i}$.

Step 3: Let M be the set of end-points of the $2Kd^m$ random walks.
Step 4: Construct a flow network $\tilde{H}_{AB}$ as in the PRAM algorithm, using only the vertices of H_{AB}, and replacing C by M.
Step 5: Use $O(n)$ processors to run a probabilistic parallel weighted matching algorithm to find the min-cost max-flow between s and t in the graph $\tilde{H}_{AB}$ of $O(n^{1/6})$ vertices.
Step 6: Let C be the set of vertices in M which were connected to vertices in $A \cup B$ by the flow algorithm.
Step 7—8: As steps 3—4 of the Random-$\mathcal{NC}$ algorithm.

Lemma 4.12. *With high probability, steps 1—6 of the algorithm generate a well-distributed set of vertices $C \subset M$, $|C| = 2K$, and a well-distributed set of paths, connecting $A \cup B$ to C.* ∎

We now turn to the question of simulating the $O(n)$ processor PRAM algorithm on a network of n processors connected by a bounded-degree interconnection expander. It has been shown in [7] that if the PRAM variables are distributed by a hash function among the processors and if the processors are connected by a butterfly network, then the simulation of a any T PRAM steps can be done in $O(T \log n)$ steps. It has been shown in [13], (Theorem 5.1) that each butterfly step can be simulated in $O(\log n)$ steps on any expander interconnection graph. Combining these results together we prove

Theorem 4.13. *Any set of K pairs of vertices, $K \leqq n^{\varrho''}$, can be connected by edge-disjoint paths, in $O(r \log^4 n)$ parallel steps of the parallel-distributed model with success probability $1 - e^{r \log \log n}$.* ∎

5. The vertex-disjoint case

While the expansion property guarantees the existence of edge-disjoint paths for a large number of pairs, this is not the case in the vertex-disjoint case.

To begin with, it is clear that for every d-regular graph (regardless of its expansion), if $K > \frac{d}{2}$ then there are sets of K pairs of nodes in the graph such that not all paths are possible. To construct such a set, simply take an arbitrary vertex as a_1, choose all its d neighbors as $a_2, b_2, \ldots, a_K, b_K$ and take any other vertex as b_1. This proves:

Theorem 5.1. *Given any d-regular graph, there is always a set of $K = \frac{d}{2} + 1$ pairs of vertices that cannot be connected by vertex-disjoint paths.* ∎

However, there are examples in which even less than $\frac{d}{2}$ pairs cannot be handled. Such examples can be constructed in the following way. Let the set of pairs be $A = \{(a_i, b_i) | 1 \leqq i \leqq K\}$, and let $X = \{a_1, v_1, v_2, \ldots, v_t\}$ and $Y = \{a_2, b_2, \ldots, a_K, b_K\}$, where $t = d - 2K + 2$. Consider any (α, β)-expander graph containing $\{a_1, b_1, \ldots$ $\ldots, a_K, b_K\} \cup \{v_1, \ldots, v_t\}$ in its vertex set and having C_X and $B_{X,Y}$ as subgraphs (where C_X is a clique on the vertices of X and $B_{X,Y}$ is the complete bipartite graph on X and Y). It is clear that there are no vertex-disjoint paths connecting the pairs (particularly, a_1 is disconnected from b_1 by the other vertices in A). In order for us to have such an expander we need to guarantee that $(2K-2) \geqq \alpha(t+1)$, or, that $K \geqq \frac{\alpha(d+3)+2}{2(\alpha+1)}$. In particular, this requires that $\alpha < \frac{d-2}{3}$. For such an expander, all we can hope for is handling at most $K \leqq \frac{\alpha(d+3)+2}{2(\alpha+1)}$ paths. Note that this bound is no larger than $\frac{d}{2}$ and it converges to 0 with α. Thus

Theorem 5.2. *For every integer $d>2$ and for every $0<\alpha<(d-2)/3$ there exist a d-regular (α, β, γ)-expander and a set of $K=\frac{\alpha(d+3)+2}{2(\alpha+1)}+1$ pairs of vertices that cannot be connected by vertex-disjoint paths.* ∎

Next we prove the existence of vertex-disjoint paths for up to $K \leqq \frac{\alpha-3}{2}$ pairs.

Theorem 5.3. *Given an n-vertex d-regular (α, β, γ)-expander with sufficiently large n, for every set of $K \leqq \frac{\alpha-3}{2}$ pairs of vertices there exist connecting vertex-disjoint paths, and these paths can be found by parallel probabilistic algorithms*
1. in $O(\log^2 n)$ time on an $O((dn)^3)$ processor PRAM.
2. in $O(\log^4 n)$ time on the parallel-distributed model.

Proof. We first show that for every two sets of vertices $Z \subseteqq U \subseteqq V$ where $|U| \leqq \beta n$ and $Z \leqq \alpha-3$ there exists a special subgraph which we call (Z, U)-forest. This subgraph is a collection of trees, in which the nodes of U are internal and the nodes of Z are all roots.

Definition 5.1. For any two given sets $Z \subset U \subset V$, a **(Z, U)-forest** is a directed acyclic subgraph $G_Z=(V_Z, E_Z)$ with the following properties:
(Z1) $U \subseteqq V_Z \subseteqq V$.
(Z2) $E_Z \subseteqq E$ (looking at the underlying undirected edges).
(Z3) For every $v \in V_Z$, *indegree*$(v) \leqq 1$.
(Z4) For every $v \in U$, *outdegree*$(v)=2$.
(Z5) For every $v \in Z$, *indegree*$(v)=0$.

