Expertise Matching via Constraint-based Optimization

Wenbin Tang, Jie Tang, and Chenhao Tan
Department of Computer Science and Technology, Tsinghua University, China
{tangwb06,tch06}@mails.tsinghua.edu.cn, jietang@tsinghua.edu.cn

Abstract—**Expertise matching, aiming to find the alignment between experts and queries, is a common problem in many real applications such as conference paper-reviewer assignment, product-reviewer alignment, and product-endorser matching. Most of existing methods for this problem usually find "relevant" experts for each query independently by using, e.g., an information retrieval method. However, in real-world systems, various domain-specific constraints must be considered. For example, to review a paper, it is desirable that there is at least one senior reviewer to guide the reviewing process. An important question is: "Can we design a framework to *efficiently* find the *optimal solution* for expertise matching under various constraints?" This paper explores such an approach by formulating the expertise matching problem in a constraint-based optimization framework. Interestingly, the problem can be linked to a convex cost flow problem, which guarantees an optimal solution under given constraints. We also present an online matching algorithm to support incorporating user feedbacks in real time. The proposed approach has been evaluated on two different genres of expertise matching problems. Experimental results validate the effectiveness of the proposed approach.**

Keywords-**Expertise matching; Constrained optimization; Paper-reviewer assignment**

I. Introduction

The fusion of computer technology and human collective intelligence has recently emerged as a popular way for users to find and share information on the internet. For example, ChaCha.com, one of the largest mobile search engines, has already attracted users to answer over 300 million questions; Epinions.com has collected millions of reviews for various products. The human-based computation offers a new direction in search with its unique use of human intelligence; however, it also poses some brand new challenges. One key problem, referred to as expertise matching, is how to align human experts with questions (queries)? Straightforward, we hope that the human experts who are assigned to answer a question have the specific expertise related to the question. But it is obviously insufficient. An ideal matching system should also consider various constraints in the real world, for example, an expert can only answer a certain number of questions (load balance); as the authoritative degree of different experts may vary largely, it is desirable that each question can be answered/reviewed by at least one senior expert (authority balance); a question may be relevant to multiple different aspects (topics), thus it is expected that the combined expertise of all assigned experts could cover all aspects of questions (topic coverage).

The problem has attracted considerable interest from different domains. For example, several works have been made for conference paper-reviewer assignment by using methods such as mining the web [10], latent semantic indexing [6], probabilistic topic modeling [14][16], integer linear programming [13], minimum cost flow [9] and hybrid approach of domain knowledge and matching model[18]. A few systems [11][5][15] have also developed to help proposal-reviewer and paper-reviewer assignment. However, most existing methods mainly focus on the matching algorithm, i.e., how to accurately find (or rank) the experts for each query, but ignore the different constraints or tackle the constraints with heuristics, which obviously results in an approximate (or even inaccurate) solution. Moreover, these methods usually do not consider user feedbacks. On the other hand, there are some methods focusing on expert finding. For example, Fang et al. [7] proposed a probabilistic model for expert finding, and Petkova et al. [17] employed a hierarchical language model in enterprise corpora. Balog et al. [2] employ probabilistic models to study the problem of expert finding, which tries to identify a list of experts for a query. However, these methods retrieve experts for each query independently, and cannot be directly adapted to deal with the expertise matching problem. Thus, several key questions arise for expertise matching, i.e., how to design a framework for expertise matching to guarantee an optimal solution under various constraints? how to develop an online algorithm so that it can incorporate user feedbacks in real time?

Problem Formulation We first formulate our problem precisely. Given a set of experts $V = \{v_i\}$, each expert has different expertise over all topics. Formally, we assume that there are in total T aspects of expertise (called topics) and one's expertise degree on topic $z \in \{1 \cdots T\}$ is represented as a probability $\theta_{v_i z}$ with $\sum_z \theta_{v_i z} = 1$. Further, given a set of queries $Q = \{q_j\}$, each query is also related to multiple topics, also represented as a T-dimensional topic distribution $\sum_z \theta_{q_j z} = 1$, where $\theta_{q_j z}$ is the probability of query q_j on topic z. Notations are summarized in Table I.

