Publish-Subscribe Overlay Network Design

Melih Onus
Department of Computer Science and Engineering
Arizona State University
Tempe, AZ 85281
Email: melih@asu.edu

Andréa W. Richa
Department of Computer Science and Engineering
Arizona State University
Tempe, AZ 85281
Email: aricha@asu.edu

Abstract

Designing an overlay network for publish/subscribe communication in a system where nodes may subscribe to many different topics of interest is of fundamental importance. For scalability and efficiency, it is important to keep the degree of the nodes in the publish/subscribe system low. It is only natural then to formalize the following problem: Given a collection of nodes and their topic subscriptions connect the nodes into a graph which has low average and maximum degree and in such a way that for each topic t, the graph induced by the nodes interested in t is connected. We present the first polynomial time parameterized sublinear approximation algorithm for this problem.

We also considered a variation of the problem which enforces that each topic-connected overlay network be of constant diameter, while keeping the average degree low. We present two heuristics for this problem which guarantees that each topic-connected overlay network will be of diameter 2 and which aims at keeping the overall average node degree low. Our experimental results validate our algorithms showing that our algorithms are able to achieve very low diameter and improve the previous algorithm.

Keywords: publish/subscribe,peer-to-peer, overlay, optimization

1 Introduction

In the publish/subscribe (pub/sub) communication paradigm, publishers and subscribers interact in a decoupled fashion. Publishers publish their messages through logical channels and subscribers receive the messages they are interested in by subscribing to the appropriate services, which deliver messages through these channels.

A pub/sub system may be *topic-based*, if messages are published to "topics", where each topic is uniquely associated with a logical channel. Subscribers in a topic-based system will receive all messages published to the topics to which they subscribe. The publisher is responsible for defining the classes of messages to which subscribers can subscribe. In a *content-based* system, messages are only delivered to a subscriber if the attributes of those messages match constraints defined by the subscriber; each logical channel is characterized by a subset of these attributes. The subscriber is responsible for classifying the messages.

Pub/sub communication systems are scalable and simple to implement (see e.g., [1–4, 6–10, 15, 17, 21–25]). Hence there are many applications which are built on top of such systems, most notably a plethora of Internet-based applications, such as stock-market monitoring engines, RSS [24] feeds [18], on-line gaming and several others. For a survey on pub/sub systems, see [14].

In this paper, we will design a (peer-to-peer) overlay network for each pub/sub topic, in the sense that for each topic t, the subgraph induced by the nodes interested in t will be connected. This translates into a *fully decentralized* topic-based pub/sub system since any given topic-based overlay network will be connected and thus nodes subscribed to a given topic do not need to rely on other nodes (agents) for forwarding their messages. Such an overlay network is called *topic-connected*.

We can evaluate the complexity of a pub/sub overlay network in terms of the cost of topic-based broadcasts on the network. As in many other systems, a space-time trade-off exists: On one hand, one would like the total time taken by the broadcast (which directly depends on the diameter of each topic-based subnetwork) to be as small as possible; on the other hand, for memory and node bandwidth considerations, one would like to keep the total degree of a node small. Those two measures are often conflicting. For example, take the simple scenario where all nodes are subscribed to the same topic: A star overlay would result in the best possible diameter but worst possible degree for the nodes. Even if we were to maintain a balanced structure (e.g., a balanced binary tree) for each topic, it is not clear how to achieve that without letting the node degrees grow as large as the sizes of the node subscription sets.

Some of the current solutions adopted in practice actually fail at maintaining *both* the diameter and the node degrees low. A naive, albeit popular, solution to topic connected-overlay network design is to construct a cycle (or a tree or any other separate overlay structure) connecting all nodes interested in a topic independently for each given topic [25]: This construction may result in a network with node degrees proportional to the nodes' subscription sizes, whereas a more careful construction, taking into account the correlations among the node subscription sets might result in much smaller node degrees (and total number of edges).

Low node degrees are desirable in practice for scalability and also due to bandwidth constraints. Nodes with a high number of adjacent links will have to manage all these links (e.g., monitor the availability of its neighbors, incurring in heartbeats and keep-alive state costs, and connection state costs in TCP) and the traffic going through each of the links, without being able to take great advantage of aggregating the traffic (which would also reduce the number of packet headers, which can be responsible for a significant portion of the traffic for small messages). See [11, 19] for further motivation.

The node degrees and number of edges required by a topic-connected overlay network will be low if the node subscriptions are well-correlated. In this case, by connecting two nodes with many coincident topics, one can satisfy connectivity of many topics for those two nodes with just one edge. Several recent empirical studies suggest that correlated workloads are indeed common in practice [18, 24].

	Chockler et. al. [11]	Onus and Richa [19]	This Paper	Lower Bound
Average Degree	$O(\log(n*t))$	$\theta(n)$ [proved in this paper]	$O(k*\log(n*t))$	$\Omega(c)$
Maximum Degree	$\theta(n)$	$O(\log(n*t))$	$O((n/k)*\log(n*t))$	$\Omega(c)$

Table 1: Summary of known results on overlay network construction for publish/subscribe communication (n: number of nodes, t: number of topics, c: constant, k is any parameter between 1 and n)

In this work, we first consider the problem of devising topic-based pub/sub overlay networks with low node degrees. More specifically, we consider the following problem:

Low Degree Topic-Connected Overlay (Low-TCO) Problem: Given a collection of nodes V, a set of topics T, and the node interest assignment I, connect the nodes in V into a topic-connected overlay network G which has low average and maximum degree.

We present a parameterized sublinear approximation algorithm for this problem which approximates both the average and the maximum degree well (See Table 1).

We also considered the CD-TCO problem which is introduced in [19]. The CD-TCO problem aims at minimizing the average degree and enforces each topic connected network to be of constant diameter (see Section A.2). We present two heuristics for this problem which guarantees that each topic's induced overlay subnetwork will be of diameter 2 and which aims at keeping the average node degree of the overall topic-connected overlay network low. Our experimental results validate our algorithms showing that our algorithms are able to achieve very low diameter and improve the previous algorithm by factor $\%20$ on average degree.

1.1 Related Work

Chockler et al. [11] introduced the *MinAvg-TCO* problem. This problem aims at *minimizing the average* degree of the overlay network. They present an algorithm, called GM, which achieves a logarithmic approximation on the minimum average degree of the overlay network. We start with a formal definition of the MinAvg-TCO problem and GM algorithm.

Minimum Topic Connected Overlay Problem (MinAvg-TCO) [11]: Given a collection of nodes V, a set of topics T, and a node interest assignment I, connect the nodes in V into a topic-connected overlay network G which has the least possible total number of edges (and hence the least possible average node degree).
The Greedy Merge (GM) Algorithm [11]: Initially we a have the set of nodes V and no edges between the nodes. At each step, add the edge which maximally reduces the total number of topic-connected components.

