Communication Structures: Sparse Partitions

Papers

1. B. Awerbuch and D. Peleg. Sparse partitions. In In 31st Annual IEEE Symposium on Foundations of Computer Science, pages 503–513, 1990.

2. David Peleg. Distributed Computing: A Locality-Sensitive Approach. SIAM Monographs on Discrete Mathematics and Applications, 2000.

1 Motivation

Traditional protocols for network control and management usually require all the sites/nodes to participate in their activities and this entails the maintenance of considerable global information. As the size of the network grows, the information requirements become problematic and also maintaining the structure is more complex. At the same time, global knowledge is not always necessary : certain tasks could be achieved in a local manner, and certain tasks *are* local. In either case, it is better if the execution of the task involved only nodes in or around the region. Further it is desirable that the cost of the task is proportional to the level of locality. This motivates the study of locality in distributed systems which is the subject matter of David Peleg's book [2]. In this part of the lecture we will discuss the following:

- Locality preserving representations
- Some basic algorithms for constructing partitions and covers

In the later part we(Seth) will present an application of LP-representations.

1.1 LP-Representations

An LP-representation is one which faithfully captures the topology of the network itself. The basic premise behind using these representations is that using them allows us to maintain smaller amount of information, thus reducing the computational and storage *costs*. The price to be paid is a certain loss in *accuracy*. The actual meaning of cost and accuracy depends on the specific application. LP-representations fall into two broad categories: *Clustered representations* and *Skeletal representations*.

- Vertices are grouped into small groups of connected *clusters*. Two design parameters : *radius* of clusters, and *interaction* or overlap between clusters. We will discuss the clustered approach in more detail.

- Sparse spanning subgraphs of the network, e.g. a spanning tree. Here the sparsity is very low but locality is sacrificed – two neighboring nodes in the original network might be very far apart in the spanning tree. This is remedied by *graph spanners*, which generalize the concept of a spanning tree (allowed to have cycles). The locality properties are measured by *stretch*, that is the (worst) factor by which the distances between vertices increase as a result of using spanner edges alone and ignoring non-spanner edges. We won't discuss the skeletal approach.

1.2 Cluster based approach

An important observation is that the graph theoretic criteria for a cluster radius and interaction mentioned above is closely related to complexity measures in distributed algorithms. Smaller radius or size of clusters lead to lower time complexities. Sparsity or low interaction among clusters guarantees low memory requirements. Communication complexity depends on both.

1.2.1 Quick review of terminology

- Arbitrary weighted graph $G(V, E, w)$, $w : E \to R^+$
- For $S \subseteq V$, $G(S)$ is the subgraph induced by S.
- A cluster S is the connected subgraph $G(S)$.
- A cover is a collection of subgraphs $\mathcal{S} = \{S_1, \ldots, S_m\}$ such that $\cup_i S_i = V$.
- A partial partition is a collection of disjoint clusters $\mathcal{S} = \{S_1, \ldots, S_m\}$, with $S \cap S' = \emptyset$.
- A partition is a partial partition which is also a cover.
- ρ-neighborhood of vertex v, $\Gamma_\rho(v) = \{w \mid dist(w, v) \leq \rho\}$. Extended to a set of vertices W, $\Gamma_\rho(W) = \{\Gamma_\rho(v) \mid v \in W\}$.
- $Diam(G) = max_{u,v \in V}(dist_G(u, v))$.
- $Rad(v, G) = max_{w \in V}(dist(v, w))$.
- $Rad(G) = min_{v \in V}(Rad(v, G))$. For $G = (\{v\}, \emptyset)$, $Rad(G) = Diam(G) = 1$.
  $Rad(G) \leq Diam(G) \leq 2Rad(G)$.