Given this, our objective is to assign m experts to each query by satisfying certain constraints. For a concrete example, an university department has five teaching staffs and

978-0-7695-4191-4/10 $26.00 © 2010 IEEE
DOI 10.1109/WI-IAT.2010.133

Table I
NOTATIONS.

SYMBOL	DESCRIPTION
M	number of experts
N	number of queries
T	number of topics
V	the set of candidate experts
Q	the set of queries
v_i	one expert
q_j	one query
$\theta_{v_i z}$	the probability of topic z given expert v_i
$\theta_{q_j z}$	the probability of topic z given query q_j

ten courses to teach. The topics corresponding to the courses (also expertise of the teachers) can be "machine learning", "data mining", "computational theory", etc. Each teacher v_i has different expertise degrees on the topics, characterized by θ_{v_i} and each course q_j also has a relevance distribution on different topics, characterized by θ_{q_j}. To assign teachers to courses, ideally the assigned teachers' expertise to each course should cover the topic of the course, and all the teachers should have a load balance with each other as well.

Contributions In this paper, we formally define the problem of expertise matching and propose a constraint-based optimization framework to solve the problem. Specifically, the expertise matching problem is transformed to a convex cost flow problem and the objective is then to find a feasible flow with minimum cost under certain constraints. We theoretically prove that the proposed framework can achieve an optimal solution and develop an efficient algorithm to solve it. We conduct experiments on two different genres of tasks: conference paper-reviewer assignment and course-teacher assignment. Experimental results validate the effectiveness and efficiency of the proposed approach. We have applied the proposed method to help assign reviewers to papers for a top conference. Feedbacks from the conference organizers confirm the usefulness of the proposed approach.

II. THE CONSTRAINT-BASED OPTIMIZATION FRAMEWORK

A. Basic Idea

The main idea of our approach is to formulate this problem in a constraint-based optimization framework. Different constraints can be formalized as penalty in the objective function or be directly taken as the constraints in the optimization solving process. For solving the optimization framework, we transform the problem to a convex cost network flow problem, and present an efficient algorithm which guarantees the optimal solution.

B. The Framework

Now, we explain the proposed approach in detail. In general, our objective can be viewed from two perspectives. On the one hand, we try to maximize the relevance between experts and queries; on the other hand, we try to satisfy the given constraints. Formally, we denote the set of experts to answer query q_j as $V(q_j)$, and the set of queries assigned to expert v_i as as $Q(v_i)$. Further, we denote the matching score (relevance) between expert v_i and query q_j as R_{ij}. Therefore, a basic objective function can be defined as follows

$$\text{Max} \sum_{v_i \in V} \sum_{q_j \in Q(v_i)} R_{ij} \tag{1}$$

The objective function can be equivalently written as $\sum_{q_j \in Q} \sum_{v_i \in V(q_j)} R_{ij}$. In different applications, the constraints can be defined in different ways. Here we use several general constraints to explain how the proposed framework can incorporate the different constraints.

The first constraint is that each query should be assigned to exactly m experts. For example, in the paper-reviewer assignment task, each paper should be assigned to 3 or 5 reviewers. This constraint can be directly added into the optimization problem. Formally, we have:

$$\textbf{ST1}: \ \forall q_j \in Q, |V(q_j)| = m \tag{2}$$

The second constraint is called as *expert load balance*, indicating that each expert can only answer a limited number of queries. There are two ways to achieve this purpose: define a *strict* constraint or add a *soft penalty* to the objective function. For *strict*, we add a constraint indicating that the number of assigned queries to every expert v_i should be equal or larger than a minimum number n_1, but be equal or smaller than a maximum number n_2. The *strict* constraint can be written as:

$$\textbf{ST2} \ (\textit{strict}): \ \forall v_i \in V, n_1 \leq |Q(v_i)| \leq n_2 \tag{3}$$

The other way is to add a soft penalty to the objective function (Eq. 1). For example, we can define a square penalty as $|Q(v_i)|^2$. By minimizing the sum of the penalty $\sum_i |Q(v_i)|^2$, we can achieve a *soft* load balance among all experts, i.e.:

$$\text{soft penalty:} \ \ \text{Min} \sum_{v_i \in V} |Q(v_i)|^2 \tag{4}$$

These two constraints can be also used together. Actually, in our experiments, soft penalty method gives better results than strict constraint. Combining them together can always yield a further improvement.

