This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2015.2468728, IEEE Transactions on Knowledge and Data Engineering

A Comprehensive Study on Willingness Maximization for Social Activity Planning with Quality Guarantee

Hong-Han Shuai, De-Nian Yang, *Senior Member, IEEE,* Philip S. Yu, *Fellow, IEEE* and Ming-Syan Chen, *Fellow, IEEE*

Abstract—Studies show that a person is willing to join a social group activity if the activity is interesting, and if some close friends also join the activity as companions. The literature has demonstrated that the interests of a person and the social tightness among friends can be effectively derived and mined from social networking websites. However, even with the above two kinds of information widely available, social group activities still need to be coordinated manually, and the process is tedious and time-consuming for users, especially for a large social group activity, due to complications of social connectivity and the diversity of possible interests among friends. To address the above important need, this paper proposes to automatically select and recommend potential attendees of a social group activity, which could be very useful for social networking websites as a value-added service. We first formulate a new problem, named Willingness mAximization for Social grOup (WASO). This paper points out that the solution obtained by a greedy algorithm is likely to be trapped in a local optimal solution. Thus, we design a new randomized algorithm to effectively and efficiently solve the problem. Given the available computational budgets, the proposed algorithm is able to optimally allocate the resources and find a solution with an approximation ratio. We implement the proposed algorithm in Facebook, and the user study demonstrates that social groups obtained by the proposed algorithm significantly outperform the solutions manually configured by users.

Index Terms—Social network, query processing, optimization

1 Introduction

Studies show that two important criteria are usually involved in the decision of a person joining a group activity [8], [14] at her available time. First, the person is interested in the intrinsic properties of the activity, which may be in line with her favorite hobby or exercise. Second, other people who are important to the person, such as her close friends, will join the activity as companions[1]. For example, if a person who appreciates abstract art has complimentary tickets for a modern art exhibition at MoMA, she would probably want to invite her friends and friends of friends with this shared interest. Nowadays, many people are accustomed to sharing information with their friends on social networking websites, like Facebook, Meetup, Plancast, and LikeALittle, and a recent line of studies [5], [18] has introduced effective algorithms to quantify the interests of a person according to the interest attributes in her personal profile and the contextual information in her interaction with friends. Moreover, social connectivity models have been widely studied [3] for evaluating the tightness between two friends in the above websites. Nonetheless, even with the above knowledge available, to date there has been neither published work nor a real system explores how to leverage the above two crucial factors for *automatic planning and recommending* of a *group activity*, which is potentially very useful for social networking websites as a value-added service[2]. For example, Meetup has 20.76 million active users, and 191,430 groups, thus creating 517,446 social events and 3.68 million RSVPs every month. At present, many social networking websites only act as a platform for information sharing and exchange in activity planning. The attendees of a group activity still need to be selected manually, and such manual coordination is usually tedious and time-consuming, especially for a large social activity, given the complicated link structure in social networks and the diverse interests of friends.

To solve this problem, this paper makes an initial attempt to incorporate the interests of people and their social tightness as two key factors to find a group of attendees for automatic planning and recommendation. It is desirable to choose more attendees who like and enjoy the activity and to invite more friends with the shared interest in the

- *Hong-Han Shuai, De-Nian Yang, and Ming-Syan Chen are with the Research Center of Information Technology Innovation, Academia Sinica, No. 128, Sec. 2, Academia Road, Taipei, 11529 Taiwan. Email: {hhshuai, dnyang, mschen}@citi.sinica.edu.tw.*
- *Hong-Han Shuai and Ming-Syan Chen are also with the Graduate Institute of Communication Engineering, National Taiwan University, No. 1, Sec. 4, Roosevelt Road, Taipei, 10617 Taiwan.*
- *Philip S. Yu is with the Department of Computer Science, University of Illinois at Chicago, IL, USA. E-mail: psyu@uic.edu.*

1. There are other criteria that are also important, e.g., activity time, and activity location. However, to consider the above factors, a promising way is to preprocess and filter out the people who are not available, live too far, etc.

2. The privacy of a person in automatic activity planning can follow the current privacy setting policy in social networking websites when the person subscribes the service. The details of privacy setting are beyond the scope of this paper.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2015.2468728, IEEE Transactions on Knowledge and Data Engineering

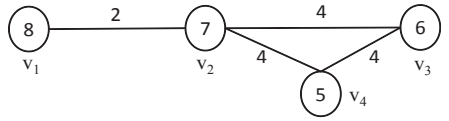

Fig. 1. Counterexample of DGreedy

activity as companions. In fact, Psychology [8], [14] and recent study in social networks [23], [24] have modeled the *willingness* to attend an activity or a social event as the sum of the interest of each attendee on the activity and the social tightness between friends that are possible to join it. It is envisaged that the selected attendees are more inclined to join the activity if the willingness of the group increases.

With this objective in mind, we formulate a new fundamental optimization problem, named *Willingness mAximization for Social grOup (WASO)*. The problem is given a social graph G, where each node represents a candidate person and is associated with an interest score of the person for the activity, and each edge has a social tightness score to indicate the mutual familiarity between the two persons. Let k denote the number of expected attendees. Given the user-specified k, the goal of automatic activity planning is to maximize the willingness of the selected group F, while the induced graph on F is a connected subgraph for each attendee to become acquainted with another attendee according to a social path[3]. For the activities without an *a priori* fixed size, it is reasonable for a user to specify a proper range for the group size, and our algorithm can find the solution for each k within the range and return the solutions for the user to decide the most suitable group size and the corresponding attendees.[4]

Naturally, to incrementally construct the group, a deterministic greedy algorithm (DGreedy) sequentially chooses an attendee that leads to the largest increment in the willingness at each iteration. For example, Figure 1 presents an illustrative example with $k = 3$. Node v_1 is first selected since its interest score is the maximum one among all nodes. Afterward, node v_2 is then extracted. Finally, v_3, instead of v_4, is chosen because it generates the largest increment on willingness, i.e., 10, and leads to a group with a willingness of 27. Note the DGreedy, though simple, tends to be trapped in a local optimal solution, since it facilitates the selection of nodes only suitable at the corresponding iterations. In this simple example, the above algorithm is not able to find the optimal solution because it makes a greedy selection at each iteration and only chooses v_1 as the start node, who enjoys the activity the most at the first iteration, but the optimal solution is $\{v_2, v_3, v_4\}$ with the total willingness being 30.

Another approach is to examine the willingness of every possible combination of k attendees. However, this enumeration approach needs to evaluate C_k^n candidate groups, where n is the number of nodes in G. In current social networking websites, the number of candidate groups is still huge even when we focus on only the candidates located in the same area, e.g., about ten thousand users in British Virgin Islands[5]. When $k = 50$, the number of candidate groups is in the order of 10^{135}. Thus, this approach is computationally intractable for a massive social network.

Indeed, we show that the problem is challenging and prove that it is NP-hard. As shown in Figure 1, DGreedy improperly chooses v_1 as the start node and explores only a single sequence of nodes in the solution space. To increase the search space, randomized algorithms have been proposed as a simple but effective strategy to solve the problems with large instances [19]. To avoid being trapped in a local optimal solution, a simple randomized algorithm for WASO is to randomly choose multiple start nodes. Each start node is considered as partial solution, and a node neighboring the partial solution is randomly chosen and added to the partial solution at each iteration afterward, until k nodes are included as a final solution. This randomized algorithm is more efficient than DGreedy, because the computation of willingness is not involved during the selection of a node. For the problem with a large k, numerous candidate nodes neighboring the partial solution are necessary to be examined in DGreedy to sum up the willingness, in order to find the one that generates the largest willingness. In contrast, the randomized algorithm simply chooses one neighboring node at random.

With randomization, the aforementioned algorithm is able to effectively avoid being trapped in a local optimal solution. It suffers, however, two disadvantages. Firstly, a start node that has the potential to generate final solutions with high willingness is not invested with more computational budgets for randomization in the following iterations. Each start node in the randomized algorithm is expanded to only one final solution. Thus, a start nodes, which has the potential to grow and become the solution with high willingness, may fail to generate a final solution with high willingness because only one solution is randomly constructed and expanded from the start node. The second disadvantage is that the expansion of the partial solution does not differentiate the selection of the neighboring nodes. Each neighboring node is treated equally and chosen uniformly at random for each iteration. In contrast, a simple way to remedy this issue is to assign the probability to each neighboring node according to its interest score and social tightness of incident edges. However, this assignment is similar to the DGreedy in that it limits the scope to the local information corresponding to each node and is not expected to generate a solution with high willingness.

Keeping in mind the above observations in an effort to guide an efficient search of the solution space, we propose two randomized algorithms, called *CBAS* (Computational Budget Allocation for Start nodes) and *CBAS-ND* (Computation Budget Allocation for Start nodes with Neighbor Differentiation), to address the above two crucial factors in selecting start nodes and expanding the partial solutions, respectively. This paper exploits the notion of Optimal Computing Budget Allocation (OCBA) [4] in ran-

3. For some group activities, it is not necessary to ensure that the solution group is a connected subgraph. Later in Section 2, we will show that WASO without a connectivity constraint can be easily solved by the proposed algorithm with simple modification.

4. The parameter settings in WASO to fit varied scenarios in everyday life will be introduced in more details in Appendix D.

5. http://www.socialbakers.com/facebook-statistics/.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2015.2468728, IEEE Transactions on Knowledge and Data Engineering

domization, in order to optimally invest more computational budgets in the start nodes with the potential to generate the solutions with high willingness. *CBAS* first selects m start nodes[6] and then randomly adds neighboring nodes to expand the partial solution stage-by-stage, until k nodes are included as a final solution. Each start node in *CBAS* is expanded to multiple final solutions. To properly invest the computational budgets, *CBAS* at each stage identifies the start nodes worth more computational budgets according to sampled results of the previous stages. Equipped with the allocation strategy of computational resources, *CBAS* is enhanced to *CBAS-ND* to adaptively assign the probability to each neighboring node during the expansion of the partial solution according to the cross entropy method. We prove that the allocation of computational budgets for start nodes and the assignment of the probability to each node are both optimal in *CBAS* and *CBAS-ND*, respectively. We further show that *CBAS* can achieve an approximation ratio, while *CBAS-ND* needs much smaller computational budgets than *CBAS* to acquire the same solution quality.

The contributions of this paper can be summarized as follows.