1.3 Sparsity and Locality Measures

Locality level of a cluster is measured by its radius or size. Given a vertex $v \in S$ the radius of S w.r.t. v is defined as in the induced subgraph $G(S)$, that is,

- $Rad(v, S) = Rad(v, G(S)) = max_{w \in S}\{dist_{G(S)}(v, w)\}$.
- $Rad(S) = min_{v \in S} Rad(v, G(S))$
- Given a cover $\mathcal{S} = \{S_1, \ldots, S_m\}$. $Rad(\mathcal{S}) = max_i Rad(S_i)$.
- Similarly $Diam(\mathcal{S}) = max_i Diam(S_i)$.

There are several plausible measures of sparsity, usually it is a function of the degrees of vertices or clusters in the cover or the graph itself.

1.3.1 Covers

- $deg(v, \mathcal{S})$ = degree of v in the hypergraph $(V, \mathcal{S})$.
- maximum degree of a cover $\Delta(\mathcal{S}) = max_{v \in V} deg(v, \mathcal{S})$.
- average degree of a cover $\bar{\Delta}(\mathcal{S}) = \Sigma_{v \in V} deg(v, \mathcal{S})/n$.

1.3.2 Partitions

Contract each cluster into a vertex of a *cluster graph* $\bar{G}$.

- vertex neighborhood of S: $\Gamma^v_\rho(S, G) = \bigcup_{v \in S} \Gamma_\rho(v, G)$
- cluster neighborhood of S: $\Gamma^c_\rho(S) = \Gamma_\rho(S, \bar{G}(\mathcal{S}))$.
- maximum vertex degree, $\Delta^C_v(\mathcal{S}) = max_{S \in \mathcal{S}} \{\frac{|\Gamma^v(S)|}{|S|}\}$, cluster degree $\Delta^C_c(\mathcal{S}) = max_{S \in \mathcal{S}} \{\frac{|\Gamma^c(S)|}{|S|}\}$,
- average vertex degree, $\bar{\Delta}_v(\mathcal{S}) = \frac{1}{n} \Sigma_{S \in \mathcal{S}} |\Gamma^v(S)|$, cluster degree $\bar{\Delta}_c(\mathcal{S}) = \frac{1}{n} \Sigma_{S \in \mathcal{S}} |\Gamma^c(S)|$

2 Basic partition algorithm

Potential problem: the cover may not conform with the topological structure of the network. E.g., consider a cover in which two neighboring nodes A and B which belong to different clusters. This would make communication between A and B more complex than it is required by locality.

2.1 A Sequential Construction `Basic_part`

- $\mathcal{S} \leftarrow \emptyset$
- **While** $V \neq \emptyset$ **do:**
  1. select arbitrary $v \in V$
  2. $S \leftarrow \{v\}$
  3. **While** $|\Gamma^v(S)| > n^{1/k} |S|$ **do:** $S \leftarrow \Gamma(S)$ **wend**
  4. $\mathcal{S} \leftarrow \mathcal{S} \cup \{S\}$; $V \leftarrow V - S$.
- **wend**
- **Output**$(\mathcal{S})$

Theorem 2.1 Given an n vertex unweighted graph G=(V,E) and an integer $k \geq 1$, Algorithm `Basic_part` constructs a partition $\mathcal{S}$ that satisfies :

1. $Rad(\mathcal{S}) \leq k - 1$,and
2. The cluster graph has at most $n^{1+1/k}$ edges.

Proof: Every S added to $\mathcal{S}$ is a (connected) cluster. Also all the clusters are disjoint. For property (2), note that at the termination of the internal loop $|\Gamma^v(S)| \leq n^{1/k}|S|$. Therefore the number of intercluster edges touching S is atmost $n^{1/k}|S|$. This number can only decrease as a result of iterations (vertices are merged into clusters). Therefore $|\bar{E}| = \Sigma_{S\in\mathcal{S}} n^{1/k}|S| = n^{1+1/k1}$. For part (1), consider some iteration of the outer loop in which the inner loop has executed J times. Let $S_i, 1 \leq i \leq J$ denote the S constructed after the i^{th} internal iteration. Note that $S_i \geq n^{i/k}$ and $S_i > n^{i/k}$ for $i > 1$. It follows that $J \leq k-1$, else $|S_J| > n$. Finally note that $Rad(v, S_i) = i$, each iteration adds one layer. Combining the above two facts we conclude $Rad(S_J) \leq k-1$.