The third constraint is called *authority balance*. In real application, experts have different expertise level (authoritative level). Take the paper-reviewer assignment problem as an example. Reviewers may be divided into 2 levels: senior reviewers and average reviewers. Intuitively, we do not expect that all assigned reviewers to a paper are average reviewers. It is desirable that the senior reviewers can cover all papers to guide (or supervise) the review process. Without loss of generality, we divide all experts into K levels, i.e., $V^1 \cup V^2 \cup \cdots \cup V^k = V$, with V^1 representing experts of the highest authoritative level. Similar to *expert load balance*, we can define a strict constraint like $|V^1 \cap V(q_j)| \geq 1$, and also add a penalty function to each query q_j over the k-level experts. Following, we give a simple method to instantiate the penalty function:

$$\text{Min} \sum_{k=1}^{K} \sum_{j=1}^{N} |V^k \cap V(q_j)|^2 \tag{5}$$

The fourth constraint is called *topic coverage*. Also in the paper-reviewer assignment example, typically, we hope that the expertise of assigned reviewers to a paper can cover all topics of the paper. Our idea here is to define a reward function to capture the coverage degree. Specifically, the reward score is quantified by the number of times that an expert v_i has the expertise to answer a query q_j on a major topic z of this query, i.e.,

$$\text{Max} \sum_{z=1}^{T} \sum_{v_i \in V(q_j)} \mathbb{I}(\theta_{q_j z} > \tau_1)\mathbb{I}(\theta_{v_i z} > \tau_2) \tag{6}$$

where $\mathbb{I}(\theta_{q_j z} > \tau_1)$ is an indicator function, taking 1 when the condition is true or 0 when the condition is false. τ_1 and τ_2 are two thresholds, indicating that we only consider the major topics of query q_j and expert v_i. Intuitively, if every aspect of the query is covered by all assigned experts, we will have a maximum reward score.

The last constraint is called *COI avoidance*. In many cases, we need to consider the conflict-of-interest (COI) problem. For example, an author, of course, should not review his own or his coauthors' paper. This can be accomplished through employing a binary $M \times N$ matrix U. An element with value of 0, i.e., $U_{ij} = 0$, represents expert v_i has the conflict-of-interest with query q_j. A simple way is to multiply the matrix U with the matching score R in (Eq.1).

Finally, by incorporating Eq. 4-6 and the COI matrix U into the basic objective function (Eq. 1), we can result in the following constrained optimization framework:

$$\begin{aligned}
\text{Max} \quad & \sum_{v_i \in V} \sum_{q_j \in Q(v_i)} U_{ij} R_{ij} - \sum_{k=1}^{K} (\mu_k \sum_{j=1}^{N} |V^k \cap V(q_j)|^2) \\
& -\beta \sum_{v_i \in V} |Q(v_i)|^2 + \lambda \sum_{q_j \in Q} \sum_{z=1}^{T} \sum_{v_i \in V(q_j)} \mathbb{I}(\theta_{q_j z} > \tau_1)\mathbb{I}(\theta_{v_i z} > \tau_2) \\
s.t. \quad & \forall q_j \in Q, |V(q_j)| = m \\
& \forall v_i \in V, n_1 \leq |Q(v_i)| \leq n_2
\end{aligned} \tag{7}$$

where λ, β and μ_k are lagrangian multipliers, used to trade off the importance of different components in the objective function.

Now the problem is how to define the topic distribution θ, how to calculate the pairwise matching score R_{ij}, and how to optimize the framework.

C. Modeling Multiple Topics

The goal of topic modeling is to associate each expert v_i with a vector $\theta_{v_i} \in \mathbb{R}^T$ of T-dimensional topic distribution, and to associate each query q_j with a vector $\theta_{q_j} \in \mathbb{R}^T$. The topic distribution can be obtained in many different ways. For example, in the paper-reviewer assignment problem, each reviewer can select their expertise topics from a predefined categories. In addition, we can use statistical topic modeling [4][12] to automatically extract topics from the input data. In this paper, we use the topic modeling approach to initialize the topic distribution of each expert and each query.