- We formulate a new optimization problem, namely WASO, to consider the topic interest of users and social tightness among friends for automatic planning of activities. We prove that WASO is NP-hard. To the best of the authors' knowledge, there is no real system or existing work in the literature that addresses the issue of automatic activity planning based on both topic interest and social relationship.
- We design Algorithm *CBAS* and *CBAS-ND* to find the solution to WASO with an approximation ratio. Experimental results on three real datasets demonstrate that the solution returned by *CBAS-ND* is very close to the optimal solution obtained by IBM CPLEX, which is widely regarded as the fastest general parallel optimizer, and *CBAS-ND* is faster than CPLEX. Also, we provide the Integer Programming (IP) formulation for finding the optimal solution of WASO.
- We implement *CBAS-ND* in Facebook and conduct a user study with 137 people. Currently, people are used to organizing an activity manually without being aware of the quality of the organized group, because there is no automatic group recommendation service available for comparison. Compared with the manual solutions, we observe that the solutions obtained by *CBAS-ND* are 50.6% better. In addition, 98.5% of users conclude that the group recommended by *CBAS-ND* is better or acceptable. Therefore, this research result has the potential to be adopted in social networking websites as a value-added service.

The rest of this paper is summarized as follows. Section 2 formulates WASO, surveys the related works, and provides the Integer Programming formulation for finding the optimal solution. Sections 3 and 4 explain *CBAS* and *CBAS-ND* and derive the approximation ratio. User study and experimental results are presented in Section 5, and we conclude this paper in Section 6.

6. The setting of m and other parameters is important and will be studied in the end of Section 5.

2 Preliminary

2.1 Problem Definition

Given a social network $G = (V, E)$, where each node $v_i \in V$ and each edge $e_{i,j} \in E$ are associated with an interest score η_i and a social tightness score $\tau_{i,j}$ assigned according to the literature [5] and [3] respectively, this paper studies a new optimization problem WASO for finding a set F of vertices with size k to maximize the willingness $W(F)$, i.e.,

$$\max_F W(F) = \max_F \sum_{v_i \in F} (\eta_i + \sum_{v_j \in F : e_{i,j} \in E} \tau_{i,j}), \quad (1)$$

where F is a connected subgraph in G to encourage each attendee to be acquainted with another attendee according to a social path in F. Notice that the social tightness between v_i and v_j is not necessarily symmetric, i.e., $\tau_{i,j}$ can be different with $\tau_{j,i}$. Therefore, the willingness in Eq. (1) considers both $\tau_{i,j}$ and $\tau_{j,i}$. It is worth noting that the illustrating example in the paper is symmetric for simplicity. As demonstrated in the previous works in Psychology and social networks [23], [24] that jointly consider the social and interest domains, the willingness of a group is represented as the sum of the topic interest of nodes and social tightness between them[7].

Notice that the network with η as 0 or τ as 0 is a special case of WASO. Previous works [8], [14] demonstrated that both social tightness and topic interest are intrinsic criteria involved in the decision of a person to join a group activity, which is in line with the results of our user study presented in Section 5. WASO is challenging due to the tradeoff in interest and social tightness, while the constraint that assures that the F is connected also complicates this problem because it is no longer able to arbitrarily choose any nodes from G. Indeed, the following theorem shows that WASO is NP-hard.

Theorem 1: WASO is NP-hard.

Proof: We prove that WASO is NP-hard with the reduction from Dense k-Subgraph (DkS) [9]. Given a graph $G_D = (V_D, E_D)$, DkS finds a subgraph with k nodes F_D to maximize the density of the subgraph. In other words, the purpose of DkS is to maximize the number of edges $E(F_D)$ in the subgraph induced by the selected nodes. For each instance of DkS, we construct an instance for WASO by letting $G = G_D$, where η_i of each node $v_i \in V$ is set as 0, and $\tau_{i,j}$ of each edge $e_{i,j} \in E$ is assigned as 1. We first prove the sufficient condition. For each instance of DkS with solution node set F_D, we let $F = F_D$. If the number of edges $E(F_D)$ in the subgraph of DkS is ϵ, the willingness of WASO $W(F)$ is also ϵ because $F = F_D$. We then prove the necessary condition. For each instance

7. Different weights λ and (1-λ) can be assigned to the interest scores and social tightness such that $W(F) = \sum_{v_i \in F}(\lambda_i \eta_i + (1 - \lambda_i) \sum_{v_j \in F : e_{i,j} \in E} \tau_{i,j})$. λ_i can be set directly by a user or according to the existing model [24]. The impacts of different λ will be studied later in Section 5.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2015.2468728, IEEE Transactions on Knowledge and Data Engineering

of WASO with F, we select the same nodes for F_D, and the number of edges $E(F_D)$ must be the same as $W(F)$. The theorem follows. □

2.2 Related Works

Given the growing importance of varied social networking applications, there has been a recent push on the study of user interest scores and social tightness scores from real social networking data. It has been demonstrated that unknown user interest attributes can be effectively inferred from a social network according to the revealed attributes of the friends [18]. On the other hand, Wilson *et al.* [22] derived a new model to quantify the social tightness between any two friends in Facebook. The number of wall postings is also demonstrated to be an effective indicator for social tightness [11]. Thus, the above studies provide a sound foundation to quantify the user interest and social tightness scores in social networks. Moreover, Yang [23] and Lee [24] sum up the two factors as willingness for marketing and recommendation. Nevertheless, the above factors crucial in social networks have not been leveraged for automatic activity planning explored in this paper.

Expert team formation in social networks has attracted extensive research interests. The problem of constructing an expert team is to find a set of people owning the specified skills, while the communications cost among the chosen friends is minimized to ensure the rapport among the team members for an efficient operation. Two communications costs, diameter and minimum spanning tree, were evaluated. Several extended models have been studied. For example, each skill i needs to contain at least k_i people in order to form a strong team [17], while all-pair shortest paths are incorporated to describe the communications costs more precisely [15]. Moreover, a skill leader is selected for each skill with the goal to minimize the social distance from the skill members to each skill leader [15], while the density of a team is also considered [10].

In addition to expert team formation, community detection as well as graph clustering and graph partitioning have been explored to find groups of nodes mostly based on the graph structure [1]. The quality of an obtained community is usually measured according to the structure inside the community, together with the connectivity within the community and between the rest of the nodes in the graph, such as the density of local edges, deviance from a random null model, and conductance. Sozio et al. [21], for example, detected community by minimizing the total degree of a community with specified nodes. Seshadhri et al. [12] propose conductance, which not only takes cuts into account, i.e., the edge number between separated sets, but also the orders of the sets that are being cut apart, and thus is able to yield more significant separations. However, the objective function of WASO is different from community detection. Each node and each edge in WASO are associated with an interest score and social tightness score in the problem studied in this paper, in order to maximize the willingness of the attendees with a specified group size, which can be very useful for social networking websites as a value-added service.

Moreover, given an undirected graph with a prize associated with each node and a weight associated with each edge, the prize-collecting Steiner tree problem aims at finding a subtree (i.e., selecting both the nodes and edges) in the graph so as to maximize the total prize of all nodes subtracted by the total weight of all edges in the tree. To solve the problem, Hajiaghayi [13] proposed a $\frac{1}{1-\sqrt{\frac{1}{e}}}$-approximation algorithm with randomized LP rounding. However, the output of PCST is different from WASO. PCST selects not only a set of nodes with high prizes but also a set of edges (to form a tree) with small weights. In contrast, WASO extracts only a set of nodes, and every edge connecting to any two nodes in the solution is included in the objective function.

2.3 Integer Programming for WASO

In the following, we describe the Integer Programming (IP) formulation for WASO. Binary variable x_i denotes if node v_i is selected in the solution F, and binary variable $y_{i,j}$ denotes if two neighboring nodes v_i and v_j are both selected in F. The objective function is

$$\max \sum_{v_i \in V} \eta_i x_i + \sum_{e_{i,j} \in E} \tau_{i,j} y_{i,j},$$

where the first term is the total interest score, and the second term is the total social tightness score of the selected nodes. The basic constraints of WASO include

$$\sum_{v_i \in V} x_i = k \tag{2}$$

$$x_i + x_j \geq 2y_{i,j}, \forall v_i \in V, \forall v_j \in N_i \tag{3}$$

Constraint (2) states that exactly k nodes are selected in F, while constraint (3) ensures that the social tightness score $\tau_{i,j}$ of any edge $e_{i,j}$ can be added to the objective function (i.e., $y_{i,j} = 1$) only when the two terminal nodes v_i and v_j are both selected (i.e., $x_i = x_j = 1$); otherwise, $y_{i,j}$ are enforced to be 0.

However, the above basic constraints cannot guarantee that F is a connected component of G, since nodes are allowed to be chosen arbitrarily. To effective address the issue, we propose the following advanced constraints for WASO to ensure that there is a path from a root node in F to every other selected node in F, where all nodes in the path must also belong to F. More specifically, let binary variable r_i denote if node v_i is the root node, and let binary variable $p_{i,j,m,n}$ denote if edge $e_{m,n}$ in E is located in the path from root node r_i to another node v_j in F. It is worth noting that since F is unknown, variables r_i and $p_{i,j,m,n}$ in the advanced constraints are correlated to x_i and x_j, respectively.

WASO contains the following advanced constraints.

$$\sum_{v_i \in V} r_i = 1 \tag{4}$$

$$r_i \leq x_i, \forall v_i \in V \tag{5}$$

Constraint (4) states that only one root node will be selected, while constraint (5) guarantees that the selected

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2015.2468728, IEEE Transactions on Knowledge and Data Engineering

root node must appear in F (i.e, $r_i = 1$ only when $x_i = 1$). Equipped with the root node r_i, let N_j denote the set of neighboring node of v_j, let $d_{i,j,m}$ denote the maximal number of edges in the path from r_i to v_m with v_j as the destination of the path, and the following four constraints identify the path from r_i to every node v_j in F.

$$r_i + x_j - 1 = \sum_{n \in N_i} p_{i,j,i,n}, \forall v_i, v_j \in V, v_i \neq v_j \quad (6)$$

$$r_i + x_j - 1 = \sum_{m \in N_j} p_{i,j,m,j}, \forall v_i, v_j \in V, v_i \neq v_j \quad (7)$$

$$\sum_{q \in N_m} p_{i,j,q,m} = \sum_{n \in N_m} p_{i,j,m,n}, \forall v_i, v_j, v_m \in V, v_i \neq v_j, v_i \neq v_m, v_j \neq v_m \quad (8)$$

$$d_{i,j,m} + (p_{i,j,m,n} - 1)\,|V| < d_{i,j,n}, \forall v_i, v_j \in V, \forall e_{m,n} \in E \quad (9)$$

For the selected root node r_i and every other node v_j in F (i.e., $x_j = 1$), the left hand side (LHS) of constraints 6 and 7 become 1, enforcing that at least one incident edge $e_{i,n}$ of v_i and one incident edge $e_{m,j}$ of v_j must be included in the path. After obtaining the first and last edge (i.e., $e_{i,n}$ and $e_{m,j}$) in the path from r_i to v_j, constraint 8 is a flow continuity constraint. For each node v_m, if it is an intermediate node in the path, flow continuity constraint states that the flow from r_i to v_m must be identical to the flow v_m to v_j. In other words, constraint 8 chooses a parent node v_q and a child node v_n for v_m in the path

Constraint 9 guarantees that the node sequence in the path contains no cycle; otherwise, for every edge $e_{m,n}$ in the cycle, $p_{i,j,m,n} = 1$, and the following inequality holds,

$$d_{i,j,m} < d_{i,j,n}$$

and it is thus impossible to find a $d_{i,j,n}$ for every node v_n in the cycle. On the other hand, for any edge with $p_{i,j,m,n} = 0$, the constraint becomes redundant since $d_{i,j,m} - |V| < d_{i,j,n}$ always holds.