Trade-off between radius and sparsity. Remark : possible to make (2) $O(k.n^{1/k})$.

2.2 Coarsening

Definition 2.2 Given two covers $\mathcal{S} = \{S_1, \ldots, S_p\}$ and $\mathcal{T} = \{T_1, \ldots, T_q\}$, $\mathcal{T}$ coarsens $\mathcal{S}$ if for every $S_i \in \mathcal{S}$ there exists $T_j \in \mathcal{T}$ such that $S_i \subseteq T_j$. Radius ratio of the coarsening is $Rad(\mathcal{T})/Rad(\mathcal{S})$.

Given some initial cover $\mathcal{S}$ (with appropriate locality level required and having the desired radius property but not necessarily sparse), the goal is to construct via repeated cluster merging, a coarsening cover $\mathcal{T}$ with relatively low overlaps. Same trade-off between radius and overlap as above. `Av_Cover` constructing covers with low average degree:

- $\mathcal{T} \leftarrow \emptyset$
- **While** $\mathcal{S} \neq \emptyset$ **do:**
  1. select arbitrary $S_0 \in \mathcal{S}$
  2. $Z \leftarrow S_0$
  3. **Repeat**
     (a) $Y \leftarrow Z$
     (b) $\mathcal{Z} \leftarrow \{S | S \in \mathcal{S}, S \cap Y \neq \emptyset\}$
     (c) $Z \leftarrow \bigcup_{S\in\mathcal{Z}} S$
     until $|Z| \leq n^{1/k}|Y|$
  4. $\mathcal{S} \leftarrow \mathcal{S} - \mathcal{Z}$
  5. $\mathcal{T} \leftarrow \mathcal{T} \cup \{Z\}$.

  wend
- **Output** $\mathcal{T}$

Theorem 2.3 Given a graph $G = (V, E), |V| = n$, a cover $\mathcal{S}$ and an integer $k \geq 1$, `AV_cover` constructs a sparsening cover $\mathcal{T}$ that satisfies:

1. $\mathcal{T}$ coarsens $\mathcal{S}$,
2. $Rad(\mathcal{T}) \leq (2k+1)Rad(\mathcal{S})$, and

[1] Note here that the collapsing of multiple edges to the same cluster is node done by this algorithm but is handled by the definition of $\bar{E}$

3. $\bar{\Delta}(\mathcal{T}) \leq n^{1/k}$.

Proof: (1) The elements in $\mathcal{S}$ are clusters and every Z added to $\mathcal{T}$ is also a cluster. Property 1 follows by noting the fact that each of the original clusters in $\mathcal{S}$ is covered by one of the clusters in $Z \in \mathcal{T}$ by construction.

Kernels are disjoint. By contradiction suppose $v \in Y \cap Y'$, and suppose Y was created in an earlier iteration. Since $v \in Y'$ there must be some cluster S' such that $v \in S'$ and $S' \in \mathcal{S}$ when the algorithm started constructing Y'. But such a cluster S' would satisfy $S' \cap Y \neq \emptyset$, and therefore $3(b)$ would have added S' to $\mathcal{Z}$ and removed S' from $\mathcal{S}$.

Property (3), from the termination condition of the internal loop $|Z| \leq n^{1/k}|Y|$. Therefore $\Sigma_Z|Z| \leq \Sigma_Y n^{1/k}|Y| \leq n^{1+1/k}$.

