Data Mining and Knowledge Discovery (2020) 34:1589–1620
https://doi.org/10.1007/s10618-020-00709-5

CrawlSN: community-aware data acquisition with maximum willingness in online social networks

Bay-Yuan Hsu[1] · Chia-Lin Tu[2] · Ming-Yi Chang[2] · Chih-Ya Shen[2]

Received: 10 January 2020 / Accepted: 25 July 2020 / Published online: 8 September 2020

Abstract

Real social network datasets with community structures are critical for evaluating various algorithms in Online Social Networks (OSNs). However, obtaining such community data from OSNs has recently become increasingly challenging due to privacy issues and government regulations. In this paper, we thus make our first attempt to address two important factors, i.e., user willingness and existence of community structure, to obtain more complete OSN data. We formulate a new research problem, namely *Community-aware Data Acquisition with Maximum Willingness in Online Social Networks* (*CrawlSN*), to identify a group of users from an OSN, such that the group is a socially tight community and the users' willingness to contribute data is maximized. We prove that CrawlSN is NP-hard and inapproximable within any factor unless, and propose an effective algorithm, named *Community-aware Group Identification with Maximum Willingness* (*CIW*) with various processing strategies. We conduct an evaluation study with 1093 volunteers to validate our problem formulation and demonstrate that CrawlSN outperforms the other alternatives. We also perform extensive experiments on 7 real datasets and show that the proposed CIW outperforms the other baselines in both solution quality and efficiency.

Keywords Social networks · Graph algorithm · Data acquisition

Responsible editor: Ira Assent, Carlotta Domeniconi, Aristides Gionis, Eyke Hüllermeier.

Electronic supplementary material The online version of this article (https://doi.org/10.1007/s10618-020-00709-5) contains supplementary material, which is available to authorized users.

✉ Chih-Ya Shen
chihya@cs.nthu.edu.tw

Extended author information available on the last page of the article

1 Introduction

Real social network datasets with community structures that include complete node and edge information are essential for evaluating various algorithms in Online Social Networks (OSNs) for different applications. For example, such real social network datasets can be used to evaluate the algorithms for link prediction, node classification, community detection, dense subgraph extraction, graph convolution networks, and graph embedding.

However, obtaining such real datasets with community structures from OSNs (represented as a graph) is not a simple task for two reasons: the node and edge information of the community should be completely acquired, and the community structure should be ensured at the same time. The main challenge of such a task is that in order to obtain detailed node (user) and edge (user–user interaction) information, such as posts, likes, check-ins, interactions/relations with other users in OSNs, one needs the users' explicit approval. However, if those users who agree to provide their detailed data collectively induce little or no community structure, the dataset cannot be used to evaluate the algorithms for OSNs that assume the existence of community structures in OSNs.

Although different algorithms have been proposed to address various issues for crawling community data from OSNs, such as detecting communities while crawling (Blenn et al. 2012), crawling multi-layer networks (Laishram et al. 2019), and crawling uniform samples (Ye et al. 2010; Gjoka et al. 2011), they usually assume that the crawler has been fully authorized by the users, i.e., that the users are willing to contribute their data for crawling. However, this assumption may be too strong in practical scenarios because of the serious concerns for individual privacy. Therefore, we argue that users' willingness to contribute data should be considered jointly with their community structures, in order to obtain more complete community data from OSNs for further analysis.

Currently, when taking users' willingness into consideration, researchers usually recruit OSN users to ask them to contribute their data manually, typically by recruiting them from online forum and crowdsourcing platforms, or sending messages to potential users in dense communities, such as in a class or office. This manual approach usually fails to assemble good community structures, because in most cases the users who are willing to contribute their data (from online forum and crowdsourcing platform) usually do not together form a community. In contrast, In contrast, when we focus only on the community structure and recruit the users in a dense community (in a class or an office), in most cases only a small portion of users may be willing to contribute their data. Our preliminary study on 1093 individuals (detailed in Sect. 6.1) shows that directly recruiting users in dense communities without considering their willingness may result in an acceptance rate (i.e., the ratio of the users who agree to contribute their data to the total number of users) below 10%. In contrast, recruiting users from online forums, although showing a very high acceptance rate, often results in an independent set with no community structure existing for these users. To address the above dilemma, we jointly consider two important yet fundamental factors for obtaining good community data from OSNs: *user willingness* and *com-*

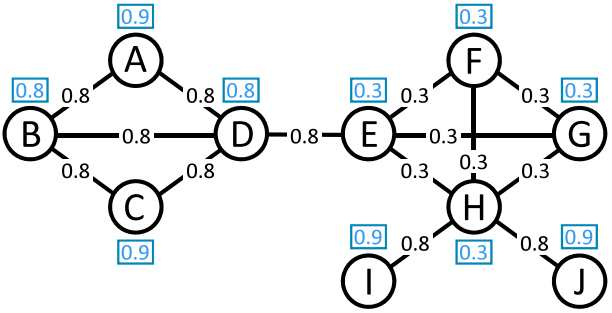


Fig. 1 Motivating example

munity structure (*social tightness*).[1] These two factors are detailed in the following paragraphs.

User willingness of data contribution. Due to privacy policies, if user data are crawled with APIs provided by OSN service providers such as Facebook and Instagram, users' explicit approval is required. The user willingness, therefore, is closely related to the successful obtaining of the node (user information) and edge (relationships among users) information. More specifically, two terms of user willingness should be considered, namely *individual willingness* and *influenced willingness*. The individual willingness is the willingness of the user to allow her data to be crawled, regardless of her friends' decisions. Influenced willingness, on the other hand, is an assessment of the user's friends influence on her decision, because research shows that a user's decision is heavily affected by the social reinforcement from her neighbors in the social network (Centola 2010; Bond et al. 2012; Deutsch and Gerard 1955). As users in a community are tightly connected and have a high influence on each other, influenced willingness should also be carefully examined.

Social tightness of the community. This factor is critical to the community structure emerging from the OSN users. If the users form a very sparse subgraph in the OSN, community structure may not exist, leaving researchers unable to benefit from data analysis. Nevertheless, as mentioned above, an intuitive approach such as recruiting users with monetary rewards in crowdsourcing platforms or forums usually fails to obtain good community structures. Therefore, an approach that jointly considers the factors of user willingness and social tightness is desired.

Figure 1 illustrates the two factors for obtaining community data. Given a social network in Fig. 1, where a greater value beside a node indicates the user has a stronger individual willingness to contribute her data, and the weight of the edge connecting nodes u and v indicates how strong u and v influence each other,[2] suppose that we'd

[1] Many other factors, such as relationship types of users are also important. Here, we discuss the two fundamental factors to crawl the community data for further analysis and discuss the other important factors in the future work.

[2] We show undirected edges here for the clarity of presentation. Directed relations can be easily incorporated in our problem formulation.

like to crawl a community with 4 users. One approach is to select the 4 users with maximum individual willingness, i.e., $G_1 = \{A, C, I, J\}$. However, G_1 induces an independent set with no community structure, which would therefore not be a good choice for analyzing the social network structures of the users. Another approach is to extract the densest subgraph $G_2 = \{E, F, G, H\}$, but the willingness of the users in G_2 is very low, indicating they are not likely to contribute their data. Finally, $G_3 = \{A, B, C, D\}$ strikes a good balance between the user willingness and structure completeness, i.e., users have a high willingness to contribute their data and G_3 is socially tight. Moreover, the users in G_3 can exert influence on each other, meaning that they are very likely to contribute their data as a whole community.

Till now, there has been scant research attention paid to jointly considering the willingness and social tightness factors to effectively obtain community data from OSNs. For this reason, we here a new research problem, namely *Community-awar e Data Acquisition with Maximum Willingness in Online Social Networks (CrawlSN)*, to select a socially tight group (community) of users from the OSN for the crawling task, such that the willingness of users' data contribution is maximized. By employing CrawlSN, the two important factors of user willingness and social tightness are addressed, and researchers will as a result have a high chance to successfully crawl community information from OSNs. Moreover, CrawlSN also incorporates a size constraint to include s users in the selected group so as to take into account the limited budget in data collection tasks. Later in our evaluation study in Sect. 6.1, we show that by considering both individual and influenced willingness, CrawlSN outperforms other alternatives for collecting user data from the OSN.

To our best knowledge, the proposed CrawlSN problem is the first research problem that jointly considers the individual and influenced willingness, social constraint, and the community size to obtain community data from OSNs. The entangled nature of the relevant factors make the CrawlSN problem very challenging, and existing approaches such as influence maximization (does not consider the community structure) and community detection approaches (no user willingness is considered) cannot be directly applied. In fact, we prove that the CrawlSN problem is NP-hard and inapproximable within any factor unless P=NP. That is, no approximation algorithm exists for CrawlSN. To tackle CrawlSN, we propose an efficient algorithm, named *Community-aware Group Identification with Maximum Willingness (CIW)*, with various ordering and pruning strategies (*MaxInner Ordering*, *Community-based Indexing for MaxInner Ordering*, *Core Pruning*, and *Willingness Pruning*) in order to boost the performance to obtain the optimal solutions.

The contributions of this paper are summarized as follows.

- We identify the crucial need to systematically obtain community data from OSNs, and propose a new research problem, called *Community-aware Data Acquisition with Maximum Willingness in Online Social Networks* (*CrawlSN*) to support the need.
- We also prove that CrawlSN is NP-hard and inapproximable within any factor, and we further propose an efficient algorithm, named *Community-aware Group Identification with Maximum Willingness* (*CIW*), with effective processing strategies.

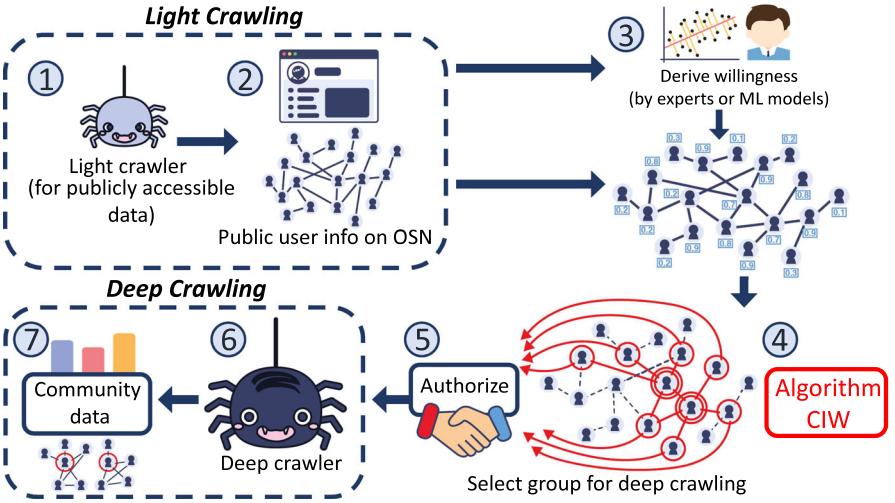


Fig. 2 Light crawling, deep crawling, and algorithm CIW

– We invite 1093 volunteers and conduct an evaluation study to validate the effectiveness of CrawlSN. The results indicate that more than 95% of users in the community selected by CrawlSN and CIW agreed to contribute their OSN data, significantly outperforming other baselines. We also show that CIW outperforms other baselines in terms of efficiency and solution quality in 7 real datasets.

The paper is organized as follows. Section 2 introduces the preliminaries, and then Sect. 3 formulates the research problem. Section 4 discusses the related works relevant to this paper, and Sect. 5 proposes the algorithm. Section 6 presents the results of the experiments. Finally, Sect. 7 concludes this paper.

2 Preliminaries

Community data acquisition with light and deep crawling. The proposed research problem aims to identify a group of socially tight users who have a high willingness to contribute their data, where the social network graph, users' individual willingness, and the influence strengths among users are assumed given. However, how do we obtain such information before the crawling is actually performed? To answer this question, we differentiate the notions of *light crawling* and *deep crawling* and then describe the two-step crawling process below. Figure 2 depicts the crawling process.

At the beginning of the crawling process (step 1 of Fig. 2), a *light crawling* is performed to obtain the publicly accessible user data (with user's consent), e.g., public profiles, public friend lists, public photos. As this step gathers only the publicly accessible user data, users are more likely to provide such data compared to more detailed personal data such as posts, likes, check-ins,[3] (as shown in step 2 of Fig. 2).

[3] We have implemented a light crawler using python 3.6, which is able to obtain the publicly accessible user data in OSNs.