The following constraint ensures that every two terminal nodes v_m and v_n of an edge $e_{m,n}$ in the path (i.e., $p_{i,j,m,n} = 1$) must participate in F (i.e., $x_m = x_n = 1$).

$$p_{i,j,m,n} \leq 2(x_m + x_n), \forall v_i, v_j \in V, \forall e_{m,n} \in E \quad (10)$$

Therefore, it is not allowed to arbitrarily choose a path in G to connect the root node r_i to another node v_j in F.

It is worth noting that the computational complexity of finding the optimal solution is too high for real applications since WASO is an NP-hard problem (unless P=NP). Therefore, IP is only suitable for very small datasets or candidate groups of small sizes.

3 Algorithm Design for WASO

To solve WASO, DGreedy incrementally constructs the solution by sequentially choosing an attendee that leads to the largest increment in the willingness at each iteration. However, while this approach is simple, the search space of DGreedy is limited because a single sequence of nodes is explored. Moreover, the algorithm is inclined to be trapped in the local maximum [16].[8] To address the above issues, this paper first proposes a randomized algorithm *CBAS* to randomly choose m start nodes. Each start node acts as a seed to be expanded to multiple final solutions. At each iteration, a partial solution, which consists of only a start node at the first iteration or a connected set of nodes at any iteration afterward, is expanded by uniformly selecting at random a node neighboring the partial solution, until k nodes are included. We leverage the notion of Optimal Computing Budget Allocation (OCBA) [4] to randomly generate more final solutions from each start node that has more potential to generate the final solutions with high willingness. Later we will prove that the number of final solutions generated from each start node is optimally assigned.

After this, we enhance *CBAS* to *CBAS-ND* by differentiating the selection of the nodes neighboring each partial solution. During each iteration of *CBAS*, each neighboring node is treated equally and chosen uniformly at random. A simple way to improve *CBAS* is to associate each neighboring node with a different probability according to its interest score and social tightness scores of incident edges. Yet, this assignment is similar to DGreedy insofar as it limits the scope to only the local information associated with each node thereby making it difficult to generate a final solution with high willingness. To prevent the generation of only a local optimal solution, *CBAS-ND* deploys the cross entropy method according to results at the previous stages in order to optimally assign a probability to each neighboring node.

One advantage of the proposed randomized algorithms is that the tradeoff between the solution quality and execution time can be easily controlled by assigning different T, which denotes the number of randomly generated final solutions. Under a given T, if m start nodes are generated, the above algorithms can optimally divide T into m parts for the m start nodes to find final solutions with high willingness. Moreover, we prove that *CBAS* is able to find a solution with an approximation ratio. Compared with *CBAS*, we further prove that the solution quality of *CBAS-ND* is better with the same computation budget[9]. The detailed settings of T and m will be analyzed in Section 5.

In the following, we first present *CBAS* to optimally allocate the computational budgets to different start nodes (Section 3.1) and then derive the approximation ratio in Section 3.2. Algorithm *CBAS-ND* will be presented in Section 4.

3.1 Allocation of Computational Budget for Start Nodes

Given the total computational budgets T specified by users, a simple approach first randomly selects m start nodes and then expands each start node to $\frac{T}{m}$ final solutions.

8. The theoretical analysis of DGreedy is provided in Appendix B.

9. It is worth noting that randomization is performed only in expanding a start node to a final solution, not in the selection of a start node. This is because the approximation ratio is not able to be achieved if a start node is decided randomly.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2015.2468728, IEEE Transactions on Knowledge and Data Engineering

TABLE 1
Parameter Summary

Notation	Description
J_i	a random variable that represents the value sampled from start node v_i
J_i^*	a random variable that represents the sample maximum from start node v_i
c_i	the smallest willingness expanded from start node v_i
d_i	the largest willingness expanded from start node v_i
c_b	the smallest willingness expanded from start node v_b which generates the best solution so far
$\tau_{i,j}$	social tightness score between node v_i and v_j
η_i	interest score of node v_i
λi	weighting between interest score and tightness score of node v_i
T	total computation budget
m	number of start nodes

However, this homogeneous approach does not give priority to the start nodes that have more potential to generate final solutions with high willingness. In contrast, *CBAS* optimally allocates more resources to the start nodes with high willingness with the following phases.

1) *Selection and Evaluation of Start Nodes:* This phase first selects m start nodes according to the interest scores and social tightness scores. Afterward, each start node is randomly expanded to a few final solutions. We iteratively select and add a neighboring node uniformly at random to a partial solution, until k nodes are selected. The willingness of each final solution is evaluated for the next phase to allocate different computational budgets to different start nodes.
2) *Allocation of Computational Budgets:* This phase derives the computational resources optimally allocated to each start node according to the previous sampled willingness.

The summary of used notations is shown in Table 1 for better reading. To optimally allocate the computational budgets for each start node, we first define the solution quality as follows.

Definition 1: The solution quality, denoted by Q, is defined as the maximum willingness among all maximal sampled results of the m start nodes,

$$Q = max\{J_1^*, J_2^*, ..., J_i^*, ..., J_m^*\},$$

where J_i^* is a random variable representing the maximal willingness sampled from a final solution expanded from start node v_i.

Since the maximal sampled result J_i^* of start node v_i is related to the number of sampling times N_i, i.e., the number of final solutions randomly generated from v_i, the mathematical formulation to optimize the computational budget allocation is defined as

$$\max_{N_1,N_2,...,N_m} Q,$$

$$\text{s.t. } N_1 + N_2 + ... + N_m = T.$$

Let v_b denote the start node that are able to generate the solution with the highest willingness. Obviously, the optimal solution in the above maximization problem is to allocate all the computational budgets to v_b. However, since v_b is not given *a priori*, *CBAS* divides the resource allocation into r stages, and each stage adjusts the allocation of computational budgets $\frac{T}{r}$ to different start nodes according to the sampled willingness from the partial solutions in previous stages.

For each node, phase 1 of *CBAS* first adds the interest score and the social tightness scores of incident edges and then chooses the m nodes with the largest sums as the m start nodes. On the other hand, allocating more computational budgets to the start node with a larger sum, similar to DGreedy, does not tend to generate a final solution with high willingness. For this reason, phase 2 evaluates the sampled willingness to allocate different computational budgets to each start node.

In stage t of phase 2, let $N_{i,t}$ denote the computational budgets allocated to start node v_i at the t-th stage. The ratio of computational budgets $N_{i,t}$ and $N_{j,t}$ allocated to any two start nodes v_i and v_j is

$$\frac{N_{i,t}}{N_{j,t}} = (\frac{d_i - c_b}{d_j - c_b})^{N_b},$$

where c_i and d_i are the true smallest and largest willingness values for the groups expanded from start node v_i as described in Table 1. Nevertheless, since c_i and d_i cannot be derived unless infinite budgets are given, CBAS and CBAS-ND follow the OCBA theory [4] to approximate them by the smallest/largest sampled values with finite budgets. Notice that v_b here is the start node that enjoys the highest willingness sampled in the previous stages, N_b is the overall computational budgets allocated to v_b in the previous stages, and c_b denotes the worst sampled willingness of the partial solution expanded from start node v_b in the previous stages. Later, we will prove that the above budgets allocation in each stage is optimal. However, if the allocated computational budgets for a start node is 0 at the t-th stage, we prune off the start node in the following $(t+1)$-th stage.

3.2 Theoretical Result of CBAS with Uniform Distribution

To correctly allocate the computational budgets T to m start nodes, we first derive the optimal ratio of computational budgets for any two start nodes. Afterward, we find the probability P_b that node v_b is actually the start node which is able to generate the highest willingness in each stage. Finally, we derive the approximation ratio. The complexity of *CBAS* is provided in Appendix E.

Definition 2: A random variable, denoted as J_i, is defined to be the value sampled from start node v_i.

Definition 3: A random variable, denoted as J_i^*, is defined to be the sample maximum from start node v_i.

Notice that the sample maximum and sample minimum are also called the largest observation is the values of the greatest element of a sample. For example, if the sample values in start node v_i are 5, 3, and 7, the sample maximum will be 5, 5, and 7. The literature of OCBA indicates that the distribution of random variable J_i in most applications is a normal distribution, but the allocation results are very

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2015.2468728, IEEE Transactions on Knowledge and Data Engineering

close to the one with the uniform distribution [4], [7]. Therefore, given space constraints, J_i here is first handled as the uniform distribution in $[c_i, d_i]$, and the derivation for the normal distribution is presented in Section 3.3. The probability density function and cumulative distribution function are formulated as

$$p_{J_i}(x) = \begin{cases} \frac{1}{d_i - c_i} & \text{if } c_i \le x \le d_i \\ 0 & \text{otherwise.} \end{cases}$$

$$P_{J_i}(x) = \begin{cases} 0 & \text{if } x \le c_i. \\ \frac{x - c_i}{d_i - c_i} & \text{if } c_i \le x \le d_i. \\ 1 & \text{otherwise.} \end{cases}$$

Therefore, for the maximal value J_i^*,

$$p_{J_i^*}(x) = N_i P_{J_i}(x)^{N_i - 1} p_{J_i}(x),$$

$$P_{J_i^*}(x) = P_{J_i}(x)^{N_i}.$$

Theorem 2: Given the best start node v_b, the probability that J_i^* exceeds J_b^* is at most $\frac{1}{2}(\frac{d_i - c_b}{d_b - c_b})^{N_b}$.
The proof is presented in Appendix A. With the result above, we allocate the computational budgets by

$$\frac{N_i}{N_j} = \frac{P(J_i^* \ge J_b^*)}{P(J_j^* \ge J_b^*)} = (\frac{d_i - c_b}{d_j - c_b})^{N_b}. \quad (11)$$

Since it is impossible to enumerate every final solution expanded from a start node, the ratio of the computational budget allocation is optimal in OCBA [4] if the first equality in Eq. (11) holds. Thus, it is optimal to allocate the computational budgets to N_i and N_j according to the ratio $(\frac{d_i - c_b}{d_j - c_b})^{N_b}$. Notice that if d_i is smaller than c_b, the probability that J_b^* is smaller than J_i^* is zero.