Property (2), consider some iteration of the algorithm, starting with selection of some cluster S. Let J be the number of times the internal loop was executed. Note that $Y_i = Z_{i-1}$, $\mathcal{Z}_i$ is constructed from Y_i and $Z_i = \cup \mathcal{Z}_i$. Observe $|Z_i| \geq n^{i/k}$ for every $0 \leq i \leq J-1$ and $|Z_i| > n^{i/k}$ for $i \geq 1$ (can be proved by induction Lemma 12.2.3). therefore $J \leq k$.

Lemma[12.2.4] For all $1 \leq i \leq J$, $Rad(Y_i) \leq (2i-1)Rad(\mathcal{S})$.
Proof: By induction on i, base case $Y_1 = S \in \mathcal{S}$. Induction step $Rad(Y_i) = Rad(Z_{i-1}) \leq Rad(Y_{i-1}) + 2Rad(\mathcal{S})$, since $\mathcal{Z}_{i-1}$ is created from Y_{i-1} by adding into it all $\mathcal{S}$ clusters intersecting it and $Y_i = Z_{i-1}$ is a merge of all the clusters in $\mathcal{Z}_{i-1}$. therefore $Rad(Y_J) \leq (2k-1)Rad(\mathcal{S})$, and $Rad(Z_J) \leq (2k+1)Rad(\mathcal{S})$.

Procedure `Cover`(called `Part` in the book) for constructing covers, partitions and partial partitions, with low max degree. Input is a graph G, and a collection of clusters $\mathcal{R}$, the output is a subset $\mathcal{DR} \subseteq \mathcal{R}$ and its coarsened set $\mathcal{DT}$, which subsumes $\mathcal{DR}$. The objective is to subsume many clusters in $\mathcal{R}$.

- $\mathcal{U} \leftarrow \mathcal{R}; \mathcal{DT} \leftarrow \emptyset; \mathcal{DR} \leftarrow \emptyset$
- **While** $\mathcal{U} \neq \emptyset$ **do:**
  1. select arbitrary $S_0 \in \mathcal{U}$
  2. $Z \leftarrow S_0$
  3. **Repeat**
     (a) $\mathcal{Y} \leftarrow \mathcal{Z}$
     (b) $Y \leftarrow \bigcup_{S \in \mathcal{Y}} S$
     (c) $\mathcal{Z} \leftarrow \{S | S \in \mathcal{U}, S \cap Y \neq \emptyset\}$
     until $|\mathcal{Z}| \leq |\mathcal{R}|^{1/k}|\mathcal{Y}|$
  4. $\mathcal{U} \leftarrow \mathcal{U} - \mathcal{Z}$
  5. $\mathcal{DT} \leftarrow \mathcal{DT} \cup \{Y\}$.
  6. $\mathcal{DR} \leftarrow \mathcal{DR} \cup \mathcal{Y}$.

  wend
- **Output** $\mathcal{DR}, \mathcal{DT}$

Main difference from the previous: keeps unmerged collections $\mathcal{Y}$ and $\mathcal{Z}$, sparsity condition is based on the number of original clusters not their sizes. Takes only the kernel of the cluster after discarding a layer.

Lemma 2.4 (3.2) Given a graph $G = (V, E), |V| = n$, a cover $\mathcal{R}$ and an integer $k \geq 1$, the collections $\mathcal{DT}$ and $\mathcal{DR}$ from `Cover` satisfy:

1. $\mathcal{DT}$ coarsens $\mathcal{DR}$,
2. $Y \cap Y' = \emptyset$,
3. $|\mathcal{DR}| \geq |\mathcal{R}|^{1-1/k}$, and
4. $Rad(\mathcal{DT}) \leq (2k-1)Rad(\mathcal{R})$.