While minimizing the average degree is a step forward towards improving the scalability and practicality of the pub/sub system, their algorithm may still produce overlay networks of very uneven node degrees where the maximum degree may be unnecessarily high. In [19], it is shown that GM algorithm may produce a network with maximum degree $|V|$ while a topic-connected overlay network of constant degree exists for the same configuration of I (See Table 1). The *MinMax-TCO* problem is introduced in [19]. This problem aims at *minimizing the maximum* degree of the overlay network. An algorithm, MinMax-ODA, is presented which achieves a logarithmic approximation on the minimum maximum degree of the overlay network. MinMax-ODA algorithm may produce overlay networks of very high average degree: As we will show in Section 3, this algorithm may produce a network with average degree $|V|-2$ while a topic-connected overlay network of constant average degree exists for the same configuration of I (See Table 1). Some of the high level ideas and proof techniques of [11, 19] have their roots in techniques used for the classical Set-Cover problem. We benefit from some of the ideas in [11, 19] and also build upon the constructions for Set-Cover, extending and modifying them to be able to handle the maximum degree and the average degree together.

To the best of our knowledge, minimizing max-degree and avg-degree together in topic-connected pub/sub overlay network design had not been directly addressed prior to this work. The overlay networks resulting from [2, 5, 10] are not required to be topic-connected. In [4, 9, 12, 25], topic-connected overlay networks are constructed, but they make no attempt to minimize the average or maximum node degree. The first papers to directly consider node degrees when building topic-connected pub/sub systems were [11] and [19], as we mentioned above. Minimizing the diameter in topic-connected pub/sub overlay network design first addressed in [19].

1.2 Our Contributions

Our main contribution in this paper is the formal design and analysis of the topic-connected overlay design algorithm (Low-ODA) which is a parameterized sublinear approximation algorithm for the Low-TCO problem. Low-ODA algorithm approximates both the average degree and the maximum degree well (See Table 1). The Low-ODA algorithm is a greedy algorithm which relies on repeatedly using a greedy approach for finding edges. No previous algorithm with sublinear approximation guarantees on both the average and maximum degree of a topic-connected pub/sub overlay network was known prior to this work.

In addition, we present two algorithms, CD-ODA-I and CD-ODA-II, which builds a topic-based pub/sub network, where each topic-connected component is guaranteed to be of constant diameter — more specifically of diameter 2 —and where we aim at keeping the average degree low. While we do not have a formal proof on any approximation guarantees on the average node degree, we validate the performance of CD-ODA-I and CD-ODA-II with experimental results.

1.3 Structure of the paper

In Section 2, we present some definitions and restate the formal problem definition. In Section 3, we present an outline of the related problem of minimizing the maximum node degree, namely the MinMax-TCO problem, and the corresponding logarithmic approximation algorithm MinMax-ODA proposed by Onus and Richa [19], since some of the ideas presented will be useful for the Low-TCO problem. Section 4 presents our topic-connected overlay design algorithm Low-ODA, whose approximation ratio is proved in Section 5. Section A.2 presents our two new algorithms for the the CD-TCO problem. We conclude the paper, also presenting some future work, in Section 7.

2 Preliminaries

Let V be the set of nodes, and T be the set of topics. Let $n = |V|$. The interest function I is defined as $I : V \times T \rightarrow \{0, 1\}$. For a node $v \in V$ and topic $t \in T$, $I(v, t) = 1$ if and only if node v is subscribed to topic t, and $I(v, t) = 0$ otherwise.

For a set of nodes V, an overlay network $G(V, E)$ is an undirected graph on the node set V with edge set $E \subseteq V \times V$. For a topic $t \in T$, let $V_t = \{v \in V | I(v, t) = 1\}$. Given a topic $t \in T$ and an overlay network $G(V, E)$, the number of topic-connected components of G for topic t is equal to the number of connected components of the subgraph of G induced by V_t. An overlay network G is *topic-connected* if and only if it has one topic-connected component for each topic $t \in T$. The diameter of a graph is the length of the longest shortest path in the graph. The degree of a node v in an overlay network $G(V, E)$ is equal to the total number of edges adjacent to v in G.

Low Degree Topic-Connected Overlay (Low-TCO) Problem: Given a collection of nodes V, a set of topics T, and the node interest assignment I, connect the nodes in V into a topic-connected overlay network G which has low average and maximum degree.

3 MinMax-TCO problem and Min-Max Overlay Design Algorithm (MinMax-ODA)

The MinMax-TCO problem was introduced by Onus and Richa [19] in which they aim at minimizing the maximum node degree. In this section we present a formal definition of the MinMax-TCO problem and outline the main techniques in the corresponding Min-Max Overlay Design Algorithm (MinMax-ODA), which will be useful for our approach to Low-TCO. We start with a formal definition of the MinMax-TCO problem and MinMax-ODA algorithm.

The Minimum Maximum Degree Topic-Connected Overlay (MinMax-TCO) Problem [19]: Given a collection of nodes V, a set of topics T, and the node interest assignment I, connect the nodes in V into a topic-connected overlay network G which has least possible maximum degree.

The Minimum Maximum Overlay Design Algorithm(MinMax-ODA) [19]: Initially we a have the set of nodes V and no edges between the nodes. At each step, add the edge which maximally reduces the total number of topic-connected components among the edges which increases maximum degree of the current graph minimally.

The MinMax-ODA algorithm does not work well for the Low-TCO problem: The approximation ratio on the average degree obtained by the MinMax-ODA algorithm may be as bad as $\Theta(n)$, as we show in the lemma below.

Lemma 3.1. *The MinMax-ODA algorithm can only guarantee an approximation ratio of at least $\Theta(n)$ on the average degree for the Low-TCO problem, where n is number of nodes in the pub/sub system.*

Proof. Consider the example where we have n nodes $v_1, v_2, ..., v_n$, and n^2 topics $T = \{t_{i,j} | 1 \leq i, j \leq n\}$. Node v_1 is interested in all topics in T and each v_i is interested in $t_{i,j}$ and $t_{j,i}$, $2 \leq i \leq n$, $1 \leq j \leq n$. The MinMax-ODA algorithm will produce an overlay network with $n(n-2)/2 + 1$ edges. The optimal overlay network with minimum number of edges is $E = \{(v_1, v_i) | 1 < i \leq n\}$, – the number of edges of this overlay network is $n - 1$. Hence the approximation ratio of the MinMax-ODA algorithm can be as large as $(n(n-2)/2 + 1)/(n-1) = \Theta(n)$. □

4 Low Degree Overlay Design Algorithm (Low-ODA)

In this section we present our overlay design algorithm (Low-ODA) for the Low-TCO problem. Let $1 \leq k \leq n$. Low-ODA starts with the overlay network $G(V, \emptyset)$. At each iteration of Low-ODA, the algorithm considers two edges:

e_1 : a maximum weight edge — where the weight of an edge (u, v) is given by the reduction on the number of topic-connected components which would result from the addition of (u, v) to the current overlay network — among the ones which minimally increases maximum degree of the current graph

e_2 : a maximum weight edge

If weight of edge e_1 is greater than weight of e_2 over k, edge e_1 is added to edge set of the overlay network; otherwise edge e_2 is added.