Intuitively, the above result indicates that if the best random sample, i.e., d_i, from a start node is small, it is unnecessary to repeat the sampling process too many times since the users nearby the start node are not really interested in the activity or they have an estranged friendship. On the other hand, as the number of sample times increases, it is expected that the identified best start node enjoys the highest willingness.

The following theorem first analyzes the probability P_b that v_b, as decided according to the samples in the previous stages, is actually the start node that generates the highest willingness. Let α denote the closeness ratio between the maximum of the start node with the highest willingness and the maximum of other start nodes, i.e., $\alpha = (d_a - c_b)/(d_b - c_b)$, where v_a generates the maximum willingness among other start nodes. Therefore, in addition to 0 and 1, α is allowed to be any other value from 0 to 1.

Theorem 3: For WASO with parameter (m, T), where m is the number of start nodes and T is the total computational budgets, the probability P_b that v_b is selected according to the previous stages is actually the start node with the highest willingness is at least $1 - \frac{1}{2}(m-1)\alpha^{\frac{T}{mr}}$.
The proof is presented in Appendix A. Given the total budgets T, the following theorem derives a lower bound of the solution obtained by *CBAS*.

Theorem 4: For a WASO optimization problem with r-stage computational budget allocation, the maximum willingness $E[Q]$ from the solution of *CBAS* is at least $N_b(\frac{1}{N_b+1})^{\frac{N_b+1}{N_b}} \cdot Q^*$, where N_b after r stages is $\frac{4+m(r-1)}{4rm}T$, and Q^* is the optimal solution.

Proof: We first derive the lower bound of $E[Q]$ as follows. The random variable Q is denoted as $max\{J_1^*, ..., J_m^*\}$. The cumulative density function is

$$\begin{aligned} F_Q(Q \le \Delta) &= F(max\{J_1^*, ..., J_m^*\} \le \Delta) \\ &= F(J_1^* \le \Delta, J_2^* \le \Delta, ..., J_m^* \le \Delta) \\ &= F_{J_1^*}(\Delta) F_{J_2^*}(\Delta) ... F_{J_m^*}(\Delta) \\ &= (\frac{\Delta - c_1}{d_1 - c_1})^{N_1} (\frac{\Delta - c_2}{d_2 - c_2})^{N_2} ... (\frac{\Delta - c_m}{d_m - c_m})^{N_m}, \end{aligned}$$

where $F_{J_i^*}(\Delta) = 1$, for $\Delta \ge d_i$. After exploiting Markov's Inequality,

$$\begin{aligned} E[Q] &\ge \Delta F_Q(Q \ge \Delta) \\ &= \Delta(1 - (\frac{\Delta - c_1}{d_1 - c_1})^{N_1} ... (\frac{\Delta - c_m}{d_m - c_m})^{N_m}) \\ &\ge \Delta(1 - (\frac{\Delta - c_b}{d_b - c_b})^{N_b}). \end{aligned}$$

We normalize the lower bound and upper bound with $c_b = 0$ and $d_b = 1$. Let Δ be the top-ρ percentile solution value, i.e. $\Delta = c_b + (1 - \rho)(d_b - c_b)$. Therefore,

$$E[\widetilde{Q}] \ge (1 - \rho)(1 - (1 - \rho)^{N_b}).$$

To find the maximum $(1 - \rho)(1 - (1 - \rho)^{N_b})$, we let

$$\frac{\partial (1 - \rho)(1 - (1 - \rho)^{N_b})}{\partial \rho} = 0.$$

The maximum $(1 - \rho)(1 - (1 - \rho)^{N_b})$ is acquired when ρ is $1 - (N_b + 1)^{-\frac{1}{N_b}}$. Therefore,

$$E[\widetilde{Q}] \ge N_b(\frac{1}{N_b + 1})^{\frac{N_b+1}{N_b}}.$$

Since $\widetilde{Q}$ is a lower bound of $\frac{Q}{Q^*}$,

$$E[Q] \ge N_b(\frac{1}{N_b + 1})^{\frac{N_b+1}{N_b}} \cdot Q^*.$$

If the computational budget allocation is $r-$stages with $T \ge mr\frac{\ln(m-1)}{\ln(\frac{1}{\alpha})}$, N_b is $\frac{T}{r}/m + \frac{1}{2}\frac{r-1}{2r}T$, which is $\frac{4+m(r-1)}{4rm}T$. □

3.3 Theoretical Result of CBAS with Gaussian Distribution

In the following, we derive the theoretical results for J_i following the normal distribution with mean μ_i and standard deviation of σ_i. The probability density function and cumulative distribution function are as follows.

$$p_{J_i}(x) = \phi(\frac{x - \mu_i}{\sigma_i}) = \frac{1}{\sigma_i\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{x-\mu_i}{\sigma_i})^2}$$

$$P_{J_i}(x) = \Phi(\frac{x - \mu_i}{\sigma_i}) = \frac{1}{2}(1 + erf(\frac{x - \mu_i}{\sigma_i\sqrt{2}}))$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2015.2468728, IEEE Transactions on Knowledge and Data Engineering

The distribution of maximal value J_i^*

$$p_{J_i^*}(x) = N_i P_{J_i}(x)^{N_i-1} p_{J_i}(x),$$

$$P_{J_i^*}(x) = P_{J_i}(x)^{N_i}.$$

Therefore, we derive the probability that J_b^* is smaller than J_i^* as follows.

$$\begin{aligned} &p(J_b^* - J_i^* \leq 0) = 1 - p(J_i^* \leq J_b^*) \\ &= 1 - \int_{-\infty}^{\infty} p_{J_b^*}(x) P_{J_i^*}(x) dx \\ &= 1 - \int_{-\infty}^{\infty} N_b P_{J_b}(x)^{N_b-1} p_{J_b}(x) P_{J_i}(x)^{N_i} dx \\ &= 1 - N_b \int_{-\infty}^{\infty} \Phi(\frac{x-\mu_b}{\sigma_b})^{N_b-1} \phi(\frac{x-\mu_b}{\sigma_b}) \Phi(\frac{x-\mu_i}{\sigma_i})^{N_i} dx \end{aligned}$$

As shown above, it is necessary that the probability is computed numerically because the $\Phi(x)$ function contains $erf(x)$ function which has no closed-form representation after being integrated. Although we can approximate the $\Phi(x)$ function with previous works [2], the $\Phi(x)$ function still becomes too complex after raising to the N_b-th or N_i-th power.

4 Neighbor Differentiation in Randomization

4.1 Greedy Neighbor Differentiation

In Section 3.1, *CBAS* includes two phases. The first phase initiates the start nodes, while the second phase allocates different computational budgets to each start node to generate different numbers of final solutions. During the growth of a partial solution, *CBAS* chooses a neighboring node uniformly at random at each iteration. In other words, each neighboring node of the partial solution is treated equally. It is expected that this homogeneous strategy needs more computational budgets, because a neighboring node inclined to generate a final solution with high willingness is not associated with a higher probability.

To remedy this issue, a simple algorithm *RGreedy* (randomized greedy) associates each neighboring node with a different probability according to its interest score and social tightness scores of the edges incident to the partial solution S_{t-1} obtained in the previous stage, which is similar to the concept in DGreedy. Given S_{t-1}, the ratio of the probabilities that *RGreedy* selects nodes v_i and v_j at iteration t is

$$\frac{P(v_i|S_{t-1})}{P(v_j|S_{t-1})} = \frac{W(\{v_i\} \cup S_{t-1})}{W(\{v_j\} \cup S_{t-1})},$$

where $W(\{v_i\} \cup S_{t-1})$ denotes the willingness of the node set $\{\{v_i\} \cup S_{t-1}\}$. At each iteration, *RGreedy* randomly selects a vertex in accordance with $W(\{v_j\} \cup S_{t-1})$, until k nodes are included.

Intuitively, *RGreedy* can be regarded as a *randomized version* of DGreedy with m start nodes, while DGreedy is a deterministic algorithm with only one start node. Thus, similar to DGreedy, the assignment of the probability limits the scope to only the local information associated with each node and incident edges. It is envisaged that *RGreedy* is difficult to generate a final solution with high willingness, which is also demonstrated in Section 5. In contrast, we propose *CBAS-ND* by exploiting the cross entropy method according to the sampling partial solutions in previous stages, in order to optimally assign a probability to each neighboring node.

4.2 Neighbor Differentiation with Cross Entropy

We enhance *CBAS* to *CBAS-ND* to differentiate the selection of a node neighboring each partial solution. Algorithm *CBAS* is divided into r stages. In each stage, it optimally adjusts the computational budgets allocated to each start node according to the sampled maximum and minimum willingness in previous stages. To effectively improve *CBAS*, *CBAS-ND* takes advantage of the cross entropy method [20] to achieve importance sampling by adaptively assigning a different probability to each neighboring node from the sampled results in previous stages. In contrast to *RGreedy* with a greedy-based probability vector assigned to the neighboring nodes, it is expected that *CBAS-ND* is able to obtain final solutions with better quality. Indeed, later in Section 4.3, we prove that the solution quality of *CBAS-ND* is better than *CBAS* with the same computational budget.

The flowchart of *CBAS-ND* is provided in Appendix E. We first define the node selection probability vector in *CBAS-ND*, which specifies the probability to add a node in G to the current partial solution expanded from a start node.

Definition 4: Let $\overrightarrow{p}_{i,t}$ denote the node selection probability vector for start node v_i in stage t.

$$\overrightarrow{p}_{i,t} = \langle p_{i,t,1}, ..., p_{i,t,j}, ..., p_{i,t,n} \rangle ,$$

where $p_{i,t,j}$ is the probability of selecting node v_j for start node v_i in the t-th stage.

In the first stage, the node selection probability vector $\overrightarrow{p}_{i,1}$ for each start node v_i is initialized homogeneously for every node, i.e. $\overrightarrow{p}_{i,1,j} = (k-1)/|V|$, $\forall v_j \in G$, $v_j \neq v_i$. That is, computational budgets $\frac{T_1}{m}$ are identically assigned to each start node, and the probability associated with every node is also the same. However, different from *CBAS* and *RGreedy*, *CBAS-ND* here examines the top-ρ samples for each start node v_i to generate $\overrightarrow{p}_{i,2}$, so that the node probability will be differentiated according to sampled result in stage 1.

Definition 5: A Bernoulli sample vector, denoted as $X_{i,q} = \langle x_{i,q,1}, ..., x_{i,q,j}, ..., x_{i,q,n} \rangle$, is defined to be the q-th sample vector from start node v_i, where $x_{i,q,j}$ is 1 if node v_j is selected in the q-th sample and 0 otherwise.