Proof: of parts (1) and (2) exactly similar to the above. For part (3), from termination condition $|\mathcal{Z}| \leq |\mathcal{R}|^{1/k}|\mathcal{Y}|$. Therefore $|\mathcal{R}| = \Sigma_{\mathcal{Z}}|\mathcal{Z}| \leq \Sigma_{\mathcal{Y}}|\mathcal{R}|^{1/k}|\mathcal{Y}| = |\mathcal{R}|^{1/k}|\mathcal{DR}|$. Part(4) $|\mathcal{Z}_i| \geq |\mathcal{R}|^{i/k}$ for every $0 \leq i \leq J-1$. This implies $i \leq k$. Also prove $Rad(Y_i) \leq (2i-1)Rad(\mathcal{R})$, therefore $Rad(Y_J) \leq (2k-1)Rad(\mathcal{R})$.

`Max_cover` uses the `Cover`: Input is a graph and a cover $\mathcal{S}$, outout collection of cluster covers $\mathcal{T}$. $\mathcal{R}$ is a set of remaining clusters which are not covered(subsumed). Each phase adds a subcollection $\mathcal{DT}$ to $\mathcal{T}$ and removes the subsumed clusters $\mathcal{DR}$ from $\mathcal{R}$.

- $\mathcal{R} \leftarrow \mathcal{S}, \mathcal{T} \leftarrow \emptyset$
- **while** $\mathcal{R} \neq \emptyset$

  1. $(\mathcal{DR}, \mathcal{DT}) \leftarrow$ `Cover`$(\mathcal{R})$
  2. $\mathcal{T} \leftarrow \mathcal{T} \cup \mathcal{DT}$
  3. $\mathcal{R} \leftarrow \mathcal{R} \setminus \mathcal{DR}$

  wend

Theorem 2.5 (3.1) 1. $\mathcal{T}$ coarsens $\mathcal{S}$

2. $Rad(\mathcal{T}) \leq (2k-1)Rad(\mathcal{S})$
3. $\Delta_v(\mathcal{T}) \leq 2k|\mathcal{S}|^{1/k}$

Proof: Let $\mathcal{R}^i$ be $\mathcal{R}$ at the beginning of phase i, $r_i = |\mathcal{R}^i|$, and $\mathcal{DT}^i$ and $\mathcal{DR}^i$ be the values at the end of phase i. $\mathcal{T}$ coarsens $\mathcal{S}$ because $\mathcal{T} = \cup\mathcal{DT}$ and $\mathcal{S} = \cup\mathcal{DR}$ and from property (1) of lemma 3.2. (2) follows from property (4) of previous lemma. For (3), note that each vertex v participates in atmost one cluster in the collection $\mathcal{DT}^i$ by property (2) of lemma 3.2. Therefore we have to bound the number of phases performed by `Max_cover`. Again by property (3) of the previous lemma in every phase i at least $\mathcal{DR}^i \geq |\mathcal{R}^i|^{1-1/k}$ clusters of $\mathcal{R}^i$ are removed., i.e., $r_{i+1} \leq r_i - r_i^{1-1/k}$.

Claim 2.6 (3.3) For the above recurrence relation, let J denote the least index i such that $x_J \leq 1$ given $x_0 = |\mathcal{S}|$. Then $J < \frac{k}{ln2}|\mathcal{S}|^{1/k}$.

Since each vertex participates in atmost J clusters $\Delta_v^C \leq 2k|\mathcal{S}|^{1/k}$.

Remark: Guaranteeing $O(n^{1/k})$ average overlap necessitates increase of radius by a factor of $\Omega(k)$ in the worst case. These trade-offs are nearly tight.

Based on this idea of coarsening algorithms $\texttt{Max\_part}_v$ and $\texttt{Max\_part}_c$ in Chapter 13 minimize Δ_v and Δ_c respectively.

3 Distributed Construction

3.1 Deterministic Distributed Algorithm for Partitions

Simple algorithm `Basic_Part`, actually sequential — at a given point of time all the activity is confined within a zone of radius $2k + 2$ radius. Within this zone exploration happens in parallel.