Let $NC(V, E)$ denote total number of topic connected components in the overlay network given by (V, E).

Algorithm 1 Low Degree Overlay Design Algorithm (Low-ODA)

```
 1: OverlayEdges ← $\emptyset$
 2: $V$ ← Set of all nodes
 3: $G'(V, E')$ ← Complete graph on $V$
 4: for $\{u, v\} \in E'$ do
 5:     $w\{u, v\}$ ← Number of topics that both of nodes $u$ and $v$ have
 6: end for
 7: while $G(V$,OverlayEdges$)$ is not topic-connected do
 8:     Find maximum-weighted edge $e_1$ on $G'(V, E', w)$ among the ones which increase the maximum
        degree of $G(V$,OverlayEdges$)$ minimally.
 9:     Find maximum-weighted edge $e_2$ on $G'(V, E', w)$
10:     if $w(e_1) \geq w(e_2)/k$ then
11:         $e = e_1$
12:     else
13:         $e = e_2$
14:     end if
15:     OverlayEdges = OverlayEdges $\bigcup e$
16:     $E' \leftarrow E' - e$
17:     for $\{u, v\} \in E'$ do
18:         $w\{u, v\}$ ← NC($V$, OverlayEdges) - NC($V$,OverlayEdges $\bigcup \{u, v\}$ )
19:     end for
20: end while
```

Steps 1-6 of Low-ODA build an initial weighted graph $G'(V, E', w)$ on V, where $E' = V \times V$ and $w(\{u, v\})$ is equal to the amount of decrease in the number of topic-connected components resulting from the addition of the edge (u, v) to the current overlay network (represented by the edges in OverlayEdges). Initially, this amount will be equal to the number of topics that nodes u and v have in common.

At each iteration of the while loop, two edges are considered: an edge (e_1) with maximum weight among the ones which increase the maximum degree of the current graph minimally and an edge (e_2) with maximum weight. If weight of the first one (e_1) is greater than or equal to weight of the second one (e_2) over k, e_1 is added to the set of overlay edges; otherwise e_2 is added. Note that the addition of an edge to OverlayEdges can either increase the maximum degree by 1 or not increase it at all. The crux in the analysis of this algorithm is to show that each of the edges will reduce the number of connected components by a "large" amount without increasing max degree too much.

Before we proceed in proving the approximation ratio on the maximum degree and the approximation ratio on the average degree guaranteed by Low-ODA, we prove that the algorithm terminates in $O(|V|^4|T|)$ time.

Lemma 4.1. *The Low-ODA algorithm terminates within $O(|V|^2)$ iterations on the while loop.*

Proof. At each iteration of the while loop, at least one edge is added to the current overlay network. Hence the algorithm will terminate in at most $O(|V|^2)$ iterations. □

Lemma 4.2. *The running time of Low-ODA is $O(|V|^4|T|)$.*

Proof. The weight initialization takes $O(|V|^2|T|)$ time. Updating the weight of each of the remaining edges takes $O(1)$ time([11], Lemma 6.4). Finding the edge with max weight will take at most $O(|V|^2)$ time. Since total weight of the edges is $O(|V|^2|T|)$ at the beginning and greater than 0 at the end, Low-ODA takes $O(|V|^2|T|) * O(|V|^2) = O(|V|^4|T|)$ time. ☐

5 Approximation Ratio

In this section, we will prove that our overlay design algorithm (Low-ODA) approximates the average degree by factor $O(k * \log(\sum_{v \in V} |\{t \in T | I(v,t) = 1\}|))$ and the maximum degree by factor $O((n/k) * \log(\sum_{v \in V} |\{t \in T | I(v,t) = 1\}|))$.

Theorem 5.1. *The overlay network output by Low-ODA has average node degree within a factor of* $O(k * \log(\sum_{v \in V} |\{t \in T | I(v,t) = 1\}|))$ *from the minimum possible average node degree for any topic-connected overlay network on* V.

Proof. The proof follows the general lines as the proof of the logarithmic approximation ratio for the classic set cover problem (which was also the basis for the approximation ratio proof of the GM algorithm for the MinAv-TCO problem [11] and the MinMax-ODA algorithm for the MinMax-TCO problem [19]). Assume we have an instance of the Low-TCO problem and that $G(V, E_{opt})$ is a solution for this instance with minimum number of edges. Let $|E_{opt}| = m$. At the beginning of the algorithm, the total number of connected components is $C_{start} = \sum_{v \in V} |\{t \in T | I(v,t) = 1\}|$ and at the end $C_{end} = |\{t | t \in T$ and $\exists v \in V$ such that $I(v,t) = 1\}|$. Note that since we count the connected components for each topic separately, once we get down to C_{end} components, there must exist *exactly one* component for each active topic t (i.e., each t such that there exists some v with $I(v,t) = 1$) — i.e., the overlay network is topic-connected.

Let e_i be the i^{th} edge added to the set by the algorithm Low-ODA. Let n_i be total number of connected components before we add i^{th} edge, so $n_1 = C_{start}$. Let $S_i = \{e_1, e_2, ..., e_{i-1}\}$ be the set of all edges found before the algorithm starts adding the i-th edge.Before Low-ODA starts adding the i^{th} edge, we have n_i components and we know that if we add all the edges $E_{opt} - S_i$, to the current solution, the total number of connected components will be reduced to C_{end}. Since $|E_{opt} - S_i| \leq m$, there exists an edge which decreases the total number of connected components by at least $(n_i - C_{end})/m$. Since our algorithm always adds at least a $(1/k)$-optimal edge, the edge e_i that our algorithm adds must decrease the total number of connected components at that time by at least $(1/k)$ of this amount. Therefore,

$$n_i - n_{i+1} \geq (n_i - C_{end})/(m * k) \quad \Rightarrow \quad n_{i+1} - C_{end} \leq (1 - 1/(m * k))(n_i - C_{end}).$$

Hence, the number of iterations for our algorithm Low-ODA is less than or equal to the smallest z which satisfies

$$1 > (n_1 - C_{end})(1 - 1/(m * k))^z \quad \Rightarrow \quad z \leq m * k ln(C_{start} - C_{end}) \quad \Rightarrow \quad z \leq m * k ln(C_{start}).$$

☐

Theorem 5.2. *The overlay network output by Low-ODA has maximum node degree within a factor of* $O((n/k) * \log(\sum_{v \in V} |\{t \in T | I(v,t) = 1\}|))$ *from the minimum possible maximum node degree for any topic-connected overlay network on* V.

Proof. The proof is in Appendix A.1. ☐

6 Constructing Constant Diameter Overlays for Publish-Subscribe

In this section, we study the optimization problem that constructs a constant diameter overlay network for publish/subscribe communication with many topics. This problem is proposed in [19]. We present two new overlay network construction heuristics that guarantees constant diameter and topic-connectivity which are most important factors for efficient routing. The formal problem is as follows:

Constant Diameter Topic-Connected Overlay (CD-TCO) Problem: [19] Given a collection of nodes V, a set of topics T, and the node interest assignment I, connect the nodes in V into a topic-connected overlay network G which has least possible average degree and constant diameter.