To extract the topic distribution, we can consider that we have a set of M expert documents and N query documents (each representing an expert or a query). An expert's document can be obtained by accumulating the content information related to the expert. For example, we can combine all publication papers as the expert document of a reviewer, thus expert v_i's document can be represented as $d_i = \{w_{ij}\}$. Each query can also be viewed as a document. Then we can learn these T topic aspects from the collection of expert documents and query documents using a topic model such as LDA [4]. We use the Gibbs sampling algorithm [8] to learn the topic distribution θ_{v_i} for each expert and each query.

D. Pairwise Matching Score

We employ language model to calculate the pairwise matching score. With language model, the matching score R_{ij} between expert v_i and query q_j is interpreted as a probability $R_{ij}^{LM} = p(q_j|d_i) = \prod_{w \in q_j} p(w|d_i)$, with

$$P(w|d_i) = \frac{N_{d_i}}{N_{d_i} + \lambda_D} \cdot \frac{tf(w, d_i)}{N_{d_i}} + (1 - \frac{N_{d_i}}{N_{d_i} + \lambda_D}) \cdot \frac{tf(w, \mathbf{D})}{N_{\mathbf{D}}} \tag{8}$$

where N_{d_i} is the number of word tokens in document d_i, $tf(w, d_i)$ is the number of occurring times of word w in d_i, $N_{\mathbf{D}}$ is the number of word tokens in the entire collection, and $tf(w, \mathbf{D})$ is the number of occurring times of word w in the collection $\mathbf{D}$. λ_D is the Dirichlet smoothing factor and is commonly set according to the average document length in the collection [21].

Our previous work extended LDA and proposed the ACT model [19] to generate a topic distribution. By considering the learned topic model, we can define another matching score as

$$R_{ij}^{ACT} = p(q_j|d_i) = \prod_{w \in q_j} \sum_{z=1}^{T} P(w|z, \phi_z) P(z|d, \theta_{d_i}) \tag{9}$$

Further, we define a hybrid matching score by combining the two probabilities together

$$R_{ij}^{H} = R_{ij}^{LM} \times R_{ij}^{ACT} \tag{10}$$

E. Optimization Solving

To solve the objective function (Eq. 7), we construct a convex cost network with lower and upper bounds imposed on the arc flows. Figure 1 illustrates the constructing process, as described in algorithm 1. Q_j indicates a query node and V_i indicates an expert node. Q_{jk} indicates query q_j being assigned to an expert of expertise level k. S and T are two virtual nodes(source and sink of the network flow). The edge in the constructed network corresponds to the constraints we want to impose. Therefore, the problem of finding the optimal match between experts and queries becomes how to find a feasible configuration to minimize the cost of flow in the network. The problem (also referred to as the convex cost flow problem) can be solved by transforming it to an equivalent minimum cost flow problem [1]. We claim that the minimum cost flow of the network gives an optimal assignment with respect to (Eq. 7).