Definition 6: $\gamma_{i,t}$ is denoted as the top-ρ sample quantile of the performances in the t-th stage of start node v_i, i.e., $\gamma_{i,t} = W_{(\lceil \rho N_{i,t} \rceil)}$.

Specifically, after collecting $N_{i,1}$ samples $X_{i,1}, X_{i,2}, ..., X_{i,q}, ..., X_{i,N_{i,1}}$ generated from $\overrightarrow{p}_{i,1}$ for start node v_i, Node Selection Probability Update in the flowchart calculates the total willingness $W(X_{i,q})$ for each sample, and sorts them in the descending order, $W_{(1)} \geq ... \geq W_{(N_{i,1})}$, while $\gamma_{i,1}$ denotes the willingness of the top-ρ performance sample, i.e. $\gamma_{i,1} = W_{(\lceil \rho N_{i,1} \rceil)}$. With those sampled results,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2015.2468728, IEEE Transactions on Knowledge and Data Engineering

the selection probability $p_{i,2,j}$ of every node v_j in the second stage is derived according to the following equation,

$$p_{i,t+1,j} = \frac{\sum_{q=1}^{N_{i,t}} I_{\{W(X_{i,q})\geq\gamma_{i,t}\}} x_{i,q,j}}{\sum_{q=1}^{N_{i,r}} I_{\{W(X_{i,q})\geq\gamma_{i,t}\}}}, \qquad (12)$$

where the indicator function $I_{\{W(X_{i,q})\geq\gamma_{i,t}\}}$ is defined on the feasible solution space χ such that $I_{\{W(X_{i,q})\geq\gamma_{i,t}\}}$ is 1 if the willingness of sample $X_{i,q}$ exceeds a threshold $\gamma_{i,t} \in \mathbb{R}$, and 0 otherwise. Eq. (12) derives the node selection probability vector by fitting the distribution of top-ρ performance samples. Intuitively, if node v_j is included in most top-ρ performance samples in t-th stage, $p_{i,t+1,j}$ will approach 1 and be selected in $(t+1)$-th stage.

Later in Section 4.3, we prove that the above probability assignment scheme is optimal from the perspective of cross entropy. Eq. (12) minimizes the Kullback-Leibler cross entropy (KL) distance [20] between node selection probability $\overrightarrow{p}_{i,t}$ and the distribution of top-ρ performance samples, such that the performance of random samples in $t+1$ is guaranteed to be closest to the top-ρ performance samples in t. Therefore, by picking the top-ρ performance samples to generate the partial solutions in the next stage, the performance of random samples is expected to be improved after multiple stages. Most importantly, by minimizing the KL distance, the convergence rate is maximized.

Moreover, it is worth noting that a smoothing technique is necessary to be included in adjusting the selection probability vector,

$$\overrightarrow{p}_{i,t+1} = w\overrightarrow{p}_{i,t+1} + (1-w)\overrightarrow{p}_{i,t},$$

to avoid setting 0 or 1 in the selection probability for any node v_j, because v_j will no longer appear or always appear in this case. An example illustrating *CBAS* and *CBAS-ND* is provided in Appendix E. As demonstrated in Section 4.3, the solution quality of *CBAS-ND* is better than *CBAS* with the same computation budget.

4.3 Theoretical Result of CBAS-ND

In the following, we prove that the probability assignment with the cross-entropy method [20] in Eq. (12) is optimal. The idea of cross-entropy method originates from importance sampling[10], i.e., by changing the distribution of sampling on different neighbors such that the neighbors having the potential to boost the willingness are able to be identified and included. Therefore, we first derive the probability of a random sample according to the sampling results in previous stages. After this, we introduce importance sampling and derive the node selection probability vector in the WASO problem to replace the original sampling vector such that the Kullback-Leibler cross entropy (KL) distance between the sampling vector and the optimal importance sampling vector is minimized. Intuitively, a small KL distance ensures that two distributions are very close and implies that the node selection probability vector is optimal because the KL distance between the node selection probability vector in *CBAS-ND* and optimal node selection probability vector is minimized. Equipped with importance sampling vector, later in this section we prove that the solution quality of *CBAS-ND* is better than *CBAS*.

More specifically, let χ denote the feasible solution space, and X is a feasible solution in χ, i.e., $X \in \chi$. WASO chooses a group of attendee X^* to find the maximum willingness γ^*,

$$W(X^*) = \gamma^* = \max_{X\in\chi} W(X).$$

To derive the probability that the willingness of a random sample X exceeds a large value γ, i.e. $W(X) \geq \gamma$, it is necessary for *CBAS* to generate many samples given that it uniformly selects a neighboring node at random. In contrast, *CBAS-ND* leverages the notion of importance sampling to change the distribution of sampling on different neighbors. In the following, we first derive the optimal distribution of sampling. First, for the initial partial solution with one start node, let $f(X;\overrightarrow{p})$ denote the probability density function of generating a sample X according a real-valued vector $\overrightarrow{p}$, and $f(\cdot;\overrightarrow{p})$ is a family of probability density functions on χ, i.e.,

$$f(\cdot;\overrightarrow{p}) = \{f(X;\overrightarrow{p})|X \in \chi\}.$$

CBAS can be regarded as a special case of *CBAS-ND* with the homogeneous assignment on the above vector. A random sample $X(\overrightarrow{p})$ for $\overrightarrow{p} = \{p_1,...,p_j,...,p_n\}$ is generated with probability $f(X(\overrightarrow{p});\overrightarrow{p})$, where p_j denotes the probability of selecting node v_j and is the same for all j in *CBAS*. The probability $P_{\overrightarrow{p}}(\gamma)$ that the willingness of $X(\overrightarrow{p})$ exceeds the threshold γ is

$$\begin{aligned} P_{\overrightarrow{p}}(\gamma) &= \mathbb{P}_{\overrightarrow{p}}(W(X(\overrightarrow{p})) \geq \gamma) \\ &= \sum_{X\in\chi} I_{\{W(X(\overrightarrow{p}))\geq\gamma\}} f(X(\overrightarrow{p});\overrightarrow{p}). \end{aligned}$$

However, the above equation is impractical and inefficient for a large solution space, because it is necessary to scan the whole solution space χ and sum up the probability $f(X(\overrightarrow{p});\overrightarrow{p})$ of every sample X with $W(X(\overrightarrow{p})) \geq \gamma$. To more efficiently address this issue, a direct way to derive the estimator $\widehat{P}_{\overrightarrow{p}}(\gamma)$ of $P_{\overrightarrow{p}}(\gamma)$ is by employing a crude Monte-Carlo simulation and drawing N random samples $X_1(\overrightarrow{p})$,..., $X_N(\overrightarrow{p})$ by $f(\cdot,\overrightarrow{p})$ to find $P_{\overrightarrow{p}}(\gamma)$,

$$\widehat{P}_{\overrightarrow{p}}(\gamma) = \frac{1}{N}\sum_{i=1}^{N} I_{\{W(X_i(\overrightarrow{p}))\geq\gamma\}}.$$

However, the crude Monte-Carlo simulation poses a serious problem when $\{W(X(\overrightarrow{p})) \geq \gamma\}$ is a rare event since rare events are difficult to be sampled, and thus a large sample number N is necessary to estimate $P_{\overrightarrow{p}}(\gamma)$ correctly.

Based on the above observations, *CBAS-ND* attempts to find the distribution $f(X(\overrightarrow{p});\overrightarrow{p})$ based on another importance sampling pdf $f(X(\overrightarrow{p_g});\overrightarrow{p_g})$ to reduce the required sample number. For instance, consider a network with 3 nodes, i.e. $V = \{v_1, v_2, v_3\}$, and the 2-node group where

10. Importance sampling [20] is used to estimate the properties of a target distribution by using the observations from a different distribution. By changing the distribution, the "important" values can be effectively extracted and emphasized by sampling more frequently to reduce the sample variance.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2015.2468728, IEEE Transactions on Knowledge and Data Engineering

the maximum willingness γ^* is $\{v_1, v_2\}$. The expected number of samples with node selection vector $\{\frac{2}{3}, \frac{2}{3}, \frac{2}{3}\}$ in *CBAS* is larger than the node selection vector of $\{1, 1, 0\}$ in *CBAS-ND*. In finer detail, let $X_i(\overrightarrow{p_g})$ denote the i-th random sample generated by $f(X(\overrightarrow{p_g}); \overrightarrow{p_g})$. *CBAS-ND* first creates random samples $X_1(\overrightarrow{p_g})$,..., $X_N(\overrightarrow{p_g})$ generated by $\overrightarrow{p_g}$ on χ and then estimates $\widehat{P}_{\overrightarrow{p}}(\gamma)$ according to the likelihood ratio (LR) estimator $\frac{f(X_i(\overrightarrow{p_g}); \overrightarrow{p})}{f(X_i(\overrightarrow{p_g}); \overrightarrow{p_g})}$,

$$\widehat{P}_{\overrightarrow{p}}(\gamma) = \frac{1}{N}\sum_{i=1}^{N} I_{\{W(X_i(\overrightarrow{p}))\geq\gamma\}} = \frac{1}{N}\sum_{i=1}^{N}\{I_{\{W(X_i(\overrightarrow{p_g}))\geq\gamma\}}\frac{f(X_i(\overrightarrow{p_g}); \overrightarrow{p})}{f(X_i(\overrightarrow{p_g}); \overrightarrow{p_g})}. \quad (13)$$

Notice that the above equation holds when N is infinity, but in most cases N only needs to be sufficiently large in practical implementation [6]. Now the question becomes how to derive $\overrightarrow{p_g}$ for importance sampling pdf $f(X(\overrightarrow{p_g}); \overrightarrow{p_g})$ to reduce the number of samples. The optimal importance sampling pdf $f^*(X_i(\overrightarrow{p_g}); \overrightarrow{p_g})$ to correctly estimate $P_{\overrightarrow{p}}(\gamma)$ thus becomes

$$f^*(X_i(\overrightarrow{p_g}); \overrightarrow{p_g}) = \frac{I_{\{W(X_i(\overrightarrow{p_g}))\geq\gamma\}} f(X_i(\overrightarrow{p_g}); \overrightarrow{p})}{P_{\overrightarrow{p}}(\gamma)}. \quad (14)$$

In other words, by substituting $f(X_i(\overrightarrow{p_g}); \overrightarrow{p_g})$ with $f^*(X_i(\overrightarrow{p_g}); \overrightarrow{p_g})$ in Eq. (13), $\widehat{P}_{\overrightarrow{p}}(\gamma) = \frac{1}{N}\sum_{i=1}^{N} P_{\overrightarrow{p}}(\gamma)$ holds, implying that only 1 sample is required to estimate the correct $P_{\overrightarrow{p}}(\gamma)$, i.e., $N = 1$. However, it is difficult to find the optimal $f^*(X(\overrightarrow{p_g}); \overrightarrow{p_g})$ since it depends on $P_{\overrightarrow{p}}(\gamma)$, which is unknown *a priori* and is therefore not practical for WASO.