In [19], a heuristic (CD-ODA) is presented for this problem. CD-ODA starts with the overlay network $G(V, \emptyset)$. At each iteration of the CD-ODA, a node which has maximum number neighbors with non-empty interest intersection is chosen. Number of neighbors is equal to $n_u = |\{v \in V | \exists t \in T, Int(v,t) = Int(u,t) = 1\}|$. After that, an edge between this node and each of its neighbors is added and the topics in this node's interest assignment is removed from the set of topics.

We present two new heuristics for this problem and validate our heuristic via experimental results. Experimental results show our heuristics improves CD-ODA [19] by factor $\%20$. We first present examples to give the intuition for our algorithms. And then, our algorithms and analysis are presented. This section ends with experimental results for the algorithms.

6.1 Constant Diameter Overlay Design Algorithm I(CD-ODA-I)

In this section, we present CD-ODA-I. CD-ODA-I starts with the overlay network $G(V, \emptyset)$. At each iteration of the CD-ODA-I, a node which has maximum number of weighted neighbors is chosen. Number of weighted neighbors for node u is equal to

$$w_u = \sum_{t \in T} |\{v \in V | Int(v,t) = Int(u,t) = 1\}|.$$

And then, we put an edge between this node and each of its neighbors. And we remove the topics in this node's interest assignment from the set of topics.

Algorithm 2 Constant Diameter Overlay Design Algorithm I(CD-ODA-I)

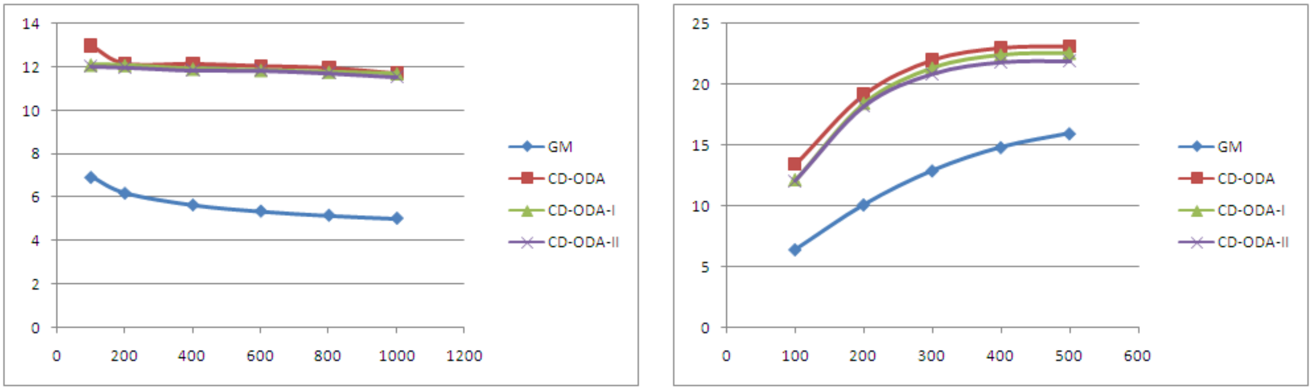
```
$T \leftarrow$ Set of all topics
while $T$ is not empty do
    For each node $u$, calculate total number of weighted neighbors. Denote this number by $w_u$.
    Find node $u$ with maximum $w_u$.
    Put an edge between $u$ and all nodes $v$ such that there exists a topic $t \in T$ and $Int(u,t) = Int(v,t) = 1$.
    Remove all topics $t$ from $T$ such that $Int(u,t) = 1$.
end while
```

With using the examples I, II and III in Figures 3(a), 3(b) and 4(b) and intuitions A.4, A.5 and A.6, we designed CD-ODA-I algorithm. CD-ODA-I finds the optimal solution for Example III in Figure 4(b).

6.2 Constant Diameter Overlay Design Algorithm II(CD-ODA-II)

In this section, we present CD-ODA-II. CD-ODA-II starts with the overlay network $G(V, \emptyset)$. At each iteration of the CD-ODA-II, a node u which has maximum connection density, d_u, is chosen. Connection density is equal to

$$d_u = \frac{\sum_{t \in T} |\{v \in V | Int(v,t) = Int(u,t) = 1\}|}{|\{v \in V | \exists t \in T, Int(v,t) = Int(u,t) = 1\}|}.$$



(a) (b)

Figure 1: Average node degree for GM, CD-ODA, CD-ODA-I and CD-ODA-II

Note that $d_u = \frac{w_u}{n_u}$. And then, we put an edge between this node and each of its neighbors. And we remove the topics in this node's interest assignment from the set of topics.

Algorithm 3 Constant Diameter Overlay Design Algorithm II(CD-ODA-II)

```
$T \leftarrow$ Set of all topics
while $T$ is not empty do
    For each node $u$, calculate connection density. Denote this number by $d_u$.
    Find node $u$ with maximum $d_u$.
    Put an edge between $u$ and all nodes $v$ such that there exists a topic $t \in T$ and $Int(u,t) = Int(v,t) = 1$.
    Remove all topics $t$ from $T$ such that $Int(u,t) = 1$.
end while
```

With using the example in Figure 5(a) and intuition A.7, we designed the CD-ODA-II algorithm. CD-ODA-II finds the optimal solution for Example IV in Figure 5(b).

6.3 Analysis of Algorithms

Lemma 6.1. *Both algorithms CD-ODA-I and CD-ODA-II terminates within* $O(|V|^2 * |T|)$ *time.*

Lemma 6.2. *Both algorithms CD-ODA-I and CD-ODA-II generate a 2-diameter overlay for each topic.*

Proof. Since algorithms generate a star for each topic, each topic overlay network will have diameter 2. □

6.4 Experimental Results

The GM algorithm [11], the CD-ODA algorithm [19] and our algorithms are implemented in Java. These algorithms are compared according to the average degree in the resulting graph. The diameter is always 2 for our algorithms and for CD-ODA and it may be $\theta(n)$, n : number of nodes, for GM algorithm. When we compare the results of GM, CD-ODA and our algorithms according to average degree, our algorithms and CD-ODA require at most 2.3 times more edges than GM. Our algorithms improve CD-ODA [19] by factor $\%20$.

6.4.1 Average Node Degree with Varying Number of Nodes

For the first experiment, the number of nodes varies between 100 to 1000. The number of topics is 100. We fixed number of subscriptions to $s = 10$. Each node is interested in each topic uniformly at random. This experimental setting is similar to previous studies [11, 19, 21].

Figure 1(a) is a comparison of GM, CD-ODA and our algorithms according to the average degree. The average degree of the graph decreases for GM algorithm when the number of nodes increases since GM algorithm can find edges with higher correlation as the number of nodes increases. The average degree of the graph slightly decreases for our algorithms and CD-ODA algorithm. When we compare the results of GM and our algorithm CD-ODA-II, our algorithm requires at most 2.3 times more edges than GM (Figure 1(a)). CD-ODA-II improves CD-ODA by factor $\%3$ on average and CD-ODA-I by factor $\%2$ on average.