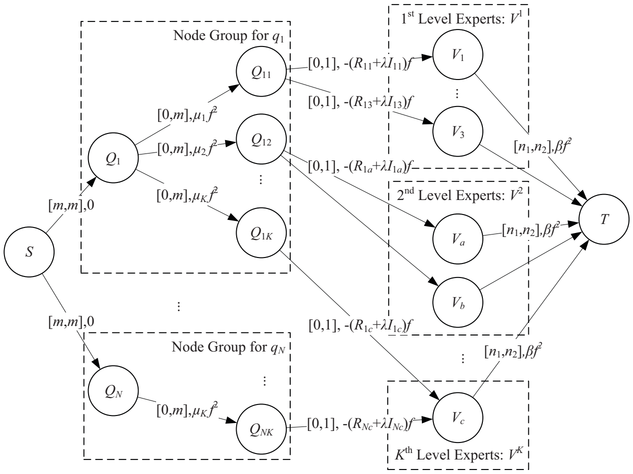

Node Group for q_1
1^{st} Level Experts: V^1
Q_{11}
Q_{12}
Q_{1K}
Q_1
S
$[m,m]$,0
$[0,m]$,$\mu_1 f^2$
$[0,m]$,$\mu_2 f^2$
$[0,m]$,$\mu_K f^2$
[0,1], -$(R_{11}+\lambda I_{11})f$
[0,1], -$(R_{13}+\lambda I_{13})f$
[0,1], -$(R_{1a}+\lambda I_{1a})f$
[0,1], -$(R_{1c}+\lambda I_{1c})f$
V_1
V_3
$[n_1,n_2]$,βf^2
2^{nd} Level Experts: V^2
V_a
V_b
T
$[m,m]$,0
Node Group for q_N
Q_N
$[0,m]$,$\mu_K f^2$
Q_{NK}
[0,1], -$(R_{Nc}+\lambda I_{Nc})f$
V_c
K^{th} Level Experts: V^K

Figure 1. The construction of convex-cost network flow according to objective function (Eq. 7). Every arc in the network is associated with lower and upper bound $[l, u]$ and a convex function of the arc flow f.

Theorem 1. *Algorithm 1 gives an optimal assignment.*

Proof: First, the minimum convex cost flow problem (MCCF) can be formulate as an optimization problem:

$$\begin{aligned} \text{Min} \quad & \textstyle\sum_{(a,b)\in E(G)} C_{ab}\Big(f(a,b)\Big) \\ s.t. \quad & \forall a \in V(G), \textstyle\sum_{b:(a,b)\in E(G)} f(a,b) = \textstyle\sum_{b:(b,a)\in E(G)} f(b,a) \\ & \forall (a,b) \in E(G), l_{ab} \le f(a,b) \le u_{ab} \end{aligned} \tag{11}$$

The model is defined on directed network $G = (V(G), E(G))$ with lower bound l_{ab}, upper bound u_{ab} and a convex cost function $C_{ab}\big(f(a,b)\big)$ associated with every arc (a, b).

Algorithm 1: Optimization solving algorithm.

```
Input: The set of experts $V$; the set of queries $Q$; the
       matching score matrix $R_{M\times N}$; the COI matrix
       $U_{M\times N}$; Number of expertise level $K$; $m$, $n_1$, $n_2$
       as described above.
Output: An assignment of experts to queries maximizing
        objective function 7.
Create a network $G$ with source node $S$ and sink node $T$;
foreach $q_j \in Q$ do
    Create $K + 1$ nodes, denoted as $Q_j, Q_{j1}, \dots, Q_{jK}$
    respectively;
    Add an arc from source node $S$ to node $Q_j$, with
    zero cost and flow constraint $[m, m]$;
    Add an arc from node $Q_j$ to $Q_{jk}$, with square cost
    function $\mu_k f^2$ and flow constraint $[0, m]$;
end
foreach $v_i \in V$ do
    Create a node $V_i$;
    Add an arc from $V_i$ to sink node $T$, with square cost
    function $\beta f^2$ and flow constraint $[n_1, n_2]$;
end
foreach $v_i \in V, q_j \in Q$, s.t. $U_{ij} = 1$ do
    $k$ = expert level of $v_i$;
    Add an arc from $Q_{jk}$ to $V_i$, with linear cost function
    $-(R_{ij} - \lambda I_{ij})f$ and flow constraint $[0, 1]$;
end
Compute the minimum cost flow on $G$;
foreach $v_i \in V, q_j \in Q$, s.t. $U_{ij} = 1$ do
    $k$ = expert level of $v_i$;
    if flow $f(Q_{jk}, V_i) = 1$ then Assign query $q_j$ to
    expert $v_i$;
end
```