Lemma 5.4. *Let $G=(V, E)$ be an n-vertex d-regular (α, β)-expander s.t. $\alpha \geqq K+3$ for some integer $K \geqq 0$. Then for every set $U \subset V$ s.t. $|U| \leqq \beta n$ and for every set $Z \subset U$ s.t. $|Z|=K$ there exists a (Z, U)-forest.*

Proof. Similar to that of Theorem 3.10, after re-defining $\hat{\Gamma}$ (for the purpose of this proof only) to be $\hat{\Gamma}(X)=\Gamma(X)-X-Z$. Some obvious modifications are necessary in Lemmas 3.2 and 3.4.5.

Lemma 5.5. *Let $G=(V, E)$ be an n-vertex d-regular (α, β)-expander, where $\alpha \geqq 2K+3$, and let $l=\lfloor \alpha-1 \rfloor$. For every set $A \cup B \subset V$, $A \cup B=\{v_1, \ldots, v_{2K}\}$, there exist trees $T_1, \ldots, T_{2K}$ in the graph such that*
(1) Each T_i is a full binary subtree of depth m, hence it has 2^m leaves.
(2) The root of T_i is v_i.
(3) The trees are vertex-disjoint.

Proof. Let $U=N(m, A\cup B)$. By the previous theorem there exists an $(A\cup B, U)$-forest. In this forest each node v_i of $A\cup B$ is a root of some tree S_{v_i} which is complete up to depth m at least, so choosing $S_{v_i}^m$ as T_i meets the requirements. ∎

To complete the proof of Theorem 5.3, we use the disjoint trees to construct a set D as in Lemma 3.11, only now the vertices in the set $A\cup B$ are connected to D by vertex disjoint paths. The rest of the proof is similar to the edge-disjoint case, since the paths constructed there between the set D and the set C, and the paths connecting pairs of vertices in C were already vertex-disjoint. ∎

Acknowledgements. We are grateful to N. Pippenger and E. Shamir for many stimulating discussions on the relationships between expanders and random-graphs.

Quite surprisingly, we could not find a reference for the fact that the edge-disjoint paths problem is $\mathcal{NP}$-complete. We thank A. LaPaugh for pointing out the reduction to ND39 [6].

References

[1] Ajtai M., Komlós J. and Szemerédi E., An $O(n \log n)$ Sorting Network, *15th Symp. on Theory of Computing*, **1983**, 1—9.
[2] R. Aleliunas, Randomized Parallel Communication, *1st Symp. on Principles of Distributed Computing*, **1982**, 60—72.
[3] N. Alon, Eigenvalues and Expanders, *Combinatorica*, **6** (1986), 83—96.
[4] P. Feldman, J. Friedman and N. Pippenger, Non-Blocking Networks, *18th Symp. on Theory of Computing*, **1986**, 247—254.
[5] J. Friedman and N. Pippenger, Expanding Graphs Contain All Small Trees, *Report RJ* 5145 (53485), *IBM Almaden Research*, May **1986**.
[6] M. R. Garey and D. S. Johnson, *Computers and Intractability; a Guide to the Theory of NP-Completeness*, W. H. Freeman and Co., San-Francisco, **1979**.
[7] A. R. Karlin and E. Upfal, Parallel Hashing — an Efficient Implementation of Shared Memory, *18th Symp. on Theory of Computing*, **1986**, 160—168.
[8] R. M. Karp, E. Upfal and A. Wigderson, Constructing Perfect Matching is in Random-$\mathcal{NC}$, *Combinatorica*, **6** (1986), 35—48.
[9] E. L. Lawler, Combinatorial Optimization: Networks and Matroids, Holt, Rinehart and Winston, 1976.
[10] A. Lubotzky, R. Phillips and P. Sarnak, Ramanugan Conjecture and Explicit Constructions of Expanders, *18th Symp. on Theory of Computing*, **1986**, 240—246.
[11] K. Mulmuley, U. V. Vazirani and V. V. Vazirani, Matching is as Easy as Matrix Inversion, *19th Symp. on Theory of Computing*, **1987**, 345—354.
[12] N. Pippenger, Parallel Communication with Bounded Buffers, *25th Symp. on Foundations of Computer Science*, **1984**, 127—136.
[13] D. Peleg and U. Upfal, The Token Distribution Problem, *26th Symp. on Foundations of Computer Science*, **1986**, 418—427.
[14] N. Robertson and P. D. Seymour, Graph Minors-XIII; Vertex-Disjoint Paths, *Manuscript*. **1986**.
[15] E. Senata, *Non-negative Matrices and Markov Chains*, Springer-Verlag, **1973**.
[16] E. Shamir and A. Schuster, Parallel Routing in Networks: Back to Circuit Switching?, *Manuscript*, **1986**.
[18] L. Valiant, A Scheme for Fast Parallel Communication, *SIAM J. on Computing*, **11** (1982), 350—361.

David Peleg

Computer Science Department
Stanford University
Stanford, CA 94305
U. S. A.

Eli Upfal

IBM Almaden Research Center
650 Harry rd.
San Jose, CA 95120
U. S. A.