6.4.2 Average Node Degree with Varying Number of Topics

For the second experiment, the number of nodes is 100. The number of topics varies between 100 and 500. We fixed number of subscriptions to $s = 20$. Each node is interested in each topic uniformly at random. This experimental setting is similar to previous studies [11, 21].

Figure 1(b) is a comparison of GM, CD-ODA and our algorithms according to the average degree. The average degree of the graph increases for our algorithms, GM algorithm and CD-ODA algorithm when the number of topics increases since algorithms can only find edges with lower correlation as the number of topics increases. When we compare the results of GM and our algorithm CD-ODA-II, our algorithm requires at most 1.9 times more edges than GM (Figure 1(b)). CD-ODA-II improves CD-ODA by factor $\%6$ on average and CD-ODA-I by factor $\%2$ on average.

6.4.3 Average Node Degree with Varying Subscription Size

For the third experiment, the number of nodes and the number of topics are fixed to 100. The subscription size varies between 10 to 50. Each node is interested in each topic uniformly randomly. This experimental setting is similar to previous studies [11, 21].

Figure 2 is a comparison of GM, CD-ODA and our algorithms according to the average degree. The average degree of the overlay network decreases for both GM, CD-ODA and our algorithms when subscription size increases since algorithms can find edges with higher correlation. When we compare the results of GM and our algorithm CD-ODA-II, our algorithm requires at most 1.8 times more edges than GM (Figure 2). CD-ODA-II improves CD-ODA by factor $\%20$ on average and CD-ODA-I by factor $\%1$ on average.

7 Conclusions

In this paper, we study a new optimization problem (Low-TCO) that constructs a practical and scalable overlay network for publish/subscribe communication with many topics. We present a topic-connected overlay network design algorithm (Low-ODA) which approximates both average degree and maximum degree well.

We present two heuristics for constructing constant diameter overlay networks. Our experimental results show that our algorithms improve the previous algorithm by factor $\%20$ on average degree.

As future work, we would like to build upon our CD-ODA-I and CD-ODA-II algorithms, by formally and experimentally evaluating the hardness of obtaining a topic-connected overlay design algorithm which achieves a “good” trade-off between low diameter and low node degree. This basically amounts to a bi-criteria optimization problem and we have to be able to “quantify” the relative importance of optimizing

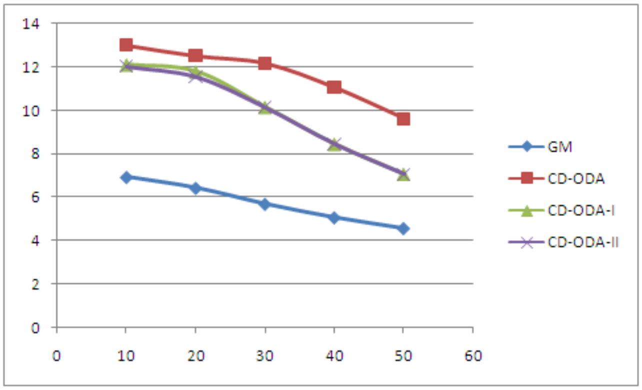


Figure 2: Average node degree for GM, CD-ODA, CD-ODA-I and CD-ODA-II

over these two parameters (e.g., in the CD-ODA-I algorithm and the CD-ODA-II algorithm we restrict our attention to networks of diameter 2, while aiming at maintaining the average degree low).

Two other important lines for future work would be to design efficient *distributed* algorithms for the Low-TCO problem, and to look at this problem under the line of a dynamic configuration of the node set V and the interest assignment I.

References

[1] *Oracle9i Application Developers Guide Advanced Queuing*, Oracle, Redwood Shores, CA.

[2] E. Anceaume, M. Gradinariu, A. K. Datta, G. Simon, and A. Virgillito, *A semantic overlay for self-peer-to-peer publish/subscribe*, In ICDCS, 2006.

[3] S. Baehni, P. T. Eugster, and E. Guerraoui, *Data-aware multicast.* In DSN, 2004.

[4] R. Baldoni, R. Beraldi, V. Quema, L. Querzoni, and S. T. Piergiovanni, *TERA: Topic-based Event Routing for Peer-to-Peer Architectures*, 1^{st} International Conference on Distributed Event-Based Systems (DEBS). ACM, 6 2007.

[5] R. Baldoni, R. Beraldi, L. Querzoni, and A. Virgillito, *Efficient publish/subscribe through a self-organizing broker overlay and its application to SIENA*, The Computer Journal, 2007.

[6] S. Banerjee, B. Bhattacharjee, and C. Kommareddy, *Scalable application layer multicast*, SIGCOMM Comput. Commun. Rev, 32(4):205-217, 2002.

[7] S. Bhola, R. Strom, S. Bagchi, Y. Zhao, and J. Auerbach, *Exactly-once delivery in a content-based publish-subscribe system.* In DSN, 2002.

[8] A. Carzaniga, M. J. Rutherford, and A. L. Wolf, *A routing scheme for content-based networking.*, IEEE INFOCOM 2004, Hon Kong, China, Mar. 2004.

[9] M. Castro, P. Druschel, A. M. Kermarrec, and A. Rowstron, *SCRIBE: a large-scale and decentralized application-level multicast infrastructure*, IEEE J. Selected Areas in Comm. (JSAC), 20(8):14891499, 2002.

[10] R. Chand and P. Felber, *Semantic peer-to-peer overlays for publish/subscribe networks*, In Euro-Par 2005 Parallel Processing, Lecture Notes in Computer Science, volume 3648, pages 11941204. Springer Verlag, 2005.

[11] G. Chockler, R. Melamed, Y. Tock and R. Vitenberg, *Constructing scalable overlays for pub-sub with many topics*, Proc. of the 26^{th} ACM Symp. on Principles of Distributed Computing (PODC), 2007, pp. 109–118.

[12] G. Chockler, R. Melamed, Y. Tock, and R. Vitenberg, *SpiderCast: A Scalable Interest-Aware Overlay for Topic-Based Pub/Sub Communication*, 1^{st} International Conference on Distributed Event-Based Systems (DEBS). ACM, 6 2007.

[13] R. Diestel, *Graph Theory*, Springer-Verlag, 2^{nd} edition, New York, 2000.

[14] P. T. Eugster, P. A. Felber, R. Guerraoui, and A. M. Kermarrec. *The many faces of publish/subscribe*. ACM Computing Surveys, 35(2):114131, 2003.

[15] R. Guerraoui, S. Handurukande, and A. M. Kermarrec, *Gossip: a gossip-based structured overlay network for efficient content-based filtering*, Technical Report IC/2004/95, EPFL, Lausanne, 2004.

[16] B. Korte, J. Vygen, *Combinatorial Optimization Theory and Algorithms*, Springer-Verlag, 2^{nd} edition, 2000.