Now we prove that minimizing (Eq. 11) on the graph G constructed in algorithm 1 is equivalent to maximizing (Eq. 7). For simplicity, we use I_{ij} to denote $\sum_{z=1}^{T} \mathbb{I}(\theta_{q_j z} > \tau_1)\mathbb{I}(\theta_{v_i z} > \tau_2)$. For the constructing process, we see a feasible flow on G is mapping to a query-expert assignment. The flow from S to Q_j indicates the number of experts assigned with query q_j, and the flow from V_i to T indicates the number of queries assigned to expert v_i. And the cost between V_i and T is corresponding to the *load balance* soft penalty function (Eq. 4). The meaning of the flow from Q_j to Q_{jk} is the number of kth-level experts assigned to q_j, thus we impose a square cost function $\mu_k \cdot f^2$ on the arcs which is equivalent to the negative of the *authority balance* penalty. The flow from Q_{jk} to V_i means we assign query q_j to expert v_i, it is easy to find that no query will be assigned to the same expert twice since we give an upper bound of 1 on the arc, while the cost is equivalent to the negative of matching score and topic coverage score. Therefore, our problem can be reduced to a equivalent MCCF problem, where the objective function of MCCF problem (Eq. 11) is the negative form of (Eq. 7). ■

In practice, it is not necessary to add all (Q_{jk}, V_i) arcs. To reduce the complexity of the algorithm, we first greedily generate an assignment and preserve corresponding arcs, then keep only $c \cdot m$ arcs for Q_{jk} and $c \cdot n_2$ arcs for V_i which have highest matching score (c is a fixed constant). We call this process *Arc-Reduction*, which will reduce the number of arcs in the network without influencing the performance too much. To process large scale data, we can also leverage the parallel implementation of convex cost flow [3].

F. Online Matching

After an automatic expertise matching process, the user may provide feedbacks. Typically, there are two types of user feedbacks: (1) pointing out a mistake match; (2) specifying a new match. Online matching aims to adjust the matching result according to the user feedback. One important requirement is how to perform the adjustment in real time. In our framework, we provide online interactive adjustment without recalculating the whole cost flow. For both types of feedbacks, we can accomplish online adjustment by canceling some flows and augmenting new assignments in our framework. We give algorithm 2 to consider the first type of feedback, which still produces an optimal solution.

Algorithm 2: Online matching algorithm.

```
Input: A minimum cost network flow $f$ on $G$
       corresponding to the current assignment; an
       inappropriate match ($v_i$,$q_j$) to be removed.
Output: A new assignment.
2.1 $k$ = expert level of $v_i$;
2.2 if $f(Q_{jk}, V_i) = 1$ then
2.3     Construct the residual network $G(f)$;
2.4     Compute the shortest path $P_{back}$ from $T$ to $S$ on
        $G(f)$ which contains backward arc $(V_i, Q_{jk})$;
2.5     Cancel(roll back) 1 unit of flow along $P_{back}$ and
        update $G(f)$;
2.6     Remove arc $(Q_{jk}, V_i)$ from $G$ and update $G(f)$;
2.7     Compute shortest augmenting path path $P_{aug}$ from $S$
        to $T$;
2.8     Augment 1 unit of flow along $P_{aug}$;
2.9 end
```

Lemma 1 (**Negative Cycle Optimality Conditions**). *[1] A feasible solution f^* is an optimal solution of the minimum cost flow problem if and only if it satisfies the negative cycle optimality conditions: namely, the residual network $G(f^*)$ contains no negative cost cycle.*

Theorem 2. *Algorithm 2 produces an optimal solution in the network without assignment (q_j, v_i).*

Proof: According to Lemma 1, the residual network $G(f)$ contains no negative cost cycle since the given flow f has the minimum cost. In algorithm 2, we remove the inappropriate match (v_i,q_j) and adjust the network flow in line 2.3-2.5. Denote the feasible flow in the network after line 2.5 as f'. According to the SAP (Short Augmenting Path) algorithm of cost flow, if f' has the minimum cost(i.e., $G(f')$ contains no negative cycle), the algorithm will give the optimal solution. We show the optimality of f' by contradiction. Assume $G(f')$ contains a negative cycle C, C must intersect with the shortest path P_{back} computed on line 2.3, since the original $G(f)$ contains no negative cycle. Thus merging C into path P_{back} will generate a shorter path, which contradicts with the assumption that P_{back} is shortest. Therefore, f' has the minimum cost. Accordingly, algorithm 2 gives the optimal solution after augmenting a new assignment. ■