Based on the above observations, *CBAS-ND* optimally finds $\overrightarrow{p_g}$ and the importance sampling pdf $f(X(\overrightarrow{p_g}); \overrightarrow{p_g})$ to minimize the Kullback-Leibler cross entropy (KL) distance between $f(X(\overrightarrow{p_g}); \overrightarrow{p_g})$ and optimal importance sampling pdf $f^*(X(\overrightarrow{p_g}); \overrightarrow{p_g})$, where the KL distance measures two densities f^* and f as

$$D(f^*, f) = \sum_{X\in\chi} f^*(X)\ln f^*(X) - \sum_{X\in\chi} f^*(X)\ln f(X). \quad (15)$$

The first term in the above equation is related to f^* and is fixed, and minimizing $D(f^*, f)$ is equivalent to maximizing the second term, i.e., $\sum_{X\in\chi} f^*(X)\ln f(X)$. It is worth noting that the importance sampling pdf $f(X(\overrightarrow{p_g}); \overrightarrow{p_g})$ is referenced to a vector $\overrightarrow{p_g}$. Thus, after substituting $f^*(X_i(\overrightarrow{p_g}); \overrightarrow{p_g})$ in Eq. (14) into the Eq. (15), the reference vector $\overrightarrow{p_g}$ of importance sampling pdf $f(X(\overrightarrow{p_g}); \overrightarrow{p_g})$ that maximizes the second term of Eq. (15) is the optimal reference vector $\overrightarrow{p_g}^*$ with the minimum KL distance, $\overrightarrow{p_g}^*$ is derived as follows:

$$\arg\max_{\overrightarrow{p_g}} \sum_{X\in\chi} \frac{I_{\{W(X(\overrightarrow{p_g}))\geq\gamma\}} f(X(\overrightarrow{p_g}); \overrightarrow{p})}{P_{\overrightarrow{p}}(\gamma)} \ln f(X(\overrightarrow{p_g}); \overrightarrow{p_g}). \quad (16)$$

Since $P_{\overrightarrow{p}}(\gamma)$ is not related to $\overrightarrow{p_g}$. Eq. (16) is equivalent to

$$\arg\max_{\overrightarrow{p_g}} \mathbb{E}_{\overrightarrow{p_g}} I_{\{W(X(\overrightarrow{p_g}))\geq\gamma\}} \ln f(X(\overrightarrow{p_g}); \overrightarrow{p_g}),$$

Because it is computationally intensive to generate and compare every feasible $\overrightarrow{p_g}$, we estimate $\mathbb{E}_{\overrightarrow{p_g}} I_{\{W(X(\overrightarrow{p_g}))\geq\gamma\}}$ by drawing N samples as

$$\arg\max_{\overrightarrow{p_g}} \frac{1}{N}\sum_{i=1}^{N} I_{\{W(X_i(\overrightarrow{p_g}))\geq\gamma\}} \ln f(X_i(\overrightarrow{p_g}); \overrightarrow{p_g}).$$

Specifically, *CBAS-ND* first generates random samples X_1,...,X_i,...,X_N, where X_i is the i-th sample and is a Bernoulli vector generated by a node selection probability vector $\overrightarrow{p_g}$, i.e., $X_i = (x_{i,1}, ..., x_{i,n}) \sim Ber(\overrightarrow{p_g})$, where $\overrightarrow{p_g} = \{p_1,...,p_j,...,p_n\}$ and p_j denotes the probability of selecting node v_j. Consequently, the pdf $f(X_i(\overrightarrow{p_g}); \overrightarrow{p_g})$ is

$$f(X_i(\overrightarrow{p_g}); \overrightarrow{p_g}) = \prod_{j=1}^{N} p_j^{x_{i,j}}(1-p_j)^{1-x_{i,j}}.$$

To find the optimal reference vector $\overrightarrow{p}^*$ with Eq. (16), we first calculate the first derivative w.r.t. p_j,

$$\frac{\partial}{\partial p_j}\ln f(X_i(\overrightarrow{p_g}); \overrightarrow{p_g}) = \frac{\partial}{\partial p_j}\ln p_j^{x_{i,j}}(1-p_j)^{1-x_{i,j}}. \quad (17)$$

Since $x_{i,j}$ can be either 0 or 1, Eq. (17) is simplified to

$$\frac{\partial}{\partial p_j}\ln f(X_i(\overrightarrow{p_g}); \overrightarrow{p_g}) = \frac{1}{(1-p_j)p_j}(x_{i,j} - p_j).$$

The optimal reference vector $\overrightarrow{p}^*$ is obtained by setting the first derivative of Eq. (16) to zero.

$$\frac{\partial}{\partial p_j}\sum_{i=1}^{N} I_{\{W(X_{i,j})\geq\gamma\}} \ln f(X_i(\overrightarrow{p_g}); \overrightarrow{p_g}) = \frac{1}{(1-p_j)p_j}\sum_{i=1}^{N} I_{\{W(X_i)\geq\gamma\}}(x_{i,j} - p_j) = 0.$$

Finally, the optimal p_j assigned to each node v_j is

$$p_j = \frac{\sum_{i=1}^{N} I_{\{W(X_i)\geq\gamma\}} x_{i,j}}{\sum_{i=1}^{N} I_{\{W(X_i)\geq\gamma\}}}.$$

Theorem 5: The solution quality of CBAS-ND is better than CBAS under the same computation budget T.

Proof: Let $\overrightarrow{v}_t$ be the node selection vector in the t-th stage, where $\overrightarrow{v}_t = \{\overrightarrow{v}_{t,1}, \overrightarrow{v}_{t,2}, ... \overrightarrow{v}_{t,n}\}$. We first define the random variables $\phi_v^{t,i} = v_{t,i} I_{\{x_i^*=1\}} + (1 - v_{t,i}) I_{\{x_i^*=0\}}$ for all $i = 1, ..., n$, where $I_{\{x_i^*=1\}}$ is the indicator function with 1 if node v_i is in the optimal solution, and 0 otherwise. Then, let ϕ_v^t denote the probability to generate the optimal solution in the t-th stage,

$$\phi_v^t = f(X^*; \overrightarrow{p}_v) = \prod_{i=1}^{n} \phi_v^{t,i}.$$

Let E_r be the event that does not sample the optimal solution in the final r-th stage. From the previous work [6], the probability for the willingness to converge to the optimal solution can be formulated as

$$1 - P(E_r) \geq 1 - P(E_1)\exp(-\frac{N_i}{r}\phi_u^1 \sum_{t=1}^{r-1} w^{tn}), \quad (18)$$

where w in Eq. (18) is the smoothing technique parameter. Therefore, since *CBAS* is identical to *CBAS-ND* with $w =$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2015.2468728, IEEE Transactions on Knowledge and Data Engineering

0, the convergence rate that *CBAS-ND* samples the optimal solution is larger than *CBAS*. Therefore, to achieve the same solution quality, *CBAS-ND* requires less computation budget than *CBAS*. When *CBAS* runs out of computation budget, i.e., T, the computation budget that *CBAS-ND* achieves the same quality is less than T. Let r_{ND} denote the number of stage that *CBAS-ND* achieves the same quality. Since $r_{ND} \leq r$, we have

$$1 - P(E_1)\exp(-\frac{N_i}{r_{ND}}\phi_u^1 \sum_{t=1}^{r_{ND}-1} w^{tn})$$

$$\leq 1 - P(E_1)\exp(-\frac{N_i}{r}\phi_u^1 \sum_{t=1}^{r-1} w^{tn})$$

Therefore, the solution quality of *CBAS-ND* is better than that of *CBAS*. The theorem follows. □

5 Experimental Results

5.1 Experiment Setup

We implement *CBAS-ND* in Facebook and invite 137 people from various communities, e.g., schools, government, technology companies, and businesses to join our user study, to compare the solution quality and the time to answer WASO with manual coordination and *CBAS-ND* for demonstrating the need of an automatic group recommendation service. Each user has been informed of the solution quality defined in Eq. (1) and is asked to compute the WASO problem given the node and edge weights, and thus users are not asked to label the data for providing the ground truth, because it is difficult for them to carefully examine the social and interest dimensions simultaneously for forming an effective group. Moreover, each user is asked to plan 10 social activities with the social graphs extracted from their social networks in Facebook. The interest scores follow the power-law distribution according to the recent analysis [5] on real datasets, which has found the power exponent $\beta = 2.5$. The social tightness score between two friends is derived according to the widely adopted model based on the number of common friends that represent the proximity interaction [3]. Then, social tightness scores and interest scores are normalized. Nevertheless, after the scores are returned by the above renowned models, each user is still allowed to fine-tune the two scores by themselves. The 10 problems explore various network sizes and different numbers of attendees in two different scenarios. In the first 5 problems, the user needs to participate the group activity and is inclined to choose her close friends, while the following 5 problems allow the user to choose an arbitrary group of people with high willingness. In other words, *CBAS-ND* in the first 5 problems always chooses the user as a start node. The execution time of manual coordination is the time interval between a users click the start button in our user study program and presses the finish button to submit the result. After the start button is clicked, the problem instance with the social network topology, node weights, and edge weights is presented to the user.

In addition to the user study, three real datasets are tested in the experiment. The first dataset is crawled from Facebook with $90,269$ users in the New Orleans network[11]. The second dataset is crawled from DBLP dataset with $511,163$ nodes and $1,871,070$ edges. The third dataset, Flickr[12], with $1,846,198$ nodes and $22,613,981$ edges, is also incorporated to demonstrate the scalability of the proposed algorithms. Due to the space constraint, detailed experimental results of the Facebook datasets are presented in Appendix G.