[17] R. Levis, *Advanced Massaging Applications with MSMQ and MQSeries.* QUE, 1999.

[18] H. Liu, V. Ramasubramanian, and E. G. Sirer. *Client behavior and feed characteristics of rss, a publish-subscribe system for web micronews*. In Internet Measurement Conference (IMC), Berkeley, California, October 2005.

[19] M. Onus, A. W. Richa, *Minimum Maximum Degree Publish-Subscribe Overlay Network Design*, 28^{th} Annual IEEE Conference on Computer Communications (INFOCOM), Rio de Janeiro, Brazil, April 2009.

[20] M. Onus, A. W. Richa, *Publish-Subscribe Overlay Network Design*, Technical Report, Arizona State University, Department of Computer Science and Engineering, TR-09-005, 2009, available at http://sci.asu.edu/news/technical/index.php.

[21] V. Ramasubramanian, R. Peterson, and E. G. Sirer. *Corona: A High Performance publish-subscribe system for the world wide web.*, In NDSI, 2006.

[22] D. Sandler, A. Mislove, A. Post and P. Druschel. *FeedTree: Sharing web micronews with peer-to-peer event notification.*, In International Workshop on Peer-to-Peer Systems(IPTPS), 2005.

[23] D. Tam, R. Azimi, and H.-A Jacobsen. *Building content-based publish/subscribe systems with distributed hash tables.*, In 1^{st} International Workshop on Databases, Information Systems, and P2P Computing(DBISP2P), Berlin, Germany, 2003.

[24] Y. Tock, N. Naaman, A. Harpaz and G. Gershindky. *Hierarchical clustering of message flows in a multicast data dissemination system.*, In 17^{th} IASTED International Conferance Parallel and Distributed Computing and Systems, pages 320-327, 2005.

[25] S. Voulgaris, E. Riviere, A. M. Kermarrec, and M. van Steen, *Sub-2-sub: Self-organizing content-based publish subscribe for dynamic large scale collaborative networks*, In IPTPS, 2006.

A Appendix

A.1 Approximation Ratio

Theorem A.1. *The overlay network output by Low-ODA has maximum node degree within a factor of* $O((n/k) * \log(\sum_{v \in V} |\{t \in T | I(v,t) = 1\}|))$ *from the minimum possible maximum node degree for any topic-connected overlay network on* V.

Proof. At a high level, the proof follows the general lines as the proof of the logarithmic approximation ratio for the classic set cover problem (which was also the basis for the approximation ratio proof of the GM algorithm for the MinAv-TCO problem [11] and the MinMax-ODA algorithm for the MinMax-TCO problem [19]).

However, before we can apply the set cover framework, we first need to carefully show that Low-ODA works as if we had many applications of a greedy matching algorithm that aims at reducing the number of connected components maximally and then relate our network overlay construction to a matching decomposition of an optimal (i.e., a minimum maximum degree) overlay network.

Assume we have an instance of the Low-TCO problem and that $G(V, E_{opt})$ is a solution with minimum possible maximum degree for this instance. Let this maximum degree is d_{opt}. We will use the following well-known result in graph theory for the proof. This result is presented and proved in [19].

Lemma A.2 ((Lemma 4 in [19])). *Given a graph* $G(V, E)$ *with maximum degree* d, *we can divide the edge set* E *into* $d + 1$ *matchings* M_i, $1 \le i \le (d + 1)$.

Using the lemma above, we can divide the edge set E_{opt} of the optimum solution into $d_{opt} + 1$ matchings M_i, $1 \le i \le (d_{opt} + 1)$.

At the beginning of the algorithm, the total number of connected components is $C_{start} = \sum_{v \in V} |\{t \in T | I(v,t) = 1\}|$ and at the end $C_{end} = |\{t | t \in T \text{ and } \exists v \in V \text{ such that } I(v,t) = 1\}|$. Note that since we count the connected components for each topic separately, once we get down to C_{end} components, there must exist *exactly one* component for each active topic t (i.e., each t such that there exists some v with $I(v,t) = 1$) — i.e., the overlay network is topic-connected.

At each iteration of the while loop, a maximum weight edge among the ones which increases the maximum degree of the current graph minimally or a maximum weight edges is added to the set of overlay edges. At start all nodes have degree 0.

After a number of iterations, the weight of a maximum weight edge among the ones which increases the maximum degree of the current graph minimally will be less than the weight of a maximum weight edge over k and we will add this maximum weight edge to the graph and this edge will increase the maximum degree of the graph by 1.

Let S_i be the edge set which contains the edge added by the algorithm Low-ODA which makes maximum degree of the graph i and the edges added while maximum degree of the graph is i and it does not contain the edge which makes maximum degree of the graph $i + 1$. Let $h = n/(2k) + 1$. Let $R_i = S_{h*(i-1)+1} \bigcup S_{h*(i-1)+2} \bigcup ... \bigcup S_{h*(i-1)+h}$. Let $RA_i = R_1 \bigcup R_2 \bigcup ... \bigcup R_{i-1}$ be the union of all edges added before the algorithm starts adding the set R_i. Let n_i be total number of connected components before the algorithm adds R_i, so $n_1 = C_{start}$.

The following lemma proves that each set R_i chosen by our algorithm decrease the current total number of connected components at least $(1/3)$ of any optimal matching.

Lemma A.3. *The set* R_i *reduces the total number of connected components of* $G(V, RA_i)$ *by at least* $1/3$ *of any optimal matching which reduces by maximum amount.*

Proof. Let P be the edge set of the matching which reduces the total number of connected components of the $G(V, RA_i)$ by the maximum amount, which we denote by c. Let $Q = \{e_1, e_2, \dots, e_j\}$ be the edge set of R_i. Let $e_l = u_l v_l$ for $1 \le l \le j$. For e_a and e_b, if $a < b$, then e_a is found before e_b by our algorithm. Let Q reduce the total number of connected components of the $G(V, RA_i)$ by c'. Let $G_0 = G(V, RA_i)$ and $G_l = G_{l-1} \bigcup e_l$, for $1 \le l \le j$. Let e_l reduce the total number of connected components of G_{l-1} by y_l. Then,

$$c' = \sum_{1 \le l \le j} y_l \tag{1}$$

Consider the case where e_l does not increase the maximum degree of current graph or $l = 1$ or e_l increases the maximum degree of current graph and there is no possible edge which does not increase maximum degree of the current graph, $1 \leq l \leq j$. Let X_l be the set of edges in P which are incident to u_l or v_l, $1 \leq l \leq j$, and not in $X_{l'}$, $1 \leq l' \leq l-1$. Thus, X_l will have zero or one or two edges for $1 \leq l \leq j$.