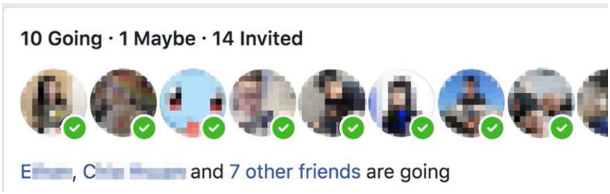


Fig. 3 Peer influence for willingness

However, the data retrieved by light crawling is very limited due to user's privacy settings. Therefore, as shown in step 3 of Fig. 2, the information obtained in light crawling is only used to construct the underlying social network structure, infer the influence strengths,[4] and predict the users' individual willingness to contribute data, i.e., their willingness to authorize the deep crawling.[5]

After this, as illustrated in step 4 of Fig. 2, the proposed algorithm *CIW* selects the group of socially-tight users who have a high willingness to contribute their data based on the input information generated in the previous step. In step 5, we request each selected user's authorization for *deep crawling*, as sending the request by Facebook *Events*, as shown in Fig. 3. Here, *deep crawling*, once authorized by the user with the API provided by OSNs, is able to obtain more detailed and private information of the user, such as the posts, replies, likes, photos, joined events, visited locations, and check-ins. Since the users are selected to maximize the willingness of data contribution, it is very likely that the set of users selected by *CIW* would authorize our deep crawling request. Therefore, in steps 6 and 7 of Fig. 2, deep crawling can obtain the detailed and private information of the users.

Willingness with peer influence in OSNs. An important feature captured by the proposed CrawlSN problem is the consideration of both individual willingness and influenced willingness. In our implementation, we leverage the *Events* function from Facebook to send our data-collection requests to allow peers to influence one another, as shown in Fig. 3. Here, the candidates selected by our algorithm can see their friends who are willing to contribute their data. In this way, they are more likely to contribute their data as a socially tight group. As the peer influence is shown to be a very effective (Bond et al. 2012; Centola 2010) factor in various activities, we incorporate this factor in our problem formulation, as detailed in Sect. 3.

3 Problem formulation and analysis

Given a social network $G = (V, E)$, where each node $v \in V$ represents a user and each undirected edge $e_{u,v} \in E$ represents the friendship between nodes $u, v \in V$. Each user $v \in V$ is associated with an individual willingness of data contribution (*individual*

[4] Influence strengths can be inferred by employing existing approaches (Kempe et al. 2003; Gomez-Rodriguez et al. 2012).

[5] We have built a simple machine learning model with SVM that predicts users' willingness with their publicly accessible information on Facebook.

willingness for short) $\delta_{v,\emptyset} \in [0, 1]$, where a larger $\delta_{v,\emptyset}$ indicates a stronger individual willingness. Moreover, each edge $e_{u,v}$ is assigned an edge weight $w_{u,v} \in [0, 1]$ to quantify how u and v influence each other.[6] In addition, given a set of nodes $S \subseteq V$, we denote $N_S(v)$ the set of v's neighbors in the subgraph induced by S.

The *willingness of data contribution* (*willingness* for short) of a user v in an induced subgraph S, i.e., $\delta_{v,S}$, considers the individual willingness of v and the influenced willingness, i.e., the willingness influenced by v's friends' willingness in $N_S(v)$, which is defined as follows.

$$\delta_{v,S} = \delta_{v,\emptyset} + (1 - \delta_{v,\emptyset}) \frac{\sum_{u \in N_S(v)} \delta_{u,\emptyset} \cdot w_{u,v}}{\tau_v}, \tag{1}$$

where $\tau_v = \sum_{u \in N_G(v)} w_{u,v}$ is the total incident edge weight of v.[7] The term $\delta_{v,\emptyset}$ is the individual willingness of v, and the second term $(1 - \delta_{v,\emptyset}) \frac{\sum_{u \in N_S(v)} \delta_{u,\emptyset} \cdot w_{u,v}}{\tau_v}$ is the *influenced willingness*.[8]

The first term in Eq. 1, i.e., *individual willingness* $\delta_{v,\emptyset}$, represents the initial willingness of user v to contribute her data without being influenced by her friends' decisions. The second term, *influenced willingness* in Eq. 1, represents the influence from v's friends, where the concept here is similar to that of *Partial Credits Model* in social influence (Goyal et al. 2010). Here, the edge weight $w_{u,v}$ quantifies how u and v influence each other. Therefore, $\frac{\sum_{u \in N_S(v)} \delta_{u,\emptyset} \cdot w_{u,v}}{\tau_v}$ can be viewed as the willingness influenced by v's neighbors in the selected group and is normalized by τ_v. Please note that $\frac{\sum_{u \in N_S(v)} \delta_{u,\emptyset} \cdot w_{u,v}}{\tau_v}$ considers v's influenced willingness based on v's friends who are also in S. This is because as shown in Fig. 3 in Sect. 2, each user v would be aware if her friends selected in S decide to contribute their data. Therefore, the selected users can discuss and make up their minds together. Moreover, $(1 - \delta_{v,\emptyset})$ in this term scales the influenced willingness such that $\delta_{v,S}$ is in the range $[0, 1]$.

Now, we define the *average willingness (of data contribution)* for a group of users $S \subseteq V$, i.e., $\Delta(S)$, as follows.

$$\begin{aligned} \Delta(S) &= \frac{\sum_{v \in S} \delta_{v,S}}{|S|} \\ &= \frac{1}{|S|} \sum_{v \in S} \left(\delta_{v,\emptyset} + (1 - \delta_{v,\emptyset}) \frac{\sum_{u \in N_S(v)} \delta_{u,\emptyset} \cdot w_{u,v}}{\tau_v} \right). \end{aligned} \tag{2}$$

[6] We can also consider directed influences in our problem formulation with a slight modification of the algorithm.

[7] If $\tau_v = 0$, we define the value of the second term of Eq. 1 as 0. That is, if $\tau_v = 0$, $\frac{\sum_{u \in N_S(v)} \delta_{u,\emptyset} \cdot w_{u,v}}{\tau_v} = 0$.

[8] In some extreme cases, for a user who is very unwilling to provide her data (i.e., with a small individual willingness), the influenced willingness may raise the value of Eq. 1 up to 1. To tackle this issue, an additional parameter $\beta \in [0, 1]$ can be added to the second term (i.e., influenced willingness) of Eq. 1 as follows. By setting a smaller β, i.e., close to 0, the user's individual willingness becomes more important in the computation of the average willingness.

Please note that $\Delta(S) \in [0, 1]$, where a larger $\Delta(S)$ indicates a stronger willingness of the group. Take Fig. 1 as an example with $S = \{A, B, C, D\}$, $\Delta(S) = 0.966$.

In addition to the average willingness of the group, we also employ a social constraint that requires the selected group to form a k-core, i.e., each member is required to have friendship with at least other k users in the group (Seidman 1983). Here, k-core is chosen as the social constraint because k-core is widely adopted as a density measurement in related research works (Shin et al. 2016; Giatsidis et al. 2011; Aksu et al. 2014; Alvarez-Hamelin et al. 2005; Zhang et al. 2017a; Wang et al. 2018; Candogan 2019; Aridhi et al. 2016). In this paper, k-core is used to quantify the social tightness of the selected group. The parameter k can be viewed as a trade-off between the two important factors: social tightness and the willingness. A larger k indicates that a more tightly connected group is desired. In contrast, if k is set smaller, we pay more attention on the users' willingness.

Furthermore, a group size constraint s is also incorporated to consider the limited budget for data crawling, which requires the selected group to contain exactly s nodes. The willingness value is a probability that is predicted by a machine learning method, such as SVM (Kubat 2015), Random Forest (Kubat 2015), or Deep Neural Networks (Goodfellow et al. 2016). The sum of the willingness values of the users of a group can be viewed as the expected number of people who will provide their data. Our objective value, i.e., the *average willingness*, is the expected number normalized by the group size. Specifically, the proposed research problem is formulated as follows.

Problem 1 ***Community-aware Data Acquisition with Maximum Willingness in Online Social Networks (CrawlSN).***

> ***Given:*** *A social network $G = (V, E)$, where each node $v \in V$ is associated with an individual willingness $\delta_{v,\emptyset} \in [0, 1]$, and each edge $e_{u,v} \in E$ is associated with an edge weight $w_{u,v} \in [0, 1]$. A social constraint k, and a size constraint s are also given.*
> ***Objective:*** *To find a group $S \subseteq V$, such that i) the induced graph of S is a k-core, ii) $|S| = s$, and iii) the average willingness of S, i.e., $\Delta(S)$, is maximized.*

Another advantage of choosing k-core as the social constraint is that it can be extended to consider other social tightness measures easily. For example, a clique of size s is a $(s-1)$-core; a k-plex of size s is a $(s-k-1)$-core. A k-core of size s is also a $\frac{k}{s-1}$-quasiclique. Moreover, our proposed algorithm can be easily extended to consider other measures, such as path-based measures: n-clubs, n-clans or n-cliques (Mokken 1979).

Effective and efficient processing of the CrawlSN problem is very challenging because we need to jointly consider the factors of willingness, social tightness, and the group size. As illustrated in Fig. 1, trivially maximizing the individual willingness or the social tightness cannot solve CrawlSN, and thus a carefully designed algorithm is required. In fact, CrawlSN is NP-hard and inapproximable within any factor, which means that no approximation algorithm exists, as stated in Theorem 1.

Theorem 1 *CrawlSN is NP-hard and inapproximable within any ratio unless P = NP.*

Proof We prove this theorem with the reduction from the p-clique problem. Given a graph $G_c = (V_c, E_c)$, and an integer p, the decision problem of p-clique decides whether their exists a complete subgraph with p nodes in G_c.

We transform each instance of p-clique to an instance of CrawlSN as follows. We construct the input graph $G = (V, E)$ by setting $V = V_c$, $E = E_c$, individuals willingness of each vertex as 1, the weight of each edge as 0. The parameters of CrawlSN are set as $s = p$ and $k = s - 1$. In the following, we prove that the decision problem of p-clique returns TRUE if and only if CrawlSN has a feasible solution. We first prove the sufficient condition. If p-clique returns TRUE with a solution H, i.e., $H \subseteq G_c$ is a complete graph with p nodes, then H must be a feasible solution of CrawlSN because $p = s$ and $k = s - 1$, i.e., H is a $s - 1$-core of size s. We then prove the necessary condition. If S is a feasible solution to CrawlSN, $|S| = s$ and S is a $s - 1$-core, i.e., each node in S must have edges linking to other $s - 1$ nodes in S, which implies that S is a clique of size s. That is, S is also a solution to p-clique. Therefore, CrawlSN is NP-hard. Moreover, if there is a ρ-approximation algorithm to CrawlSN with a finite ρ, it means that the p-clique can be decided in polynomial time, indicating that P=NP. The theorem follows. □

Since the proposed CrawlSN is a new research problem, there is no existing approaches that can directly solve it. To evaluate the performance of CIW with different α values, in this paper, we formulate an Integer Linear Programming (ILP) for the CrawlSN problem. The ILP formulation, which can be solved with any commercial solver (e.g., CPLEX[9] or Gurobi[10]), obtains the optimal solution and serves as a baseline for evaluation of the proposed algorithm. Please refer to "Appendix D of Supplementary information" for the detailed description of the ILP forumulation.

4 Related work

Influence maximization. The *Influence Maximization (IM)* problem and its extensions have been actively studied for years (Li et al. 2014; Chen et al. 2015; Li et al. 2015; Yang et al. 2016; Song et al. 2017; Lu et al. 2013) with the aim of finding a small number of users as *seeds* to maximize the number of activated users in a social network. Although at first glance the IM problems look similar to our CrawlSN, they are totally different. The IM problems aim to select a set of seeds to activate the maximum number of users; however, in our CrawlSN problem, we aim to find a set of users who have the maximum willingness to contribute their data while the set of users form a dense community. Since the willingness and social tightness factors are not jointly considered by IM, the IM algorithms cannot be directly applied to solve the CrawlSN problem. Moreover, extensions of the IM problem, such as competitive viral marketing (Lu et al. 2013), recommendation system (Yang et al. 2013), and rumor blocking (Song et al. 2017), have been proposed and discussed to consider different scenarios of influence maximization.

[9] https://www.ibm.com/analytics/cplex-optimizer.

[10] http://www.gurobi.com.