In the following, we compare the deterministic greedy algorithm (DGreedy), randomized greedy algorithm (RGreedy), *CBAS*, *CBAS-ND*, and *IP (Integer Programming)* solved by IBM CPLEX in an HP DL580 server with four Intel E7-4870 2.4 GHz CPUs and 128 GB RAM. IBM CPLEX is regarded as the fastest general-purpose parallel optimizer, and we adopt it to solve the Integer Programming formulation for finding the optimal solution to WASO[13]. It is worth noting that even though *RGreedy* performs much better than its counterpart *DGreedy* and is closer to *CBAS* and *CBAS-ND*, it is computation intensive and not scalable to support a large group size. Therefore, we can only plot a few results of *RGreedy* in some figures. The default m is set to be n/k since n/k different k-person groups can be partitioned from a network with n. With m equal n/k, the start nodes averagely cover the whole network. Nevertheless, the experimental analysis manifests that m can be set to be smaller than n/k in WASO since the way we select start nodes efficiently prunes the start nodes which do not generate good solutions. The computational budget of *CBAS-ND* is not wasted much since the start node that do not generate good solutions will be pruned after the first stage. The default cross-entropy parameters ρ and w are 0.3 and 0.9 respectively, and α is 0.99 as recommended by the cross-entropy method [20]. The results with different settings of parameters will be presented. Since *CBAS* and *CBAS-ND* natively support parallelization, we also implemented them with OpenMP for parallelization, to demonstrate the gain in parallelization with more CPU cores.

5.2 User Study

The weights λ and (1-λ) in Section 2 for interest scores and social tightness scores are directly specified by the users according to their preferences, and Figure 2(a) shows that the range of the weight mostly spans from 0.37 to 0.66 with the average as 0.503, indicating that both social tightness and interest are crucial factors in activity planning. Figures 2(b)-(e) compare manual coordination and *CBAS-ND* for the WASO problem with an initiator in the solution in the user study. It is worth noting that we generate the ground truth of user study with *IP* solved by IBM CPLEX to evaluate the solution quality. Each user knows the solution quality defined in this paper and is asked to compute the WASO problem given the node and edge weights.

11. http://socialnetworks.mpi-sws.org/data-wosn2009.html.
12. http://socialnetworks.mpi-sws.org/data-imc2007.html.
13. Note that because WASO is NP-Hard, it is only possible to find the optimal solutions to WASO with IBM CPLEX in small cases.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2015.2468728, IEEE Transactions on Knowledge and Data Engineering

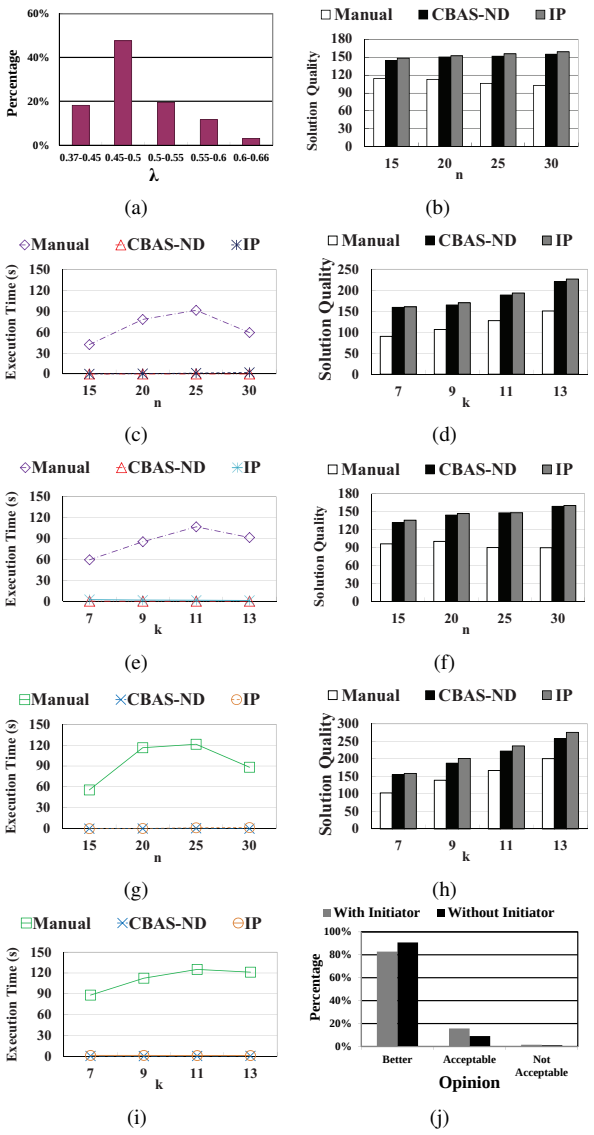

Fig. 2. Results of user study

Figures 2(b) and (c) present the solution quality and running time with different network sizes, where the expected number of attendees k is 7. The result indicates that the solutions obtained by *CBAS-ND* is very close to the optimal solutions acquired from solving *IP* with IBM CPLEX. WASO is challenging for manual coordination, even when the network contains only dozens of nodes. It is interesting that $n = 30$ is too difficult for manual coordination because some users start to give up thus require smaller time for finding a solution. Figures 2(d) and (e) present the results with different k where the network size is 30. The results show that the solution quality obtained by manual coordination with $k = 7$ is only 66% of *CBAS-ND*, since it is challenging for a person to jointly maximize the social tightness and interest. Similarly, we discover that some users start to give up when $k = 13$, and the processing time of manual selection grows when the user is not going to join the group activity.

Figures 2(f)-(i) compare manual coordination and *CBAS-ND* for the WASO problem without an initiator in the solution in the user study. The trend is similar with the WASO problem with an initiator in the solution. However, WASO problem without an initiator is more difficult and more time-consuming because users can arbitrarily choose a group with high willingness and thus considers many more candidate groups. Finally, we return the solutions obtained by *CBAS-ND* to the users, and Figure 2(j) manifests that 98.5% of users think the solutions are better or acceptable, as compared to the solutions found by themselves. Therefore, it is desirable to deploy *CBAS-ND* as an automatic group recommendation service, especially to address the need of a large group in a massive social network nowadays.

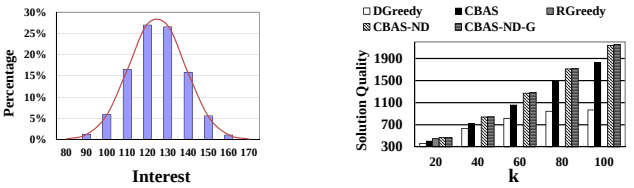

Fig. 3. Experimental results of WASO with Gaussian distribution

5.3 Performance Comparison and Sensitivity Analysis

5.3.1 Facebook

Figure 3(a) shows the interest histogram of random samples on Facebook, which indicates that the distribution follows a Gaussian distribution with the mean as 124.71 and standard deviation as 13.83. Notice that the allocation ratio for the variant *CBAS-ND-G* of *CBAS-ND* by replacing the uniform distribution with the Gaussian distribution in Theorem 2 is derived in Section 3.3. Figure 3(b) indicates that the solution quality of *CBAS-ND* and *CBAS-ND-G* is very close. In contrast to *CBAS-ND-G*, however, *CBAS-ND* is more efficient and easier to be implemented because it does not involve the probability integration to find the probability of the best start node.

5.3.2 DBLP

CBAS and *CBAS-ND* is also evaluated on the DBLP dataset. Figures 4(a) and (b) compare the solution quality and running time. The results show that *CBAS-ND* outperforms *DGreedy* by 92% and *RGreedy* by 32% in solution quality. Both *CBAS* and *CBAS-ND* are still faster than *RGreedy* by an order of 10^{-2}. However, *RGreedy* runs faster on the DBLP dataset than on the Facebook dataset, because the DBLP dataset is a sparser graph with an average node degree of 3.66. Therefore, the number of candidate nodes for each start node in the DBLP dataset increases much more slowly than in the Facebook dataset with an average node degree of 26.1. Nevertheless, *RGreedy* is still not able to generate a solution for a large group size k due to its unacceptable efficiency.

Figures 4(c) and (d) present the solution quality and running time of *RGreedy*, *CBAS,* and *CBAS-ND* with different numbers of start nodes, i.e., m. The solution quality of

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2015.2468728, IEEE Transactions on Knowledge and Data Engineering

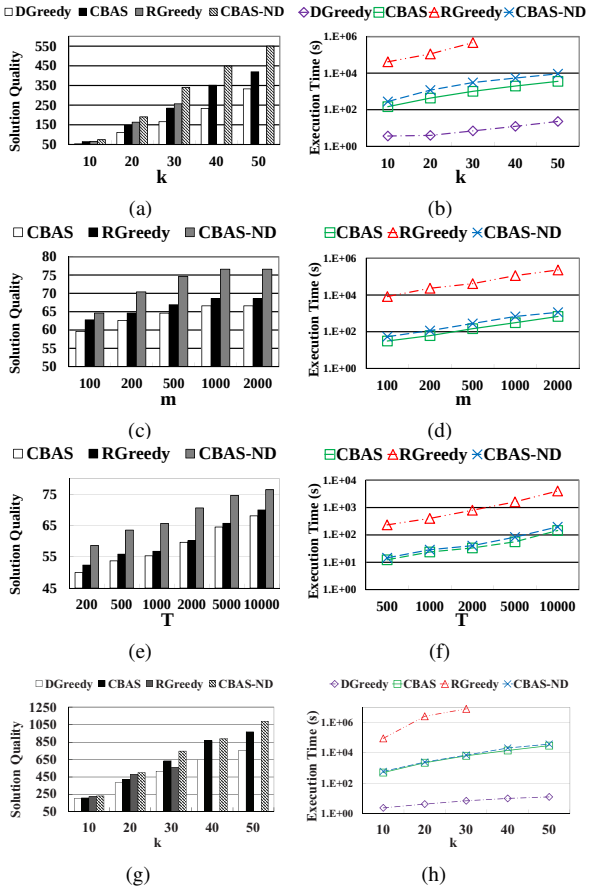

Fig. 4. Experimental results on DBLP and Flickr datasets

CBAS-ND converges when m is 1000, indicating that here it is sufficient to assign m as a number much smaller than $\frac{n}{k} = \frac{511163}{10} \approx 51116$, because the way we select start node efficiently filter out the start nodes that do not generate good solutions. Compared to $m = 500$ in Facebook dataset, *CBAS* and *CBAS-ND* need a larger m as 1000 due to a larger network size in DBLP dataset. Figures 4(e) and (f) compare the solution quality and running time with different T. As T increases, the solution quality of *CBAS-ND* also grows faster than the other approaches. Both *CBAS* and *CBAS-ND* outperform *RGreedy* by an order of 10^{-1}.

5.3.3 Flickr

Finally, to evaluate the scalability of *CBAS* and *CBAS-ND*, Figures 4(g) and (h) compare the solution quality and running time on Flickr dataset. The results show that *CBAS-ND* outperforms *DGreedy* by 31% in solution quality when $k = 50$. *CBAS* and *CBAS-ND* are both faster than *RGreedy* in an order of 10^{-2}. The trend of running time on Flickr dataset is similar to Facebook dataset, instead of DBLP dataset, because the average node degrees of the Flickr dataset and Facebook dataset are similar. Moreover, *RGreedy* can support only $k = 30$ in the Flickr and DBLP dataset, manifesting that it is not practical to deploy *RGreedy* in a real massive social network.