Now consider the case where e_l increases the maximum degree of current graph and there are some edges which does not increase maximum degree of the current graph, $2 \leq l \leq j$. Let X_l be the set of the first k maximum weight edges in P which are not in $X_{l'}$, $1 \leq l' \leq l-1$. If there are less than k elements in P which are not in $X_{l'}$, $1 \leq l' \leq l-1$, X_l will only have these edges. If there are edges which are incident to u_l or v_l and not in $X_{l'}$, $1 \leq l' \leq l$, then replace any edges from X_l with those edges (note that there may be at most two edges of this kind).

Let $P_0 = P$ and $P_l = P_{l-1} - X_l$ for $1 \leq l \leq j$. Let X_l reduce the total number of connected components of G_{l-1} by x_l for $1 \leq l \leq j$. Let P_l reduce the total number of connected components of G_l by c_l for $0 \leq l \leq j$.

If there is an edge e_l which increases the maximum degree of current graph and there is no possible edge which does not increase maximum degree of the current graph, $2 \leq l \leq j$, then for each vertex of the graph, there is at least one edge $e_{l'}$ incident to this vertex, $1 \leq l' \leq l-1$. So, union of sets $X_{l'}$, $1 \leq l' \leq l-1$, contains all the edges in P. Thus, $P_j = \emptyset$. Now, consider the case when there is no edge which satisfies these properties (so, when algorithm chooses an edge e_l which increases the maximum degree, there is always an edge which does not increase maximum degree of the current graph). Since R_i contains h sets of $S_{i'}$, there are $h-1 = n/(2k)$ edges which increases maximum degree of the current graph. So (n/2k) of X_l sets has k edges in it (if X_l has less than k edge than all edges of set P are already in one of sets $X_{l'}$, $1 \leq l' \leq l-1$, so $P_j = \emptyset$). Union of sets X_l has at least $(n/2k) * k = n/2$ edges. Since P is a matching, it has at most $n/2$ edges. So, $P_j = \emptyset$. Hence,

$$c_0 = c, c_j = 0 \tag{2}$$

Consider the case where e_l does not increase the maximum degree of current graph or $l = 1$ or e_l increases the maximum degree of current graph and there is no possible edge which does not increase maximum degree of the current graph, $1 \leq l \leq j$. If X_l has two edges, then our algorithm did not choose one of these two edges at that step and choose e_l instead, $0 \leq l \leq j$. Since our algorithm greedily choose the edges, e_l reduces the total number of connected components of G_{l-1} by at least as much as each of the edges in X_l. Hence, $y_l \geq x_l/2$. Similarly, if X_l has one or zero edges, then $y_l \geq x_l$.

Now consider the case where e_l increases the maximum degree of current graph and there are some edges which does not increase maximum degree of the current graph, $2 \leq l \leq j$. X_l has at most k edges. Our algorithm did not choose one of these k edges at that step and choose e_l instead, $0 \leq l \leq j$. Since our algorithm greedily choose the edges, e_l reduces the total number of connected components of G_{l-1} by at least as much as k times any of the edges in X_l. Since X_l has at most k edges, $y_l \geq x_l$. So,

$$y_l \geq \frac{x_l}{2}, 1 \leq l \leq j \quad \Rightarrow \quad \sum_{1 \leq l \leq j} y_l \geq \frac{1}{2} \sum_{1 \leq l \leq j} x_l \tag{3}$$

Since $P_{l+1} = P_l - X_{l+1}$ and $G_{l+1} = G_l \bigcup e_{l+1}$, $0 \leq l \leq j-1$, the amount that P_l reduces the total number of connected components of G_l is smaller than sum of the amount that P_{l+1} reduces the total number of connected components of G_{l+1} and the amount that e_{l+1} reduces the total number of connected components of G_l and the amount X_{l+1} reduces the total number of connected components of G_l. Hence,

$$c_{l+1} \geq c_l - (x_{l+1} + y_{l+1}) \text{ for } 0 \leq l \leq j-1 \tag{4}$$

If we add all the inequalities (2) and (4), we will have

$$\sum_{1 \leq l \leq j} x_l + \sum_{1 \leq l \leq j} y_l \geq c \tag{5}$$

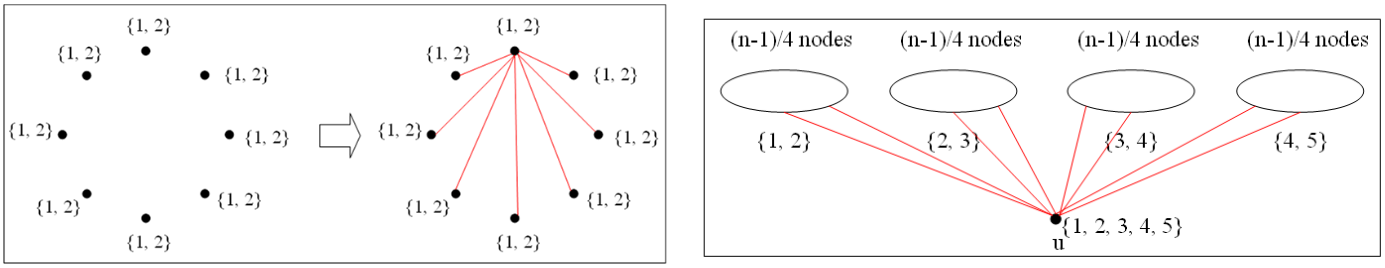
From the inequalities (3) and (5), we will have

$$3 \sum_{1 \leq l \leq j} y_l \geq c \tag{6}$$

From the inequalities (1) and (6), we will have

$$c' \geq c/3$$

□



(a) (b)

Figure 3: (a) Example I (b) Example II

Before Low-ODA starts adding the set R_i, we have n_i components and we know that if we add all the $(d_{opt} + 1)$ matchings $M_j - RA_i$, $1 \leq j \leq (d_{opt} + 1)$, to the current solution, the total number of connected components will be reduced to C_{end}. Therefore, there exists a matching $M_j - RA_i$ which decreases the total number of connected components by at least $(n_i - C_{end})/(d_{opt} + 1)$. Since our algorithm always finds the set R_i which reduces the total number of connected components of $G(V, RA_i)$ by at least $1/3$ of any optimal matching which reduces by maximum amount(Lemma A.3), the set R_i that our algorithm uses must decrease the total number of connected components at that time by at least $(1/3)$ of this amount. Therefore,

$$n_i - n_{i+1} \geq (n_i - C_{end})/(3(d_{opt} + 1)) \quad \Rightarrow \quad n_{i+1} - C_{end} \leq (1 - 1/(3(d_{opt} + 1)))(n_i - C_{end}).$$

Hence, the number of iterations for our algorithm Low-ODA is less than or equal to the smallest m which satisfies $1 > (n_1 - C_{end})(1 - 1/(3(d_{opt} + 1)))^m \;\Rightarrow\; m \leq 3(d_{opt} + 1)ln(C_{start} - C_{end}) \;\Rightarrow\; m \leq 3(d_{opt} + 1)ln(C_{start})$

Since $h = n/(2k) + 1$, the maximum degree of resulting graph is less than or equal to $(n/(2k) + 1) * 3(d_{opt} + 1) * ln(C_{start})$.