Dense subgraph extraction. Extracting dense subgraphs has been an active research field for decades. Cliques (and their relaxations) model the cohesive subgroups in social networks and are closely related to a number of fundamental graph problems (Cheng et al. 2010; Balasundaram et al. 2011; Zhang and Parthasarathy 2012). Another research topic to recently receive attention is how to efficiently extract k-plex (Balasundaram et al. 2011) or k-truss structures (Zhang and Parthasarathy 2012). Moreover, some research aims at extracting dense subgraphs and strengthening the social relations through network intervention (Hung et al. 2020; Hsu et al. 2019a).

However, although these studies propose effective approaches to extract dense subgraphs, they do not consider the user willingness of data contribution, which is the core concept of our proposed CrawlSN problem.

Community search. The *community search* problem aims at extracting densely connected community in social networks (Fang et al. 2016; Huang et al. 2015, 2014; Cui et al. 2014; Zhang et al. 2017b). For example, Fang et al. consider the keywords while the selected nodes form a k-core (Fang et al. 2016); Huang et al. discuss the closest community search problem (Huang et al. 2015) with k-truss based community model; and Zhang et al. study the (k, r)-core to find the group by considering pairwise similarity among users (Zhang et al. 2017b). On the other hand, some community search works focus on finding communities for different scenarios, such as querying geo-social groups (Yang et al. 2012; Shen et al. 2016; Zhu et al. 2017), extracting socially tenuous groups (Shen et al. 2017), identifying a group of socially tight viewers for multi-streaming (Shen et al. 2018), and finding a group for willingness maximization (Shuai et al. 2013). Finally, (Li et al. 2017) consider the social influence factor that aims at finding a group such that most people are activated by the targeted group However, although these works deal with a wide variety of community search problems, they cannot be applied to CrawlSN directly because they do not consider the individual and influenced willingness for crawling community data in OSNs.

Crawling online social networks. Due to the scale of OSNs and their privacy control policies, a crawled partial data set is often used for analysis. But as Ye et al. (2010) and Gjoka et al. (2011) point out, there is potential bias and skew introduced when crawling the online social network data. Blenn et al. (2012) propose to crawl the network and detect community structures at the same time, and the idea of crawling while identifying communities has also been extended to consider multi-layer networks (Laishram et al. 2019; Mucha et al. 2010). Although these works deal with different issues of crawling OSNs, the important factor of the willingness of the users to be crawled was not considered. Therefore, our proposed CrawlSN problem can complement the approaches mentioned above.

Finally, Hsu et al. (2019b) discuss the initial idea of crawling community-aware OSN data. In constrast to the high-level ideas, this paper designs and details the whole data acquisition process and the algorithm. Most importantly, the new evaluation study with 1093 users and the extensive experiments on multiple real datasets in this paper illustrate the effectiveness of the proposed approach.

5 Algorithm design

As stated in Theorem 1, the CrawlSN problem is NP-hard and inapproximable within any factor, i.e., no approximation algorithm exists. So, instead of proposing heuristic algorithms without any performance guarantee, we propose an efficient algorithm that is able to generate the optimal solution. The proposed algorithm, named *Community-aware Group Identification with Maximum Willingness (CIW)*, follows a branch-and-bound framework to efficiently obtain the optimal solution to CrawlSN.

To guide CIW to accurately explore various promising candidate groups, we propose the effective ordering and pruning strategies of *MaxInner Ordering*, *Community-based Indexing for MaxInner Ordering (CIMO)*, *Core Pruning*, and *Willingness Pruning* in order to address the willingness of users while taking into account the social and size constraints. Later we formally prove that the proposed CIW is able to generate the optimal solution. The pseudo code of CIW is listed in Algorithm 1.

Algorithm 1 CIW

Require: Graph G, size constraint s, social constraint k
Ensure: Selected group S
1: **function** FINDGROUP(S_R, S_C, s, k)
2: **while** $|S_C| \neq 0$ **do**
3: **if** $|S_R| + |S_C| < s$ **then**
4: **return**
5: **end if**
6: **if** $|S_R| = 0$ **then**
7: $S_R \leftarrow v$, where v is the node with the maximum δ_{v,S_C} in S_C
8: **end if**
9: Let v be the node selected by the ordering strategy (MaxInner Ordering or CIMO)
10: $S_R \leftarrow S_R \cup \{v\}$; $S_C \leftarrow S_C \backslash \{v\}$
11: **if** S_R is pruned by Core Pruning **then**
12: **return**
13: **end if**
14: **if** S_R is not pruned by Willingness Pruning **then**
15: FINDGROUP(S_R,S_C,s,k)
16: **else if** S_R is a feasible solution **and** $\Delta(S_R) > \Delta(S^*)$ **then**
17: $S^* \leftarrow S_R$
18: **end if**
19: $S_R \leftarrow S_R \backslash \{v\}$
20: **end while**
21: **end function**
22: FINDGROUP($\emptyset$, V, s, k)
23: **Output** S^*

The basic operations of the branch-and-bound framework for CIW are as follows. Let S_R be the group constructed by CIW so far, and S_C denotes the set of candidate users that could be moved into S_R. CIW also maintains *currently best solution* S^* to keep track of the best solution obtained so far while CIW progresses. Initially, S_R and S^* are set to empty, and S_C is set to V. At each iteration, CIW iteratively selects one node in S_C (based on an *ordering strategy* which will be detailed later) and moves it into S_R. When S_R contains exactly s nodes and is a feasible solution (i.e., S_R is a k-core of size s), CIW updates S^* as S_R if $\Delta(S_R) > \Delta(S^*)$. CIW then backtracks to the previous S_R and selects another node in S_C based on the ordering strategy and moves it to S_R to explore another new group. If all the nodes in S_C are considered by the

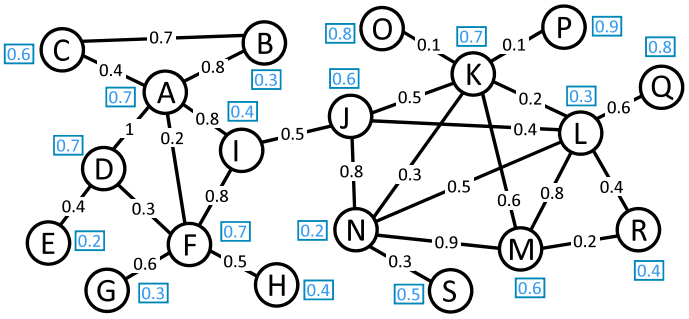
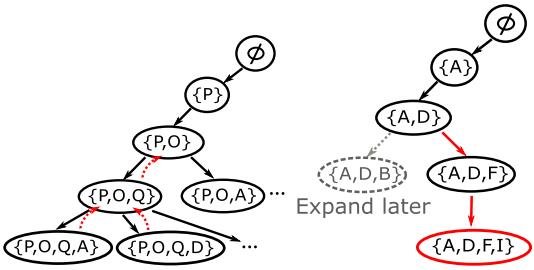


Fig. 4 Illustrative example for CIW ($s = 4$, $k = 2$)

Fig. 5 Ordering strategies



(a) *GIW* ordering **(b)** *MaxInner* ordering

current S_R, CIW backtracks to the previous state of S_R and the iterations repeat until all the candidate solutions are examined. Note that by following the above strategy, CIW examines all the combinations of different groups.

As CIW iteratively examines different candidate groups and updates S^*, the order of considering the nodes in S_C is critical to the performance, because if CIW is able generate good solutions earlier, it can avoid exploring a large number of redundant groups that cannot become better solutions (by leveraging the pruning strategies). Therefore, a good ordering strategy to select promising nodes from S_C to generate good solutions earlier is necessary.

Consider Fig. 4 as an example with $s = 4$ and $k = 2$. An intuitive ordering strategy, named *Greedy-based Individual Willingness (GIW)* ordering, iteratively selects the unvisited node in S_C with the largest individual willingness, as illustrated in Fig. 5a. However, the early-generated groups with s users, i.e., $\{P, O, Q, A\}$ and $\{P, O, Q, D\}$ are independent sets, indicating that this GIW ordering is not a good strategy because it does not consider the social dimension. To address the weakness of GIW ordering, we propose in Sect. 5.1 two advanced ordering strategies (*MaxInner Ordering* and *Community-based Indexing for MaxInner Ordering*) that consider all social, willingness, and size factors to efficiently derive feasible solutions.

Another major weakness of the above basic approach is that a large number of generated groups with s nodes are examined. However, since most of these groups are infeasible or come with poor average willingness, we propose in Sects. 5.2 and 5.3 two effective pruning strategies (*Willingness Pruning* and *Core Pruning*) to avoid

examining most of the infeasible or poor groups which will significantly reduce the computation time.

5.1 Ordering strategies: MaxInner and CIMO

To effectively identify promising nodes from S_C, at each iteration, an ordering strategy aims at identifying a node u in S_C such that i) including u largely increases the average willingness for S_R, i.e., to help obtain groups with good average willingness, and ii) u has a sufficient number of social edges connecting to nodes in S_R, i.e., to ensure that the result group satisfies the social constraint. In the following, we consider the above concepts to propose two ordering strategies, which represent different trade-offs. The first ordering strategy, called *Complete MaxInner Ordering* (*MaxInner Ordering* for short) selects the suitable node u with more complete information, but incurs more computation time; on the other hand, the second strategy, called *Community-based Indexing for MaxInner Ordering* (*CIMO* for short), employs some offline process to identify the promising node, thus significantly reducing the computation time. In our discussion of our experimental results in Sect. 6, we compare the performance of the two ordering strategies.

5.1.1 MaxInner Ordering

To address the first issue above, i.e., including a node that largely increases the average willingness for S_R, we first define $\iota(u)$ as the increment of the sum of willingness in S_R after including a node $u \in S_C$ into S_R as below.

$$\iota(u) = \sum_{v \in S_R \cup \{u\}} \delta_{v, S_R \cup \{u\}} - \sum_{v \in S_R} \delta_{v, S_R}.$$

Then, MaxInner Ordering computes $\iota(u)$ for each $u \in S_C$ and identifies the nodes that largely increase the average willingness as good candidates nodes.

On the other hand, to address the second issue of the selected node having a sufficient number of social edges connecting to nodes in S_R, we observed that if we always move a node u with at least $\min\{|S_R|, k\}$ edges connecting to S_R at each iteration, when $|S_R| = s$, it must be a k-core. Therefore, we define $m_u = |N_{S_R}(u)|$ as the number of edges linking from node $u \in S_C$ to the vertices in S_R. In this case, a larger $m_u - \min\{|S_R|, k\}$ indicates that there are more edges between u and S_R, which means that including u in S_R is more likely to generate a feasible solution.

To summarize, MaxInner Ordering prioritizes the inclusion of the node $u \in S_C$ into S_R if both $\iota(u)$ and $m_u - \min\{|S_R|, k\}$ are large. Therefore, among the vertices in S_C, MaxInner Ordering selects the vertex u based on the following equation to consider both factors simultaneously.

$$u = \arg\max_{\hat{u} \in S_C} \iota(\hat{u}) \times e^{m_{\hat{u}} - \min\{|S_R|, k\}}. \tag{3}$$

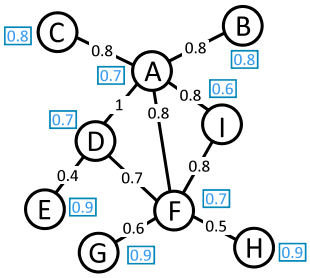


Fig. 6 Example of Maxinner Ordering

It is worth noting that $min\{|S_R|, k\}$ is not a fixed number because $|S_R|$ changes while the algorithm progresses, i.e., $|S_R|$ increases when new nodes are added from S_C, and $|S_R|$ decreases when CIW backtracks to consider other potential users. Moreover, at the beginning of CIW when $S_R = \varnothing$, we assign the vertex v in S_C which incurs the maximum willingness δ_{v,S_C} to avoid the *cold start* problem of $\iota(\cdot)$ (lines 6–8 of Algorithm 1), i.e., the case when $S_R = \varnothing$, it simply selects $u \in S_C$ with the maximum individual willingness but does not consider the social tightness factor.

MaxInner Ordering has a nice property of prioritizing the selection of nodes with high connections over those with good willingness but poor connections; this is critical to obtain feasible solutions early and makes it possible to allow the pruning strategies (detailed in Sects. 5.2 and 5.3) to avoid having to examine a large number of redundant groups.

Although MaxInner Ordering effectively identifies the vertex u which increases the average willingness with a sufficiently large number of social edges linking to the current S_R, however, extracting u with Eq. 3 takes $O(|V|^2)$ time, which may not be feasible for large OSNs. For this reason, we were motivated to propose the second ordering strategy (*CIMO*) to enhance the efficiency of the above approach.