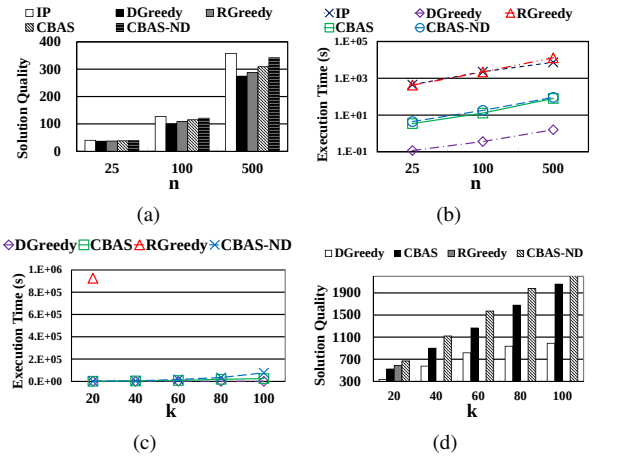

Fig. 5. Experimental results on Integer Programming and WASO-dis

5.3.4 Integer Programming and WASO-dis

To evaluate the solution quality of *CBAS-ND*, Figures 5(a) and (b) compare the solution quality and running time of *IP* (ground truth) with $k = 10$. Since WASO is NP-hard, i.e., the running time for obtaining the ground truth is unacceptably large, we extract 1000 small real datasets from the DBLP dataset with the node sizes as 25, 100, and 500 respectively. The result shows that the solution quality of *CBAS-ND* is very close to *IP*, while the running time is smaller by an order of 10^{-2}. It is worth noting that *CBAS-ND* here is single-threaded, but IP is solved by IBM CPLEX (parallel version).

For separate groups, Figure 5(c) first presents the running time with different group sizes, i.e., k, where $m = \frac{n}{k}$, $\rho = 0.3$, and $w = 0.9$, respectively. For all algorithms, the virtual node $\overline{v}$ is added to the selection set V_S to relax the connectivity constraint. *RGreedy* computes the incremental willingness of every node in V_A to the selection set V_S, where V_A includes all nodes, and thus are computationally intractable. Therefore, *RGreedy* is unable to return a solution within 24 hours when the group size is larger than 20. Figure 5(d) presents the solution quality with different activity sizes. The results indicate that *CBAS-ND* outperforms *DGreedy*, *RGreedy*, and *CBAS*, especially under a large k. In addition, compared to the experimental results in WASO, the difference between *CBAS-ND* and *DGreedy* becomes more significant as k increases. The reason is that the greedy algorithm selects the node with the largest incremental willingness to the current group and thus is inclined to select a connected group, where the optimal solution may be disconnected.

6 Conclusion

To the best of our knowledge, there is no real system or existing work in the literature that addresses the issues of automatic activity planning based on topic interest and social tightness. To fill this research gap and satisfy an important practical need, this paper formulated a new optimization problem called WASO to derive a set of attendees and maximize the willingness. We proved that WASO is NP-hard and devised two simple but effective

randomized algorithms, namely *CBAS* and *CBAS-ND*, with an approximation ratio. The user study demonstrated that the social groups obtained through the proposed algorithm implemented in Facebook significantly outperforms the manually configured solutions by users. This research result thus holds much promise to be profitably adopted in social networking websites as a value-added service.

The user study resulted in practical directions to enrich WASO for future research. Some users suggested that we integrate the proposed willingness optimization system with automatic available time extraction to filter unavailable users, such as by integrating the proposed system with Google Calendar. Since candidate attendees are associated with multiple attributes in Facebook, e.g., location and gender, these attributes can be specified as input parameters to further filter out unsuitable candidate attendees. Last but not the least, some users pointed out that our work could be extended to allow users to specify some attendees that must be included in a certain group activity.

REFERENCES

[1] U. Brandes, D. Delling, M. Gaertler, R. Goerke, M. Hoefer, Z. Nikoloski, and D. Wagner. On modularity clustering. *IEEE Transactions on Knowledge and Data Engineering*, 20:172–188, 2008.

[2] W. Bryc. A uniform approximation to the right normal tail integral. *In Proc. Appl. Math. Comput.*, 2002.

[3] V. Chaoji, S. Ranu, R. Rastogi, and R. Bhatt. Recommendations to boost content spread in social networks. In *WWW*, pages 529–538, 2012.

[4] C. H. Chen, E. Yucesan, L. Dai, and H. C. Chen. Efficient computation of optimal budget allocation for discrete event simulation experiment. *IIE Transactions*, 42(1):60–70, 2010.

[5] A. Clauset, C. R. Shalizi, and M. E. J. Newman. Power-law distributions in empirical data. In *SIAM*, 51(4):661–703, 2009.

[6] A. Costa, J. Owen, and D. P. Kroese. Convergence properties of the cross-entropy method for discrete optimization. *Operations Research Letters*, 35(5):573–580, 2007.

[7] L. Dai, C. H. Chen, and J. R. Birge. Large convergence properties of two-stage stochastic programming. *J. Optimization Theory and Applications*, 106(3):489–510, 2000.

[8] M. Deutsch and H. B. Gerard. A study of normative and informational social influences upon individual judgment. *J. Abnormal and Social Psychology*, 51(3):291–301, 1955.

[9] U. Feige, D. Peleg, and G. Kortsarz. The dense k-subgraph problem. *Algorithmica*, 29(3):410–421, 2001.

[10] A. Gajewar and A. D. Sarma. Multi-skill collaborative teams based on densest subgraphs. In *SDM*, pages 165–176, 2012.

[11] E. Gilbert and K. Karahalios. Predicting tie strength with social media. In *CHI*, pages 211–220, 2009.

[12] D. F. Gleich and C. Seshadhri. Vertex neighborhoods, low conductance cuts, and good seeds for local community methods. In *KDD*, pages 597–605, 2012.

[13] M. T. Hajiaghayi and K. Jain. The prize-collecting generalized steiner tree problem via a new approach of primal-dual schema. *In Proc. SODA*, 2006.

[14] M. F. Kaplan and C. E. Miller. Group decision making and normative versus informational influence: Effects of type of issue and assigned decision rule. *Journal of Personality and Social Psychology*, 53(2):306–313, 1987.

[15] M. Kargar and A. An. Discovering top-k teams of experts with/without a leader in social networks. In *CIKM*, pages 985–994, 2011.

[16] A. Krause and D. Golovin. Submodular function maximization. In *Tractability: Practical Approaches to Hard Problems (to appear)*. Cambridge University Press, 2014.

[17] C. Li and M. Shan. Team formation for generalized tasks in expertise social networks. In *SocialCom*, pages 9–16, 2010.

[18] A. Mislove, B. Viswanath, K. P. Gummadi, and P. Druschel. You are who you know: Inferring user profiles in online social networks. In *WSDM*, pages 251–260, 2010.

[19] M. Mitzenmacher and E. Upfal. Probability and computing: Randomized algorithms and probabilistic analysis. Cambridge University Press, 2005.

[20] R. Y. Rubinstein. Combinatorial optimization, cross-entropy, ants and rare events. In S. Uryasev and P. M. Pardalos, editors, *Stochastic Optimization: Algorithms and Applications*, pages 304–358. Kluwer Academic, 2001.

[21] M. Sozio and A. Gionis. The community-search problem and how to plan a successful cocktail party. In *KDD*, pages 939–948, 2010.

[22] C. Wilson, B. Boe, A. Sala, K. P. N. Puttaswamy, and B. Y. Zhao. User interactions in social networks and their implications. In *Eurosys*, pages 205–218, 2009.

[23] D. N. Yang, W. C. Lee, N. H. Chia, M. Ye, and H. J. Hung. Bundle configuration for spread maximization in viral marketing via social networks. In *CIKM*, pages 2234–2238, 2012.

[24] M. Ye, X. Liu, and W. C. Lee. Exploring social influence for recommendation - a probabilistic generative model approach. In *SIGIR*, pages 671–680, 2012.

Hong-Han Shuai received the B.S. degree from the Department of Electrical Engineering, National Taiwan University (NTU), Taipei, Taiwan, R.O.C., in 2007, and the M.S. degree in computer science from NTU in 2009. He is currently an Ph.D. student of the Graduate Institute of Communication Engineering, NTU, and also a research assistant of Research Center for Information Technology Innovation, Academia Sinica. His research interests are in the area of multimedia networking, image/video processing, and data mining.

De-Nian Yang received the BS and PhD degrees from the Department of Electrical Engineering at National Taiwan University in 1999 and 2004, respectively. He is now an associate research fellow in the Institute of Information Science, Academia Sinica, Taiwan. His research interests include mobile social networks and mobile multimedia networking. He received K.-T. Li Distinguished Young Scholar Award in ACM Taipei/Taiwan Chapter, Research Exploration Award from Pan Wen Yuan Foundation, and Project for Excellent Junior Research Investigators in NSC, Taiwan. He is a senior member of IEEE and a member of ACM.

Philip S. Yu received the B.S. Degree in E.E. from National Taiwan University, the M.S. and Ph.D. degrees in E.E. from Stanford University, and the M.B.A. degree from New York University. He is a Distinguished Professor in Computer Science at the University of Illinois at Chicago and also holds the Wexler Chair in Information Technology. Before joining UIC, Dr. Yu was with IBM, where he was manager of the Software Tools and Techniques group at the Watson Research Center. His research interest is on big data, including data mining, data stream, database and privacy. He has published more than 830 papers in refereed journals and conferences. He holds or has applied for more than 300 US patents. Dr. Yu is a Fellow of the ACM and the IEEE. He is the Editor-in-Chief of ACM Transactions on Knowledge Discovery from Data. He received the IEEE Computer Society 2013 Technical Achievement Award for pioneering and fundamentally innovative contributions to the scalable indexing, querying, searching, mining and anonymization of big data, the ICDM 2013 10-year Highest-Impact Paper Award, the EDBT Test of Time Award (2014), and the Research Contributions Award from IEEE Intl. Conference on Data Mining (2003).

Ming-Syan Chen received the BS degree in Electrical Engineering from National Taiwan University, Taipei, Taiwan, and the MS and PhD degrees in Computer, Information and Control Engineering from the University of Michigan, Ann Arbor, MI, USA, in 1985 and 1988, respectively. He is now a Distinguished Research Fellow and the Director of Research Center of Information Technology Innovation (CITI) in the Academia Sinica, Taiwan, and is also a Distinguished Professor jointly appointed by EE Department, CSIE Department, and Graduate Institute of Communication Eng. (GICE) at National Taiwan University. His research interests include databases, data mining, and cloud computing. He is a Fellow of ACM and a Fellow of IEEE.