□

A.2 Constructing Constant Diameter Overlays for Publish-Subscribe

A.2.1 Examples I and II

In Figure 3(a), each node is subscribed to topics 1 and 2. In an optimal solution, we have $n - 1$ edges and and we have only one star. From this example, we can have the intuition that we should have minimum number of stars.

Intuition A.4. *Construct minimum number of stars.*

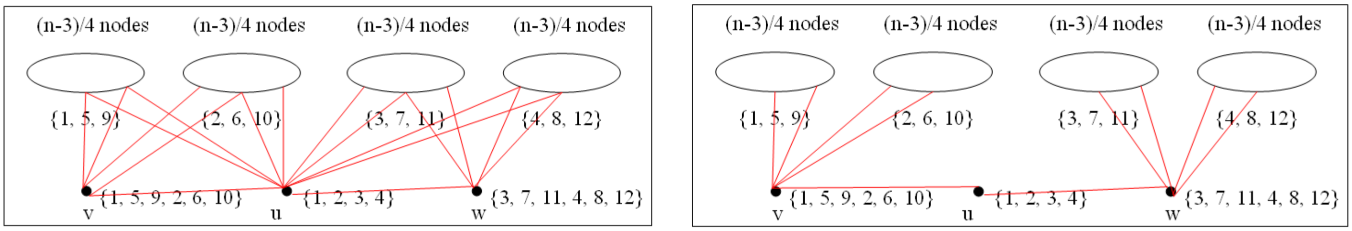
In Figure 3(b), node u is subscribed to topics 1, 2, 3, 4 and 5. $(n - 1)/4$ nodes are subscribed to topics 1 and 2; $(n - 1)/4$ nodes are subscribed to topics 2 and 3; $(n - 1)/4$ nodes are subscribed to topics 3 and 4; $(n - 1)/4$ nodes are subscribed to topics 4 and 5. In the optimal solution, we have $n - 1$ edges and we have only one star with center node u. From this example, we can have the intuition that nodes with many neighbors are good candidates for center of stars.

Intuition A.5. *Nodes with many neighbors are good candidates for center of stars.*

CD-ODA [19] finds the optimal solution for Examples I and II in Figures 3(a), 3(b).

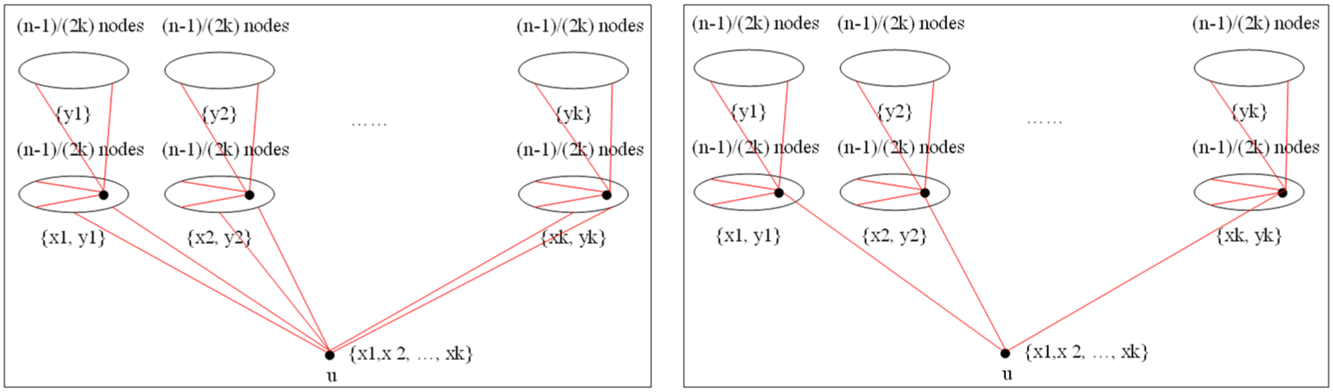
A.2.2 Example III

In Figure 4(a), node u is subscribed to topics 1, 2, 3 and 4; node v is subscribed to topics 1, 5, 9, 2, 6 and 10; and node w is subscribed to topics 3, 7, 11, 4, 8 and 12. $(n - 3)/4$ nodes are subscribed to topics 1, 5 and 9; $(n - 3)/4$ nodes are subscribed to topics 2, 6 and 10; $(n - 3)/4$ nodes are subscribed to topics 3, 7 and 11; $(n - 3)/4$ nodes are subscribed to topics 4, 8 and 12. For this example, CD-ODA first puts edges between node u and all the other nodes



(a) (b)

Figure 4: (a) CD-ODA on Example III (b) Optimal solution for Example III



(a) (b)

Figure 5: (a) CD-ODA and CD-ODA-I on Example IV (b) Optimal solution for Example IV

and then it puts edges between node v and first two sets of $(n-3)/4$ nodes and it puts edges between node w and last two sets of $(n-3)/4$ nodes. Thus, CD-ODA uses $(n-1)+(n-3)/2+(n-3)/2=(2n-4)$ edges.

In the optimal solution (Figure 4(b)), there are edges between node v and first two sets of $(n-3)/4$ nodes and there are edges between node w and last two sets of $(n-3)/4$ nodes and there are edges (u, v) and (u, w). In the optimal solution, only $(n-3)/2+(n-3)/2+2=(n-1)$ edges are required. From this example, we can have the intuition that nodes with many weighted neighbors are good candidates for center of stars.

Intuition A.6. *Nodes with many weighted neighbors are good candidates for center of stars.*

A.2.3 Example IV

In Figure 5(a), node u is subscribed to topics x_1, x_2,..., x_k. $(n-1)/(2k)$ nodes (set B_1) are subscribed to y_1, $(n-1)/(2k)$ nodes (set B_2) are subscribed to y_2, ..., $(n-1)/(2k)$ nodes (set B_k)are subscribed to y_k. $(n-1)/(2k)$ nodes (set A_1) are subscribed to x_1, y_1, $(n-1)/(2k)$ nodes (set A_2) are subscribed to x_2, y_2, ..., $(n-1)/(2k)$ nodes (set A_k) are subscribed to x_k, y_k. For this example, CD-ODA and CD-ODA-I first put edges between node u and all the nodes in sets $A_1, A_2, ..., A_k$ and then it puts edges between nodes $u_i \in A_i$ and all nodes in A_i and B_i, for $1 \leq i \leq k$. Thus, CD-ODA and CD-ODA-I use $((n-1)/(2k)) * k + ((n-1)/k - 1) * k = 3 * (n-1)/2 - k$ edges.

In the optimal solution (Figure 5(b)), there are edges between nodes $u_i \in A_i$ and all nodes in A_i and B_i, for $1 \leq i \leq k$. And there are edges between node u and nodes u_i, for $1 \leq i \leq k$. In the optimal solution, only $(k + ((n-1)/k - 1) * k = (n-1)$ edges are required. From this example, we can have the intuition that nodes with dense connections are good candidates for center of stars.

Intuition A.7. *Nodes with dense connections are good candidates for center of stars.*