We illustrate the MaxInner Ordering with Fig. 6. In this example, we set $s = 4$ and $k = 2$. At beginning when $S_R = \emptyset$ and $S_C = \{A, B, C, D, E, F, G, H, I\}$, CIW selects the node with the maximum δ_{v,S_C}, which is A, and moves A into S_R. Now $S_R = \{A\}$ and $S_C = \{B, C, D, E, F, G, H, I\}$. MaxInner Ordering then selects the node with the maximum $\iota(\hat{u}) \times e^{m_{\hat{u}} - \min\{|S_R|,k\}}$ from S_C, which is D ($\iota(D) \times e^{m_D - \min\{|S_R|,k\}} = 0.85 \times e^{1-\min\{1,2\}} = 0.85$. Please note that although G has a willingness higher than D, i.e., $\iota(G) = 0.9 > \iota(D) = 0.85$, however, as G has no connection to $S_R = \{A\}$, $\iota(G) \times e^{m_G - \min\{|S_R|,k\}} = 0.9 \times e^{0-\min\{1,2\}} = 0.331$. This illustrates the nice property of MaxInner Ordering, *it prioritizes the selection of nodes with high connections (i.e., D) over those with willingness but poor connections (i.e., G)*. This ensures the feasibility of the solution returned earlier by CIW. In summary, CIW selects $\{A, D, F, I\}$ first, and the nodes $\{B, C, E, G, H\}$ with poor connections are considered later.

5.1.2 Community-based indexing for Maxinner Ordering

As mentioned above, we observed that computing Eq. 3 is time-consuming and may not be practical for large OSNs. To address this issue, we propose the strat-

egy named *Community-based Indexing for MaxInner Ordering (CIMO)*, which is an offline preprocessing strategy. CIMO first employs a community detection approach (e.g., Blondel et al. 2008) to partition the input graph into multiple non-overlapping subgraphs and then calculates the number of edges across each pair of partitions. Such information can then be used to successfully and very efficiently identify promising nodes in S_C.

Given the input graph G, all the information required by CIMO can be computed off-line and needs to be computed only once, i.e., they are independent of the parameters s and k. Once the communities are generated, the information can be used by CIMO in multiple CrawlSN instances with different s and k values on the same input graph G. That is, these communities are regarded as index structures, and can be used multiple times for different parameters to improve the efficiency of the CIW algorithm.

Specifically, we are given the sets of the non-overlapping subgraphs obtained by the community detection approach $\{C_1, C_2, \ldots, C_x\}$, where $\bigcup_{i\in[1,x]} C_i = V$. CIMO in each iteration selects the node u in S_C with high number of edges connecting to $C_i \cap S_R$ for all C_i based on Eq. 4. Here, $I(\cdot)$ is the indicator function such that $I(C_i \cap S_R \neq \emptyset) = 1$ if $C_i \cap S_R \neq \emptyset$ holds, and $I(C_i \cap S_R \neq \emptyset) = 0$ otherwise. If CIMO finds that for some community C_i such that $C_i \cap S_R = \varnothing$, it can simply skip the examination of C_i. In other words, CIMO priorities the inclusion of the nodes u in S_C with a greater *extensibility*, i.e., including u to S_R makes it more likely to satisfy the social constraint, while the user u also has a high willingness value. The equation of CIMO is listed below.

$$u = \arg\max_{\hat{u}\in S_C} \delta_{\hat{u},\emptyset} \times \sum_{\forall C_i} |N_{C_i}(\hat{u})| \cdot I(C_i \cap S_R \neq \emptyset), \tag{4}$$

where $I(\cdot)$ is the indicator function such that $I(C_i \cap S_R \neq \emptyset) = 1$ if $C_i \cap S_R \neq \emptyset$ holds, and $I(C_i \cap S_R \neq \emptyset) = 0$ otherwise.

Since CIMO avoids the computation of the exact number of edges linking $u \in S_C$ and S_R, the time complexity of selecting $u \in S_C$ to move into S_R is reduced to $O(|V| \cdot s \cdot x)$, where s is the size of the solution and x is a constant representing the number of communities obtained by the community detection algorithm.

We analyze the time complexity of employing CIMO to select the node $u \in S_C$ and move it into S_R as follows. Given the communities $\{C_1, C_2, \ldots, C_x\}$, it is worth noting that the community detection algorithm is considered an off-line preporcessing step because we only need to find the communities once, regardless of the parameters k and s. Moreover, the term $|N_{C_i}(\hat{u})|$ in Eq. 4 can also be pre-computed offline, i.e., these values are also independent of parameters s and k. Then, to facilitate the fast lookup of the community each node $u \in V$ belongs to, we create a reverse index (an array W). Here, the reverse index is an array W with length $|V|$, where $W[u] = i$ if and only if $u \in C_i$. Therefore, the time complexity of computing $C_i \cap S_R$ is $O(|S_R|) = O(s)$. Since $|S_C| = O(|V|)$, and the number of clusters is x, the time complexity of selecting the node $u \in S_C$ to S_R by CIMO is thus $O(|V| \cdot s \cdot x)$.

5.2 Core pruning

Although the ordering strategies detailed above jointly consider the average willingness, social tightness, and the size constraint, during the running of the algorithm, S_R might not always satisfy the social constraint, and infeasible solutions may be generated. Please note that while the statement in Sect. 5.1.1 is true: "*if we always move a node u with at least* $min\{|S_R|, k\}$ *edges connecting to* S_R *at each iteration, when* $|S_R| = s$, *it must be a k-core*", it may not always be possible to find a node $u \in V_C$ satisfying the above condition. Consider an example in Fig. 4 with $s = 4$ and $k = 2$. Assume that at some iteration of the CIW algorithm, we have $S_R = \{A, I\}$ and $S_C = \{N, P, Q, R\}$. Here, $min\{|S_R|, k\} = 2$. However, for any node in S_C, it has no edge connecting to the current S_R. In this case, if CIW has to move two nodes from S_C to S_R, we may end up with an infeasible solution when $|S_R| = s$.

To effectively prevent the examination of such infeasible solutions, we propose the pruning strategy named *Core Pruning*, which trims off S_R at an early stage if it cannot grow into a feasible solution, i.e., a k-core of size s. In the following, we propose two pruning conditions of Core Pruning, *Group Member Connectivity (GM) condition* and *Candidate Connectivity (CC) condition*. During CIW, when the current S_R and S_C satisfy *either* one of the two pruning conditions, the current S_R and S_C can be safely pruned.

5.2.1 Group member connectivity (GM) condition of core pruning

For the first condition of Core Pruning, i.e., *Group Member Connectivity (GM) condition*, we first propose the notion of *Non-violating Quota* for S_R, i.e., $\omega(S_R)$, which is an upper bound on the number of nodes in S_C that S_R can include without violating the social constraint. Specifically, $\omega(S_R) = \min_{v \in S_R}\{|N_{S_C}(v)| + (s - k - (|S_R| - |N_{S_R}(v)|))\}$, where $N_{S_C}(v)$ is the set of neighboring nodes of v in S_C. Moreover, as the number of non-neighboring nodes of v in S_R is $|S_R| - |N_{S_R}(v)| - 1$, the algorithm can select at most $(s - k - (|S_R| - |N_{S_R}(v)|))$ additional non-neighboring nodes of v from S_C into S_R. The intuition behind why CIW can select at most $(s - k - (|S_R| - |N_{S_R}(v)|))$ additional non-neighboring nodes of $v \in S_R$ from S_C into S_R is as follows. First, S_R can only include $(s - |S_R|)$ additional nodes from S_C because the size constraint is s. Second, we need to select at least $(k - |N_{S_R}(v)|)$ neighboring nodes of v from S_C because the number of neighbors of v in S_R is $|N_{S_R}(v)|$, and when $|S_R| = s$, $|N_{S_R}(v)|$ needs to be at least k to satisfy the social constraint. Therefore, the difference of $(s - |S_R|)$ and $(k - |N_{S_R}(v)|)$, i.e., $((s - |S_R|) - k - |N_{S_R}(v)|)$, is the maximum number of non-neighboring nodes of v we can select from S_C to S_R. After rearranging the terms, we have $(s - |S_R|) - (k - |N_{S_R}(v)|) = (s - k - (|S_R| - |N_{S_R}(v)|))$.

We prove the above statement as follows.

Proposition 1 *CIW can select at most* $(s - k - (|S_R| - |N_{S_R}(v)|))$ *additional non-neighboring nodes of* $v \in S_R$ *from* S_C *into* S_R.

Proof S_R can only include $s - |S_R|$ additional nodes from S_C because the size constraint is s. Moreover, we need to select at least $k - |N_{S_R}(v)|$ neighboring nodes of v from S_C because the number of neighbors of v in S_R is $|N_{S_R}(v)|$. When $|S_R| = s$, $|N_{S_R}(v)|$

needs to be at least k. Therefore, we can add at most $(s - |S_R|) - (k - |N_{S_R}(v)|) = (s - k - (|S_R| - |N_{S_R}(v)|))$ non-neighboring nodes of v from S_C into S_R. Otherwise, S_R cannot form a k-core of size s. □

In the following, we detail how the definition of $\omega(S_R)$ (upper bound on the number of nodes in S_C that S_R can include without violating the social constraint) maps to the equation $\omega(S_R) = \min_{v \in S_R}\{|N_{S_C}(v)| + (s - k - (|S_R| - |N_{S_R}(v)|))\}$. From Proposition 1, we can add at most $(s - k - (|S_R| - |N_{S_R}(v)|))$ additional non-neighboring nodes of v from S_C into S_R without violating the social constraint. Moreover, we can add at most $|N_{S_C}(v)|$ neighboring nodes of v from S_C into S_R. Therefore, for every v in S_R, we can add at most $|N_{S_C}(v)| + (s - k - (|S_R| - |N_{S_R}(v)|))$ nodes from S_C into S_R without violating the social constraint. Since $\omega(S_R)$ is an upper bound on the number of nodes in S_C that S_R can include without violating the social constraint, we need to consider the minimum for all $v \in S_R$ that can be added from S_C. Therefore, $\omega(S_R) = \min_{v \in S_R}\{|N_{S_C}(v)| + (s - k - (|S_R| - |N_{S_R}(v)|))\}$ holds.

If $\omega(S_R) < (s - |S_R|)$, i.e., the maximum number of nodes that can be moved from S_C to S_R without violating the social constraint is smaller than the number of additional nodes S_R requires to satisfy the size constraint s, the current S_R and S_C can be safely discarded, and CIW backtracks to consider other candidate solutions. Moreover, if we replace $\omega(S_R) < (s - |S_R|)$ by its definition, we have $\omega(S_R) = \min_{v \in S_R}\{|N_{S_C}(v)| + (s - k - (|S_R| - |N_{S_R}(v)|))\} < (s - |S_R|)$, which can be rewritten as $\min_{v \in S_R}\{|N_{S_C}(v)| - k + |N_{S_R}(v)|\} < 0$, which is the same as $\min_{v \in S_R}\{|N_{S_C}(v)| + |N_{S_R}(v)|\} < k$.

The following lemma states the Group Member Connectivity (GM) Condition of Core Pruning.