Three parts of the algorithm:

1. `Cluster_cons`: construct cluster around a given vertex v. Constructs a BFS-tree rooted at v. But exploration is limited to *free* vertices and upto a depth at which the $n^{1/k}$ condition is satisfied.

   Time for each cluster is $O(r_i)$, `Time(Cluster_cons)` $= O(nk)$ careful analysis gives $O(n)$. `Message(Cluster_cons)` $= O(nk + |E|)$.

2. `Next_ctr`: select next cluster center. Selecting the next vertex from the rejected layer does not work generally, Z could be empty even before all the vertices are considered. Embed clustering algorithm in global DFS.

   This is more expensive than normal DFS, at every vertex can require $O(r_i)$ time and $O(n)$ messages. `Time(Next_ctr)` $= O(nk)$, `Message(Next_ctr)` $= O(n^2)$.

3. `Rep_edge`: select representative inter-cluster edges. Define ordering on edges then both clusters would select the same edge. `Time(Rep_edge)`$= O(n)$, `Message(Rep_edge)` $= O(n^2)$.

Improved `Dist_part` time $O(n)$ message $O(n^2)$.

3.2 Low communication synchronous algorithms

We would like the number of messages required for the merging process to be proportional to the size of the size of the constructed cluster. So preconstructing is out of the question, the total size of clusters may be $O(n^2)$. So the way to do it is by BFS, but BFS may often go into regions where it will not find any new clusters to merge, to avoid such fruitless expeditions one can think of marking the vertices at the border of unexplored territory, but this again requires every vertex to know which clusters it belongs to.

Four layered construction (we wont discuss, Chapter 21.2) gives `message(Sync_Av_Cover)`$= O(E + n^{1+1/k}\log n)$, `message(Sync_Max_Cover)`$= O(E + n^{1+1/k}k^2)$.

3.3 Randomized Distributed Algorithm for Covers

Weak clusters, disconnected subgraphs are also considered to be clusters. Similarly weak partitions. The measure of radius and diameter is now not based on the subgraph induced by the cluster but on the entire graph. $WRad(v, G(S)) = max_{w\in S}\{dist_G(v, w)\}, WRad(S) = min_{v\in S}\{WRad(v, S)\}$. Similarly weak diameter. The time complexity of the algorithm is $O(r)$ where r is the radius of the neighborhood.

- $B \leftarrow \lfloor\frac{2\log n}{\log(1/p)}\rfloor$

- $P_r[r_v = j] = \begin{cases} p^j(1-p), & j = 0, \ldots, B-1 \\ p^B, & j = B \end{cases}$
- Broadcast $(ID(v), r_v)$ to all vertices in the r_v-neighborhood, $\Gamma_{r_v}(v)$.
- Select the pair with the largest identifier $ID(w)$, among all messages received (including self), set $H(v) = w$.
- If $dist(w, v) < r_w$, then join the cluster C_w rooted at w[2].

It is possible to create disconnected clusters and also the root of a cluster may join a different cluster.

Lemma 3.1 With probability at least $1 - 1/n$ every vertex v chooses a radius $r_v < B$.

Lemma 3.2 The partition $\mathcal{S}$ constructed by the above algorithm satisfies:

1. the partition S is a 2-separated partial partition (min dist between any two clusters is 2).
2. $WRad(\mathcal{S}) \leq \frac{2\log n}{\log(1/p)}$.
3. $P_r[v \in R] \geq p(1 - 1/n)$ for each vertex in G.

Proof sketch: (1) is by contradiction, $\mathcal{S}$ is clearly disjoint. If not 2 separated then there would be two adjacent vertices belonging to differnet clusters. (2) is by definition of r_v.

3.3.1 Related problems

- Graph decomposition
- Graph coloring and independent sets.
- Synchronizers
- Extending the algorithms to the asynchronous systems
- Failures and stabilization.

[2]Due to this strict inequality $<$, nodes may remain homeless, and this also gives rise to disconnected clusters and root nodes not belonging to its own particular cluster