Lemma 1 *If $\min_{v \in S_R}\{|N_{S_C}(v)| + |N_{S_R}(v)|\} < k$, at least one node $v \in S_R$ cannot follow the social constraint (i.e., $|N_{S_R}(v)| < k$) for any possible selection of vertices from S_C.*

Proof If $\omega(S_R) < (s - |S_R|)$, then at least one node $v \in S_R$ such that $|N_{S_C}(v)| + (s - k - (|S_R| - |N_{S_R}(v)|)) < (s - |S_R|)$ exists. That is, $(s - k - (|S_R| - |N_{S_R}(v)|)) < (s - |S_R|) - |N_{S_C}(v)|$ holds. As $|S_R| - |N_{S_R}(v)| - 1$ is the number of non-neighboring vertices for v, and $s - k - (|S_R| - |N_{S_R}(v)|)$ is the number of non-neighboring nodes for v to choose from S_C. For any possible selection $\widehat{S}_C \subseteq S_C$, let $\widehat{\lambda}_C$ be the number of neighboring nodes of v in S_C. Since $\widehat{\lambda}_C \leq |N_{S_C}(v)|$, $(s - |S_R|) - |N_{S_C}(v)| \leq (s - |S_R|) - \widehat{\lambda}_C$. Therefore, if $\omega(S_R) < (s - |S_R|)$, $s - k - (|S_R| - |N_{S_R}(v)|) < (s - |S_R|) - \widehat{\lambda}_C$, and $|n_{S_R}(v)| < k$ holds after $\widehat{S}_C$ is moved into S_R. The lemma follows. □

5.2.2 Candidate connectivity (CC) condition of core pruning

Here, we introduce the second Core Pruning condition, i.e., *Candidate Connectivity (CC) condition*. Specifically, given S_R and its corresponding S_C, the proposed *Candidate Connectivity (CC) condition* considers the connectivity of vertices in S_C. Here, the current S_R and S_C can be pruned if $\sum_{v \in S_C} |(S_C \cup S_R) \cap N_G(v)| < (s - |S_R|)(k - (|S_R| - 1))$ holds. The left-hand-side (LHS) of the inequality is the

sum of the numbers of neighbors in S_C and S_R each node in S_C has. The right-hand-size (RHS) of the inequality corresponds to the minimum required value of the total number of neighbors within S_R on any set of nodes from S_C that add to S_R, in order to make S_R a feasible solution. The intuition behind this claim is that if the LHS is smaller than the RHS, it implies that S_R cannot be expanded to a feasible solution (i.e., satisfying the social constraint) from S_C. Therefore, we can stop checking the nodes in S_C and backtrack to the previous state to consider other candidate solutions. In addition, this strategy can be further improved by replacing the LHS with $\sum_{v \in M_C} |(S_C \cup S_R) \cap N_G(v)|$, where M_C denotes the set of $s - |S_R|$ vertices in S_C with the largest numbers of neighbors also in S_C. Since $\sum_{v \in M_C} |(S_C \cup S_R) \cap N_G(v)| \leq \sum_{v \in S_C} |(S_C \cup S_R) \cap N_G(v)|$, our algorithm is able to prune off more infeasible solutions. Specifically, the proposed *Candidate Connectivity (CC) condition* is specified below and proved in Lemma 2.

$$\sum_{v \in M_C} |(S_C \cup S_R) \cap N_G(v)| < (s - |S_R|)(k - (|S_R| - 1)) \tag{5}$$

Lemma 2 *If* $\sum_{v \in M_C} |(S_C \cup S_R) \cap N_G(v)| < (s - |S_R|)(k - (|S_R| - 1))$*, the current* S_R *and* S_C *can be pruned because no feasible solution can be generated from them.*

Proof Since we only extract $s - |S_R|$ nodes from S_C to add into S_R, the upper bound of total number of neighbors within S_R of the $s - |S_R|$ extracted nodes is $\sum_{v \in M_C} |(S_C \cup S_R) \cap N_G(v)|$. If these $s - |S_R|$ extracted nodes follow the social constraint, each of them must be connected with at least $k - (|S_R| - 1)$ extracted nodes. This is because the extracted nodes can connect to at most $|S_R| - 1$ nodes in S_R and need at least $k - (|S_R| - 1)$ more connections to satisfy the social constraint. Since there are $s - |S_R|$ extracted nodes, the total number of neighbors they have within the set of $s - |S_R|$ extracted nodes is at most $(s - |S_R|)(k - (|S_R| - 1))$. Therefore, if $\sum_{v \in M_C} |(S_C \cup S_R) \cap N_G(v)| < (s - |S_R|)(k - (|S_R| - 1))$ holds, it indicates that there is at least one node's degree would be smaller than k after any $s - |S_R|$ nodes from S_C are moved into S_R. In other words, the current S_R and S_C cannot generate any feasible solution and thus can be pruned. □

Please note that the Core Pruning strategy prunes off S_R and its corresponding S_C which satisfy either the GM condition or the CC condition. Take Fig. 4 as an example. Let $s = 5$, $k = 4$, $S_R = \{K\}$, and $S_C = \{J, O, P, Q, R, M, S\}$. Therefore, $s - |S_R| = 5 - 1 = 4$, $\sum_{v \in M_C} |(S_C \cup S_R) \cap N_G(v)| = 2 + 1 + 1 + 1 = 5$, $k - (|S_R| - 1) = 4 - (1 - 1) = 4$. Since $\sum_{v \in M_C} |(S_C \cup S_R) \cap N_G(v)| = 5 < 4 * 4$, we can stop selecting nodes from S_C and CIW backtracks to consider other candidate solutions.

5.3 Willingness pruning

We further introduce the Willingness Pruning that prunes off S_R that cannot become a solution better than the currently best solution S^*, where S^* is maintained by CIW to record the best solution obtained so far. The idea of Willingness Pruning is to compute

an upper bound on the average willingness of S_R (according to the current S_C). If the upper bound does not exceed $\Delta(S^*)$, the current S_R can be pruned because its average willingness will never be larger than that of S^*.

Before we detail the Willingness Pruning, we first derive a property on the willingness of a user v in different subgraphs of the input social network G. In Lemma 3, given two groups of users S and S' where $S \subseteq S'$ and for any user $v \in S$, the willingness of v in S, i.e., $\delta_{v,S}$ must be upper-bounded by the willingness of v in S', i.e., $\delta_{v,S'}$. In other words, $\delta_{v,S} \leq \delta_{v,S'}$ must hold.

Lemma 3 *For any $S \subseteq S'$, $\delta_{v,S} \leq \delta_{v,S'}$ holds, $\forall v \in S$.*

Proof Given $v \in S$ and two groups S and S', where $S \subseteq S'$, recall that $\delta_{v,S} = \delta_{v,\emptyset} + (1 - \delta_{v,\emptyset}) \frac{\sum_{u \in N_S(v)} \delta_{u,\emptyset} \cdot w_{u,v}}{\tau_v}$ and $\delta_{v,S'} = \delta_{v,\emptyset} + (1 - \delta_{v,\emptyset}) \frac{\sum_{u \in N_{S'}(v)} \delta_{u,\emptyset} \cdot w_{u,v}}{\tau_v}$. That is, if $\delta_{v,S} > \delta_{v,S'}$, $\sum_{u \in N_S(v)} \delta_{u,\emptyset} w_{u,v} > \sum_{u \in N_{S'(v)}} \delta_{u,\emptyset} w_{u,v}$ must hold. Since $S \subseteq S'$, $N_S(v) \subseteq N_{S'}(v)$ holds. In other words, there must exist some $u \in S'$ such that $w_{u,v} < 0$ or $\delta_{u,\emptyset} < 0$, in order to make $\sum_{u \in N_S(v)} \delta_{u,\emptyset} w_{u,v} > \sum_{u \in N_{S'(v)}} \delta_{u,\emptyset} w_{u,v}$ hold, which contradicts the definition of $w_{u,v}$ and $\delta_{u,\emptyset}$ (both should be in the range of [0, 1]). Therefore, for any $S \subseteq S'$, $\delta_{v,S} \leq \delta_{v,S'}$ holds, $\forall v \in S$. □

Figure 1 presents an example. Let $S = \{B, D\}$ and $S' = \{B, C, D\}$ where $S \subseteq S'$. Then, $\delta_{C,S} = 0.9 + 0.1 * (0.8 * 0.8)/(0.8 + 0.8) = 0.94$ and $\delta_{C,S'} = 0.9 + 0.1 * (0.8 * 0.8 + 0.8 * 0.8)/(0.8 + 0.8) = 0.98$. If $\delta_{C,S} > \delta_{C,S'}$, this implies $\delta_{D,\emptyset} \cdot w_{D,C} < 0$, which contradicts the definition of $w_{D,C}$ and $\delta_{D,\emptyset}$.

Then, given S_R and S_C, we can derive an upper bound of the average willingness that can be achieved by them, i.e., $\Upsilon(S_R, S_C)$, as follows.

$$\Upsilon(S_R, S_C) = \frac{1}{s} \cdot \left(\sum_{v \in S_R} \delta_{v,S_R \cup S_C} + \sum_{v \in S_C}^{s - |S_R|} \max \delta_{v,S_R \cup S_C} \right),$$

where $\sum_{v \in S_C}^{s-|S_R|} \max \delta_{v,S_R \cup S_C}$ denotes the sum of the top $(s - |S_R|)$ maximum values in $\sum_{v \in S_C} \delta_{v,S_R \cup S_C}$.

In the following, we show an important property of $\Upsilon(S_R, S_C)$.

Proposition 2 *If $S_R \subseteq S'$ and $|S'| = s$, then $\Upsilon(S_R, S_C) \geq \max_{S' \subseteq S_R \cup S_C} \Delta(S')$ holds.*

Proof Since $S_R \subseteq S'$, $S' = S_R \cup S' \setminus S_R$ holds, and we can rewrite $\Delta(S')$ as $\Delta(S') = \frac{1}{s} \cdot (\sum_{v \in S_R} \delta_{v,S'} + \sum_{v \in S' \setminus S_R} \delta_{v,S'})$ because $|S'| = s$. Recall that $\Upsilon(S_R, S_C) = \frac{1}{s} \cdot (\sum_{v \in S_R} \delta_{v,S_R \cup S_C} + \sum_{v \in S_C}^{s-|S_R|} \max \delta_{v,S_R \cup S_C})$. By Lemma 3, we know that $\sum_{v \in S_R} \delta_{v,S'} \leq \sum_{v \in S_R} \delta_{v,S_R \cup S_C}$ because $S' \subseteq S_R \cup S_C$. Moreover, $\sum_{v \in S' \setminus S_R} \delta_{v,S'} \leq \sum_{v \in S_C}^{s-|S_R|} \max \delta_{v,S'} \leq \sum_{v \in S_C}^{s-|S_R|} \max \delta_{v,S_R \cup S_C}$ because i) $S' \setminus S_R \subseteq S_C$, and $|S' \setminus S_R| = s|S_R|$ (for the first inequality), and ii) from Lemma 3 $\delta_{v,S'} \leq \delta_{v,S_R \cup S_C}$ holds, $\forall v \in S'$. Therefore, $\Upsilon(S_R, S_C) \geq \max_{S' \subseteq S_R \cup S_C} \Delta(S')$ if $S_R \subseteq S'$ and $|S'| = s$. □

By Proposition 2, if $S_R \subseteq S'$ and $|S'| = s$, then $\Upsilon(S_R, S_C) \geq \max_{S' \subseteq S_R \cup S_C} \Delta(S')$ holds. This upper bound is very effective for pruning redundant search space exploration of CIW. That is, if $\Upsilon(S_R, S_C) \leq \Delta(S^*)$, we can safely truncate the current S_R (and its corresponding S_C) because the current S_R will never become a solution better than S^*. That is, by adding any $s - |S_R|$ users in S_C into S_R will never make it a solution better than S^*. Therefore, the current S_R does not need to be expanded and can be safely pruned. We prove this in the following lemma.

Lemma 4 *If $\Upsilon(S_R, S_C) \leq \Delta(S^*)$, the current S_R will never become a solution better than S^*.*

Proof We prove this lemma with contradiction. Suppose that the current S_R can grow into a better solution when $\Upsilon(S_R, S_C) \leq \Delta(S^*)$. That is, there exists a better solution S' expanded from the current S_R (by including nodes in the current S_C), i.e., $S_R \subseteq S'$ and $|S'| = s$. This indicates that $\Delta(S') > \Delta(S^*) \geq \Upsilon(S_R, S_C)$. However, by Proposition 2, $\Upsilon(S_R, S_C) \geq \max_{S' \subseteq S_R \cup S_C} \Delta(S')$, if $S_R \subseteq S'$ and $|S'| = s$. This implies that $\Upsilon(S_R, S_C) \geq \Delta(S')$ and causes a contradiction. Therefore, if $\Upsilon(S_R, S_C) \leq \Delta(S^*)$, the current S_R will never become a solution better than S^*. □

It is worth noting that if $S_R = \emptyset$ and $\Upsilon(S_R, S_C) < \Delta(S^*)$, the Willingness Pruning here can be viewed as an early termination strategy, i.e., *CIW* can stop. For instance in Fig. 4, after we expand every branch which contains K, we pop K from S_R and $S_R = \emptyset$ here. The current S_C becomes $\{J, L, M, N, R\}$ with $S_R = \emptyset$ and $\Upsilon(\emptyset, S_C) < \Delta(S^*)$, and CIW can stop since S_C does not include any solution better than S^*.

Theorem 2 *CIW finds the optimal solution to CrawlSN.*

Proof As MaxInner Ordering (or CIMO) stops extracting nodes from S_C when $S_C = \emptyset$, CIW with MaxInner Ordering (or CIMO) examines all combinations of groups of size s. According to Lemmas 1 and 2, Core Pruning only trims off S_R that will not grow into a k-core. Moreover, Lemmas 3 and 4 state that Willingness Pruning only trims off S_R that will not become a solution better than the best solution so far S^*. Therefore, CIW is able to obtain the optimal solution to CrawlSN. The theorem follows. □

Discussion of CIW in large OSNs. In real-life scenarios, the OSNs may be very large. Although the proposed CIW is able to effectively minimize the number of examined candidate groups by employing the ordering and pruning strategies, the computation time of deriving the optimal solution may be quite large. Therefore, we may sacrifice a small portion of solution quality to significantly boost the efficiency. Specifically, we can incorporate a parameter α in CIW to bound the number of generated feasible solutions. That is, after α feasible solutions have been generated, CIW can return S^* directly (the best solution obtained so far) and stops. As will be shown in the experiments (in Sect. 6), setting a proper α makes it possible to obtain near-optimal solutions with a significant improvement on the efficiency. Please note that the CIW algorithm that obtains the optimal solution can be regarded as setting $\alpha = \infty$.

6 Experimental results

In this section, we first discuss the validation of the problem formulation of CrawlSN with an evaluation study on 1093 users. We then detail the experiments we conduct on 7 real datasets to evaluate the performance of the proposed CIW algorithm.[11]

6.1 Evaluation study

We first provide the details of the setup of the evaluation study and then present the results by comparing our proposed approach with 5 other baseline approaches.

6.1.1 Setup

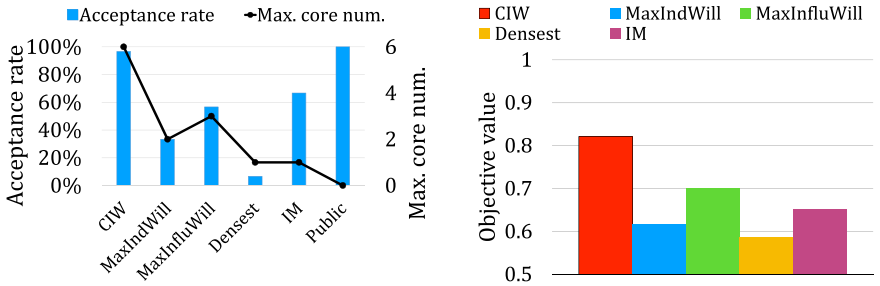
We recruit 1093 volunteers, who form a connected social network, and perform an evaluation study to validate the proposed CrawlSN and algorithm CIW. Please refer to "Appendix A of Supplementary information" for the detailed recruitment procedure and the statistics of the volunteers. Please note that these volunteers only agree to provide us with their Facebook public page (i.e., for light crawling). Then, when our algorithm or other baseline approaches select them, they decide on their own free will whether or not to authorize the deep crawling request.

After the volunteers are recruited, the subsequent steps are similar to those illustrated in Fig. 2. At the beginning, with the volunteers' consents, we perform a light crawling on their Facebook page to obtain their public information, such as profile, public photos, and publicly-available friend lists (steps 1, 2 in Fig. 2). We then infer the individual willingness (node weight) for each of them as well as the influence strength (edge weight) between them with a machine learning model based on their light-crawled data. We describe how the individual willingness values and the influence strengths are inferred in "Appendix B of Supplementary information". We also discuss the scenario if information acquired by light crawling is limited in "Appendix F of Supplementary information".

We employ the proposed `CIW` and other 5 different baselines to let each approach select 30 users from the above network (step 4 in Fig. 2). We then ask the selected users to authorize the deep crawler to crawl their posts and check-in information in Facebook (step 5 in Fig. 2). The authorization requests are sent through Facebook *Events*, similar to Fig. 3, to allow the users to know the decisions of their friends (who are also selected). Please note that in this step, the volunteers are asked to freely decide whether or not to authorize our deep crawling request. Then, for those who grant authorization, our deep crawler crawls their data on Facebook (steps 6, 7 in Fig. 2). Here, a user is considered *accepted* the request if she authorizes our deep-crawling request in 5 days; otherwise, the user is assumed to have *rejected* the request. Please refer to "Appendix C of Supplementary information" for further details for this evaluation study.

We set $s = 30, k = 5$ for `CIW` to select 30 users who form a 5-core, and the following 5 additional baselines are implemented and compared: i) `MaxIndWill`, which

[11] The source codes are available online http://www.cs.nthu.edu.tw/~chihya/CIW_download/.



(a) Acceptance rates and maximum core numbers

(b) Average willingness

Fig. 7 Results of evaluation study

chooses 30 users with the maximum summation of individual willingness, i.e., it maximize the first term in the objective function of CrawlSN in Eq. 2. ii) `MaxInfluWill`, which selects the 30 users with the maximum influenced willingness, i.e., the second term in Eq. 2. iii) `Densest`, which selects 30 users with the maximum density, where the density of a group $\widehat{S}$ with induced vertex set $V(\widehat{S})$ and induced edge set $V(\widehat{S})$ is defined as $\frac{|E(\widehat{S})|}{|V(\widehat{S})|}$. iv) `IM`, which is the algorithm for influence maximization to find k seeds to maximize the spread (Chen et al. 2009); the seed number is set to 5 and the 30 earliest activated users are selected. v) `Public`: we post the recruitment information in a public forum dedicated to recruiting participants for user studies and select the 30 users who were earliest to respond to our recruitment post.

6.1.2 Results of evaluation study

In Fig. 7a, we compare the *acceptance rates*, defined as the ratios of the selected users who accept the requests (each approach selects 30 users), and the *maximum core numbers* for different approaches. Here, the maximum core number is the largest $\widehat{k}$ such that the users who accept the requests form a $\widehat{k}$-core. A larger maximum core number indicates a tighter community formed by the selected users.

The proposed `CIW` outperforms the other baselines for the acceptance rate and the maximum core number because it jointly considers the individual and influenced willingness of the users as well as the social tightness of the returned group. Specifically, 29 out of 30 users selected by `CIW` accept the request, and the maximum core number of them is 6, indicating that it is able to identify a socially tight set of users who have a high willingness to contribute their data.

`MaxIndWill` and `MaxInfluWill` do not perform well with both having low acceptance rates and low maximum core numbers. This is because for these two baselines, each of them only corresponds to one term in the objective function of CrawlSN. This validates the effectiveness of the objective function of CrawlSN and confirms that considering both the individual willingness and influenced willingness is crucial. For `MaxInfluWill` and `MaxIndWill`, their average individual willingness values are quite close (0.55 and 0.57, respectively). However, as `MaxInfluWill` finds a more socially tight group with high influence willingness and `MaxIndWill`

Table 1 Dataset descriptions

Network	FB_168	FB_224	FB_347	Email	Youtube	LiveJournal	Friendster
$\|V\|$	168	224	347	36,692	1,134,890	3,997,962	65,608,366
$\|E\|$	1656	3192	2514	367,662	2,987,624	34,681,189	1,806,067,135

selects some users with very few social connections to the others, the users selected by `MaxInfluWill` achieve higher acceptance rate compared to `MaxIndWill`.

`Densest` performs very poorly because it does not consider the willingness factor. As very few users accept the data crawling request, the number of edges induced by the users selected by it is thus low, which results in a low maximum core number. `IM` also has a very poor maximum core number because it does not consider the social factor, and the selected users do not form a socially tight community. Finally, the acceptance rate of `Public` is the maximum because users who respond to our recruitment post must have accepted the data crawling request. However, these users form an independent set, which means that no community structure exists for them. From Fig. 7a, we conclude that our proposed CrawlSN problem and algorithm `CIW` effectively obtain a socially tight community in which users are willing to contribute their data.

Figure 7b further demonstrates the *average willingness* (objective value of CrawlSN) calculated from the users selected by each algorithm. Please note that `Public` is not included because the users who respond to the recruitment post must have had 100% average willingness. Our `CIW` achieves very high average willingness whereas the other baselines which have poor acceptance rates (Fig. 7a), all have poor average willingness. This indicates that the individual willingness and the influenced willingness considered by CrawlSN precisely capture user's willingness of data contribution.

6.2 Performance evaluations

We evaluate the performance of CIW and other baselines with 7 real datasets as listed in Table 1. Here, datasets *FB_168*, *FB_224*, and *FB_347* are the three largest components of the OSN dataset *ego-facebook* (Leskovec and Mcauley 2012), whereas *Email* is an e-mail social network (Leskovec et al. 2009), and *Youtube* is the social network extracted from Youtube (Yang and Leskovec 2015). Moreover, dataset *Friendster* includes approximately 65 million nodes and 1.8 billion edges (Yang and Leskovec 2015), while *LiveJournal* has approximately 4 million nodes and 34 million edges (Yang and Leskovec 2015). The individual willingness and edge weights are assigned randomly in [0, 1].

In our experiments, `CIW` refers to the proposed CIW algorithm, equipped with CIMO, Core Pruning, and Willingness Pruning. Later in Fig. 9c, we also present our component analysis to illustrate the effectiveness of CIW's different ordering and pruning strategies. To demonstrate the effectiveness and efficiency of `CIW`, we compare our proposed `CIW` with four baseline approaches: (1) `Random`, which randomly selects

s users from the social network, (2) IC (Li et al. 2015), which extracts a k-core that maximizes the influence of the group, but does not consider the willingness and size constraint as in CrawlSN, (3) IM (Chen et al. 2009), which is the algorithm for influence maximization (also detailed and compared in the evaluation study). For IM, we set the seed number to 5 and regard the s earliest activated users as the selected group, and (4) ILP, which is an Integer Linear Programming (ILP) for the CrawlSN problem as mentioned in Sect. 3 and detailed in "Appendix D of Supplementary information".

Unless specifically indicated, we set the parameter α of CIW to ∞, which means that CIW obtains the optimal solution to each CrawlSN instance. For the parameter k, we set this parameter according to IC (Li et al. 2015), which sets $k \in [2, 256]$ with a default $k = 32$. Please note that in our datasets, setting a large k, i.e., $k \geq 64$ may result in no solution due to the strict social constraint specified by k. For parameter s, we set its range within 30 to 3, 000 to consider different scenarios. All the algorithms are implemented in C++ and tested on a computer with Intel Core i7-7700K and 128GB RAM. Each result is averaged over 20 runs.

6.2.1 Sensitive tests on small networks

Figure 8a–d compare the performance of CIW with other baselines on different small datasets. Figure 8a compares the average willingness (objective value of CrawlSN) of the groups obtained by each approach. Please note that CIW (with $\alpha = \infty$) and ILP here both obtain the optimal solution. However, as shown in Fig. 8c later, ILP incurs unacceptable computation time even for these small social networks, which is much greater than that of CIW.

CIW outperforms other baselines, i.e., IC, IM, and Random significantly because it jointly considers individual and influenced willingness, whereas IC focuses more on social connectivity, IM pays more attention to the influenced willingness, and Random randomly selects users without considering any factor. Therefore, these baselines achieve poor solution quality.

Figure 8b presents the maximum core number of each group selected by each approach. The maximum core numbers of CIW and ILP are high, indicating that the community structure is well preserved for the selected group. In contrast, IM and Random are 0, i.e., no community structure exists for their selected users. IC has a maximum core number near 5, however, the returned groups are too small, i.e., between 6 and 18, not satisfying the size constraint. Figure 8c indicates that as the network size increases, the time required by CIW gradually increases, but CIW is still able to obtain the optimal solution efficiently. In contrast, ILP and IM incur large computation time for different datasets. Figure 8d presents the results on network *FB_347* with different k. Again, CIW constantly outperforms the other baselines in terms of average willingness.

In the discussion of Sect. 5, we mention that a parameter α can be employed to request the proposed algorithm to return the best solution obtained so far as soon as α feasible solutions have been obtained. In Fig. 8e, f, we compare CIW, which obtains the optimal solution, against its variations with $\alpha = \{20, 100, 500\}$ and ILP on the *FB_347* network. Figure 8e shows that the objective values (average willingness) for $\alpha = \{20, 100, 500\}$ are very close to CIW and ILP, i.e., the optimal solutions, while

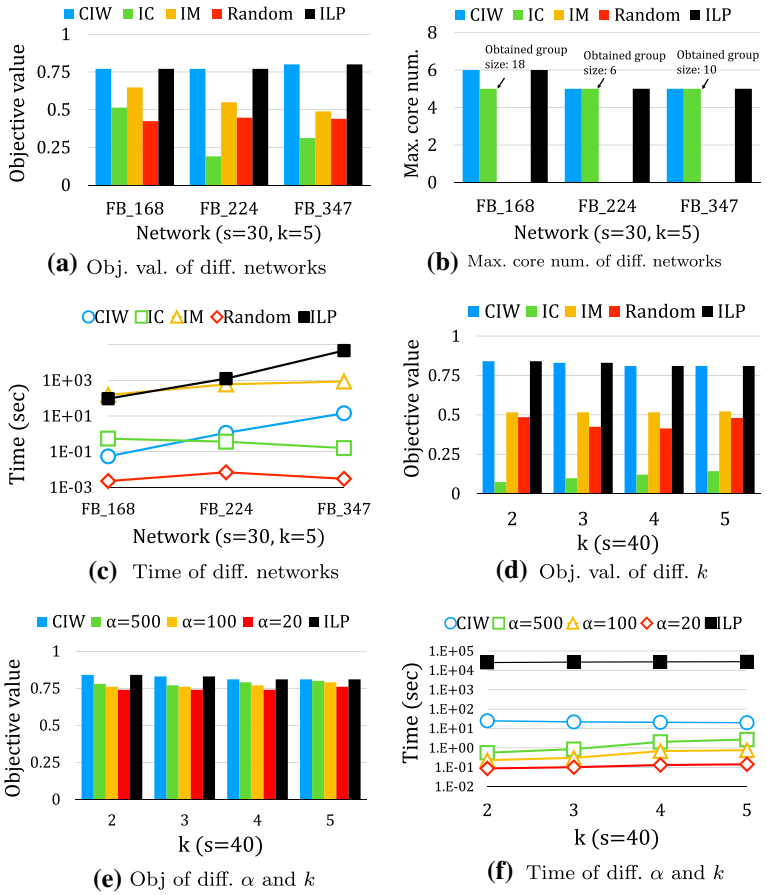


Fig. 8 Sensitivity tests on small datasets

Fig. 8f indicates that CIW with $\alpha = \{20, 100, 500\}$ are much more efficient than the original CIW and ILP that obtain the optimal solution. This indicates that by setting a small α, we are able to sacrifice only a tiny part of the average willingness to significantly improve the efficiency. This makes our algorithm very suitable for large-scale OSNs.

6.2.2 Scalability and sensitivity tests on large networks

We present the results of the proposed CIW and the other baselines on two large datasets, i.e., *Email* and *Youtube*. Please note that IM is absent because it does not return any solution within 4 days. Here, the α values of CIW for *Email* and *Youtube* are set to 200 and 1000, respectively. That is, CIW stops when 200 and 1000 feasible solutions are examined for the two respective datasets.

Figure 10a, b show the results on Youtube with more than 1M users for testing the scalability of the algorithms. Figure 10a, b show that the proposed CIW outperforms

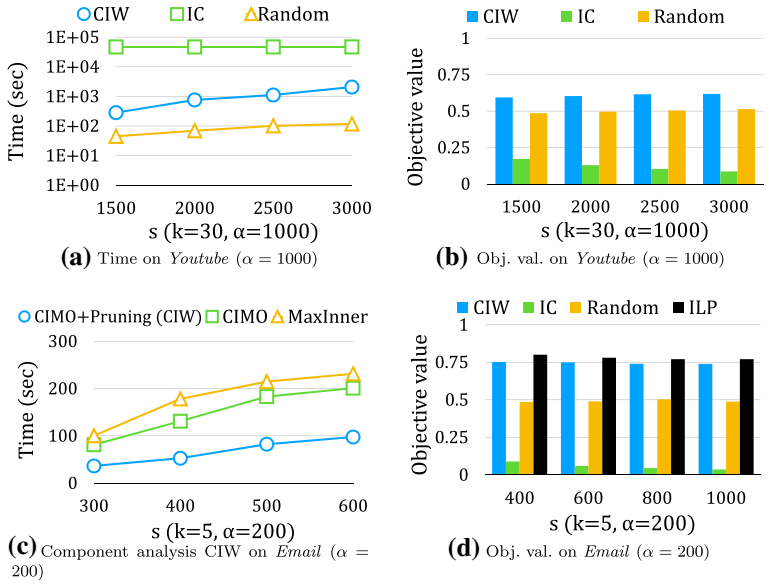


Fig. 9 Experiments on *Youtube* and *Email*

`IC` in both efficiency and solution quality because `CIW` takes into consideration all the willingness, social, and size factors of CrawlSN. `Random` again returns independent sets with poor objective values, i.e., no community structure is obtained while the users' average willingness for data contribution is low.

Figure 9c, d compare the proposed `CIW` for different s on the *Email* network. Figure 9c evaluates the effectiveness of each component of `CIW`. We first contrast the ordering strategies proposed in Sect. 5.1. Equipped with *Community-based Indexing for MaxInner Ordering (CIMO)*, CIW with CIMO (`CIMO` for short) outperforms CIW with MaxInner Ordering (`MaxInner` for short) because `CIMO` significantly reduces the computation overhead of Eq. 3. Moreover, when equipped with all the ordering (CIMO) and pruning strategies, `CIMO+Pruning` outperforms `CIMO` and `MaxInner` because the Willingness and Core Pruning strategies effectively trim off a huge number of S_R that cannot grow into feasible solutions. Please note that under the condition of $\alpha = 200$ in Fig. 9c, `MaxInner` on average achieves a 1.2% average willingness improvement as compared to `CIMO`. This confirms that CIMO trades off small solution quality for significant improvement in the efficiency. Please note that we do not compare `CIW` with `IC` and `Random` here on *Email* in terms of the computation time because those algorithms show similar trends in both *Email* and *Youtube* datasets.

Figure 9d demonstrates that `CIW` outperforms other baselines in average willingness on the *Email* dataset. Since the willingness of the users is not well examined, `IC` performs poorly. Although `Random` has a higher average willingness than `IC`, the returned groups are all independent sets, which *does not satisfy our requirement for obtaining community structures*.

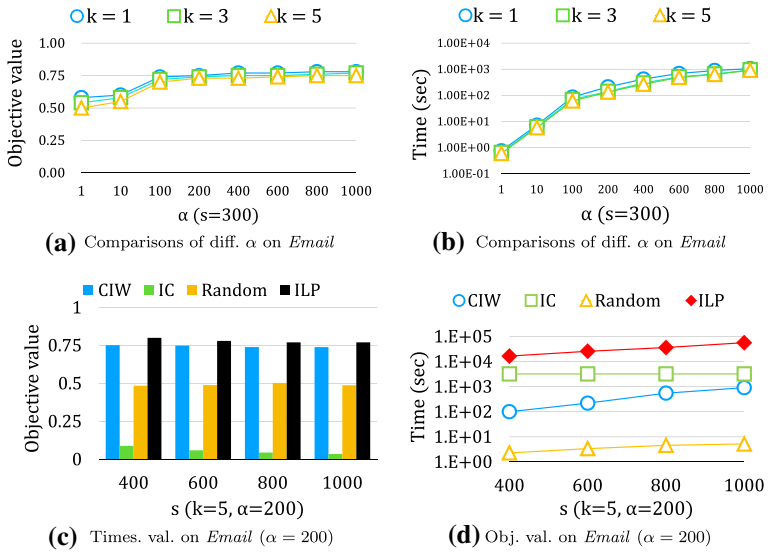


Fig. 10 Comparisons on *Email*

In the following, we show the scalability of CIW on large-scale datasets, including *Friendster* (approx. 65 million nodes and 1.8 billion edges) (Yang and Leskovec 2015), *LiveJournal* (approx. 4 million nodes and 34 million edges) (Yang and Leskovec 2015), and *Email* (36,692 vertices and 367,662 edges).

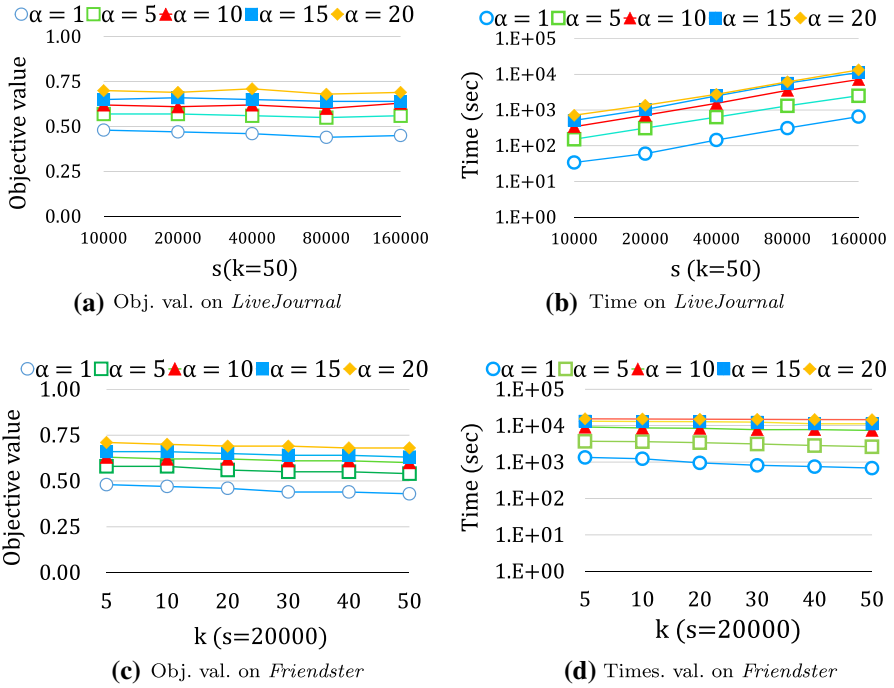
To demonstrate that CIW with various α achieves very good solution quality while incurring much less computation time compared to ILP, we first compare the results of ILP and CIW with different α values on *Email* in Fig. 10. Then, we show the scalability of the proposed CIW on datasets *Friendster* and *LiveJournal* in Fig. 11.

Figure 10a, b present the objective values and computation time of different α and k on dataset *Email*. Figure 10a shows that the objective values of CIW converge quickly after $\alpha = 100$, which, as shown in Fig. 10b, takes no more than 2 minutes for $\alpha = 100$. This indicates that by setting a small α value, CIW is able to achieve very good objective values while significantly reducing the computation time. In addition, we also compare CIW with ILP on *Email* in Figs. 10c, d, and we set $\alpha = 200$ for CIW. Figure 10c, d show that CIW is able to achieve very comparative results on solution quality (objective value), while incurring much less computation time.

Figure 11 presents the results of CIW with various α values on *LiveJournal* and *Friendster* datasets. As α gradually increases, the solution quality becomes much better while the computation time increases slowly. This indicates that CIW is able to trade off a small portion of solution quality to significantly improve the efficiently in large-scale social networks.

6.2.3 Component analysis of the proposed approaches

Trade-off for CIMO. In our implementation, we adopt the Louvain algorithm (Blondel et al. 2008), an efficient algorithm to identify the communities off-line for CIMO. Its



(a) Obj. val. on *LiveJournal* **(b)** Time on *LiveJournal*

(c) Obj. val. on *Friendster* **(d)** Times. val. on *Friendster*

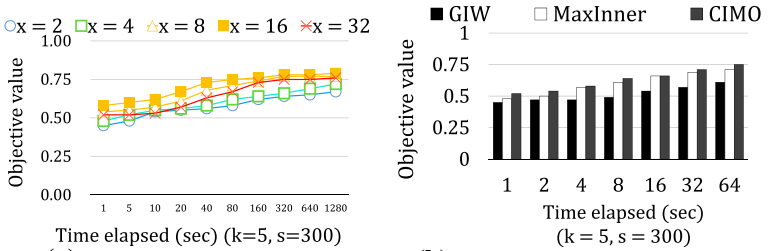
Fig. 11 Experiments on *LiveJournal* and *Friendster*

running time is $O(nlog^2n)$, where n is the number of nodes in the graph. The Louvain algorithm is a hierarchical clustering algorithm which iteratively merges small clusters into large ones, until the maximum modularity is achieved. During the clusters are merging, we can stop the algorithm when the cluster number is x.

In the following, we compare the performance of CIW equipped with CIMO with different x values. Figure 12a presents the objective values of CIW with different values of x for CIMO on the *Email* dataset. Here, the y-axis (time elapsed) is the time after CIW starts, and we measure the objective value of the best solution obtained by CIW at different time elapsed. For a given elapsed time, it is more preferable to have a larger objective value because this indicates that CIW with the specific x can obtain better solution within the same amount of time.

Figure 12a indicates that setting x too small (i.e., $x = 2$ and $x = 4$) or too large (i.e., $x = 32$) may not always be beneficial because there are too few or too many communities for CIMO. Here, setting $x = 16$ obtains the best solution quality because it strikes a good balance between computation time and ordering capability.

Comparisons of ordering strategies. As mentioned earlier, GIW is a very intuitive approach that acts as a naïve ordering baseline. On the other hand, MaxInner, which performs much better compared to GIW, may incur much computation overhead. This motivates us to consider a more advanced indexing strategy, CIMO, to improve the performance.



(a) CIMO with diff. x on *Email* **(b)** Diff. ordering strategies on *Email*

Fig. 12 Comparisons of ordering strategies

To understand the performance of each ordering strategy, we have added additional experiments to compare the performance of GIW, MaxInner, and CIMO in Fig. 12b. Figure 12b compares the objective values of the three ordering strategies (equipped by CIW), i.e., `GIW`, `MaxInner`, and `CIMO` with time elapsed after the algorithm starts. Here, an ordering approach is considered better if it obtains better solution for the same elapsed time, which can facilitate the willingness pruning to avoid redundant generation of candidate solutions. Figure 12b shows that `GIW` does not perform well because it does not consider the social tightness of the selected users, resulting in many infeasible solutions. In contrast, `CIMO` achieves the best objective values because it employs the offline processing strategy to boost the performance of the proposed `CIW`.

7 Conclusion

In this paper, we introduce a new research problem, *Community-aware Data Acquisition with Maximum Willingness in Online Social Networks (CrawlSN)* to select a community of users who are most willing to contribute their data. We analyze the hardness of CrawlSN and propose the algorithm *Community-aware Group Identification with Maximum Willingness (CIW)* along with effective processing strategies to solve CrawlSN efficiently. We perform an evaluation study with 1093 users to validate CrawlSN, and conduct extensive experiments on real datasets to demonstrate that CIW outperforms other baselines in both solution quality and efficiency. In our future work, we plan to recruit a much larger number of users with a diverse set of backgrounds to train the willingness prediction model to better capture their willingness to contribute their data.

Acknowledgements This work was supported in part by the Ministry of Science and Technology, Taiwan, under Grant MOST 109-2636-E-007-019 and MOST 108-2218-E-468-002.

References

Aksu H, Canim M, Chang Y, Korpeoglu I, Ulusoy O (2014) Distributed k -core view materialization and maintenance for large dynamic graphs. IEEE Trans Knowl Data Eng 26(10):2439–2452

Alvarez-Hamelin J, Dall'Asta L, Barrat A, Vespignani A (2005) K-core decomposition of internet graphs: hierarchies, self-similarity and measurement biases. Networks and Heterogeneous Media 3, Dec

Aridhi S, Brugnara M, Montresor A, Velegrakis Y (2016) Distributed k-core decomposition and maintenance in large dynamic graphs. In: Proceedings of the 10th ACM international conference on distributed and event-based systems, pp 161–168

Balasundaram B, Butenko S, Hicks IV (2011) Clique relaxations in social network analysis: the maximum k-plex problem. Oper Res 59(1):133–142

Blenn N, Doerr C, Van Kester B, Van Mieghem P (2012) Crawling and detecting community structure in online social networks using local information. In Bestak R, Kencl L, Li LE, Widmer J, Yin H (eds) Networking 2012, pp 56–67

Blondel VD, Guillaume J-L, Lambiotte R, Lefebvre E (2008) Fast unfolding of communities in large networks. J Stat Mech Theory Exp 2008(10):P10008

Bond RM, Fariss CJ, Jones JJ, Kramer AD, Marlow C, Settle JE, Fowler JH (2012) A 61-million-person experiment in social influence and political mobilization. Nature 489(7415):295

Candogan O (2019) Persuasion in networks: public signals and k-cores. In Proceedings of the 2019 ACM conference on economics and computation, EC '19, pp 133–134. Association for Computing Machinery

Centola D (2010) The spread of behavior in an online social network experiment. Science 329(5996):1194–1197

Chen W, Wang Y, Yang S (2009) Efficient influence maximization in social networks. In Proceedings of the 15th ACM SIGKDD international conference on knowledge discovery and data mining, pp 199–208

Chen S, Fan J, Li G, Feng J, Tan K-L, Tang J (2015) Online topic-aware influence maximization. Proc VLDB Endow 8(6):666–677

Cheng J, Ke Y, Fu AW-C, Yu JX, Zhu L (2010) Finding maximal cliques in massive networks by h*-graph. In: Proceedings of the 2010 ACM SIGMOD international conference on management of data, pp 447–458

Cui W, Xiao Y, Wang H, Wang W (2014) Local search of communities in large graphs. In: Proceedings of the 2014 ACM SIGMOD international conference on management of data, SIGMOD '14, pp 991–1002

Deutsch M, Gerard HB (1955) A study of normative and informational social influences upon individual judgment. J Abnormal Soc Psychol 51(3):629

Fang Y, Cheng R, Luo S, Hu J (2016) Effective community search for large attributed graphs. Proceedings of the VLDB Endowment 9(12):1233–1244

Giatsidis C, Thilikos DM, Vazirgiannis M (2011) Evaluating cooperation in communities with the k-core structure. In: 2011 international conference on advances in social networks analysis and mining, pp 87–93

Gjoka M, Kurant M, Butts CT, Markopoulou A (2011) Practical recommendations on crawling online social networks. IEEE J Sel Areas Commun 29(9):1872–1892

Gomez-Rodriguez M, Leskovec J, Krause A (2012) Inferring networks of diffusion and influence. ACM Trans Knowl Discov from Data 5(4)

Goodfellow I, Bengio Y, Courville A (2016) Deep learning. The MIT Press, Cambridge

Goyal A, Bonchi F, Lakshmanan LV (2010) Learning influence probabilities in social networks. In: Proceedings of the third ACM international conference on web search and data mining, WSDM '10, pp 241–250

Hsu B, Shen C, Yan X (2019a) Network intervention for mental disorders with minimum small dense subgroups. IEEE Trans Knowl Data Eng. 1–1

Hsu B-Y, Tu C-L, Chang M-Y, Shen C-Y (2019b) On crawling community-aware online social network data. In: Proceedings of the 30th ACM conference on hypertext and social media, pp 265–266

Huang X, Cheng H, Qin L, Tian W, Yu JX (2014) Querying k-truss community in large and dynamic graphs. In: Proceedings of the 2014 ACM SIGMOD international conference on management of data, pp 1311–1322

Huang X, Lakshmanan LV, Yu JX, Cheng H (2015) Approximate closest community search in networks. Proc VLDB Endow 9(4):276–287

Hung H-J, Lee W-C, Yang D-N, Shen C-Y, Lei Z, Chow S-M (2020) Efficient algorithms towards network intervention. In: Proceedings of the web conference 2020

Kempe D, Kleinberg J, Tardos E (2003) Maximizing the spread of influence through a social network. In Proceedings of the ninth ACM SIGKDD international conference on knowledge discovery and data mining, KDD '03, pp 137–146

Kubat M (2015) An introduction to machine learning, 1st edn. Springer, Berlin

Laishram R, Wendt J, Soundarajan S (2019) Crawling the community structure of multiplex networks. In: Proceedings of the AAAI conference on artificial intelligence, vol 33, pp 168–175
Leskovec J, Mcauley JJ (2012) Learning to discover social circles in ego networks. In: Advances in neural information processing systems, pp 539–547
Leskovec J, Lang KJ, Dasgupta A, Mahoney MW (2009) Community structure in large networks: Natural cluster sizes and the absence of large well-defined clusters. Internet Math 6(1):29–123
Li G, Chen S, Feng J, Tan K-l, Li W-s (2014) Efficient location-aware influence maximization. In: Proceedings of the 2014 ACM SIGMOD international conference on management of data, pp 87–98
Li R-H, Qin L, Yu JX, Mao R (2015) Influential community search in large networks. Proc VLDB Endow 8(5):509–520
Li Y, Zhang D, Tan K-L (2015) Real-time targeted influence maximization for online advertisements. Proc VLDB Endow 8(10):1070–1081
Li J, Wang X, Deng K, Yang X, Sellis T, Yu JX (2017) Most influential community search over large social networks. In: 2017 IEEE 33rd international conference on data engineering, pp 871–882
Lu W, Bonchi F, Goyal A, Lakshmanan LV (2013) The bang for the buck: fair competitive viral marketing from the host perspective. In: Proceedings of the 19th ACM SIGKDD international conference on knowledge discovery and data mining, pp 928–936
Mokken RJ (1979) Cliques, clubs and clans. Quality & Quantity 13(2):161–173
Mucha PJ, Richardson T, Macon K, Porter MA, Onnela J-P (2010) Community structure in time-dependent, multiscale, and multiplex networks. Science 328(5980):876–878
Reproducibility materials. http://www.cs.nthu.edu.tw/~chihya/CIW_download/, 2020
Seidman SB (1983) Network structure and minimum degree. Soc Netw 5(3):269–287
Shen C-Y, Yang D-N, Huang L-H, Lee W-C, Chen M-S (2016) Socio-spatial group queries for impromptu activity planning. IEEE Trans Knowl Data Eng 28(1):196–210
Shen C-Y, Huang L-H, Yang D-N, Shuai H-H, Lee W-C, Chen M-S (2017) On finding socially tenuous groups for online social networks. In: Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining, pp 415–424
Shen C-Y, Fotsing CPK, Yang D-N, Chen Y-S, Lee W-C (2018) On organizing online soirees with live multi-streaming. In: AAAI conference on artificial intelligence
Shin K, Eliassi-Rad T, Faloutsos C (2016) Corescope: Graph mining using k-core analysis—patterns, anomalies and algorithms. In: 2016 IEEE 16th international conference on data mining, pp 469–478
Shuai H-H, Yang D-N, Yu PS, Chen M-S (2013) Willingness optimization for social group activity. Proc VLDB Endow 7(4):253–264
Song C, Hsu W, Lee ML (2017) Temporal influence blocking: Minimizing the effect of misinformation in social networks. In: 2017 IEEE 33rd international conference on data engineering, pp 847–858
Wang K, Cao X, Lin X, Zhang W, Qin L (2018) Efficient computing of radius-bounded k-cores. In: 2018 IEEE 34th international conference on data engineering (ICDE), pp 233–244
Yang J, Leskovec J (2015) Defining and evaluating network communities based on ground-truth. Knowl Inf Syst 42(1):181–213
Yang D-N, Shen C-Y, Lee W-C, Chen M-S (2012) On socio-spatial group query for location-based social networks. In: Proceedings of the 18th ACM SIGKDD international conference on knowledge discovery and data mining, KDD '12, pp 949–957
Yang D-N, Hung H-J, Lee W-C, Chen W (2013) Maximizing acceptance probability for active friending in online social networks. In: Proceedings of the 19th ACM SIGKDD international conference on knowledge discovery and data mining, pp 713–721
Yang Y, Mao X, Pei J, He X (2016) Continuous influence maximization: What discounts should we offer to social network users? In: Proceedings of the 2016 international conference on management of data, pp 727–741
Ye S, Lang J, Wu F (2010) Crawling online social graphs. In: 2010 12th international Asia-Pacific web conference, pp 236–242
Zhang Y, Parthasarathy S (2012) Extracting analyzing and visualizing triangle k-core motifs within networks. In: 2012 IEEE 28th international conference on data engineering, pp 1049–1060
Zhang F, Zhang W, Zhang Y, Qin L, Lin X (2017) Olak: an efficient algorithm to prevent unraveling in social networks. Proc VLDB Endow 10(6):649–660
Zhang F, Zhang Y, Qin L, Zhang W, Lin X (2017) When engagement meets similarity: efficient (k, r)-core computation on social networks. Proc VLDB Endow 10(10):998–1009

Zhu Q, Hu H, Xu C, Xu J, Lee W-C (2017) Geo-social group queries with minimum acquaintance constraints. VLDB J 26(5):709–727

Affiliations

Bay-Yuan Hsu[1] · Chia-Lin Tu[2] · Ming-Yi Chang[2] · Chih-Ya Shen[2]

Bay-Yuan Hsu
byhsu@mail.ntpu.edu.tw

Chia-Lin Tu
cltu@lbstr.cs.nthu.edu.tw

Ming-Yi Chang
changmy@mx.nthu.edu.tw

1 Department of Computer Science, National Taipei University, New Taipei City, Taiwan

2 Department of Computer Science, National Tsing Hua University, Hsinchu, Taiwan