III. Experimental Results

The proposed approach for expertise matching is very general and can be applied to many application to align experts and queries. We evaluate the proposed framework on two different genres of expertise matching problems: paper-reviewer assignment and course-teacher assignment. All data sets, code, and detailed results are publicly available.[1]

A. Experimental Setting

Data Sets The paper-reviewer data set consists of 338 papers and 354 reviewers. The reviewers are program committee members of KDD'09 and the 338 papers are those published on KDD'08, KDD'09, and ICDM'09. For each reviewer, we collect his/her all publications from an academic search system Arnetminer[2][20] to generate the expertise document. As for the COI problem, we generate the COI matrix U according to the coauthor relationship in the last five years and the organization they belong to. Finally, we set that a paper should be reviewed by $m = 5$ experts, and an expert at most reviews $n_2 = 10$ papers.

In the course-teacher assignment, we manually crawled graduate courses from the department of Computer Science (CS) of four top universities, namely CMU, UIUC, Stanford, and MIT. In total, there are 609 graduate courses from the fall semester in 2008 to 2010 spring, and each course is instructed by 1 to 3 teachers. Our intuition is that teachers' research interest often match the graduate courses he/she is teaching. Thus we still use the teachers' recent (five years) publications as their expertise documents, while the course description and course name are taken as the query.

On both data sets, we employ topic model [4] to extract the topic distribution of each expert and each query. We performed topic model learning with the same setting, topic number $T = 50$, $\alpha = 50/T$, and $\beta = 0.01$. The code for learning the topic model is also online available.[1]

Baseline Methods and Evaluation Metrics We employ a greedy algorithm as the baseline. The greedy algorithm assigns experts with highest matching score to each query, while keeping the load balance for each expert (i.e., $|Q(v_i)| \leq n_2$) and avoiding the conflict of interest.

In the paper-reviewer problem, as there are no standard answers, in order to quantitatively evaluate our method, we define the following metrics:

[1] http://www.arnetminer.org/expertisematching/

[2] http://arnetminer.org

Matching Score (MS): It is defined as the accumulative matching score.

$$MS = \sum_{v_i \in V} \sum_{q_j \in Q(v_i)} U_{ij} R_{ij}$$

Load Variance (LV): It is defined as the variance of the number of papers assigned to different reviewers.

$$LV = \sum_{i=1}^{M} \left(|Q(v_i)| - \frac{\sum_{i=1}^{M} |Q(v_i)|}{M} \right)^2$$

Expertise Variance (EV): It is defined as the variance of the number of top level reviewers assigned to different papers.

$$EV = \sum_{j=1}^{N} \left(|V(q_j) \cap V^1| - \frac{\sum_{j=1}^{N} |V(q_j) \cap V^1|}{N} \right)^2$$

In the course-teacher assignment experiment, we extract the real assignment as the ground-truth, thus we perform the evaluation in terms of Precision.

Experiment Setting We tune the different parameters to analyze the influence on the accumulative matching score. We also evaluate the efficiency performance of our proposed approach. All the experiments are carried out on a PC running Windows XP with Intel Core2 Quad CPU Q9550(2.83GHz), 3.2G RAM.

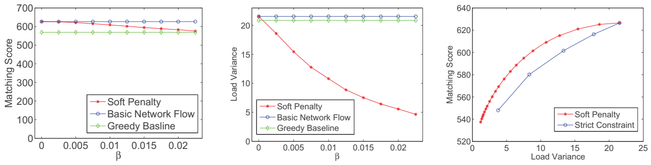
B. Experiment Results

Paper-reviewer Assignment Experiment In the experiment, we first set $\mu = \mathbf{0}$ and tune the parameter β to find out the effects of soft penalty function. Figure 2 (a) illustrates how soft penalty function influences the matching score with different β. We see that the matching score decreases slightly with β increasing. Figure 2 (b) shows the effects of load variance with β varied. We see that the load variance changes very fast toward balance.

In figure 2 (c), we compare the two different methods to achieve load balance, namely, strict constraint and soft penalty. The two LV-MS curves are respectively generated by setting different minimum numbers n_1 for strict constraint and varying the weight parameter β for soft load balance penalty. The curves show that soft penalty outperforms strict constraint towards load balance.

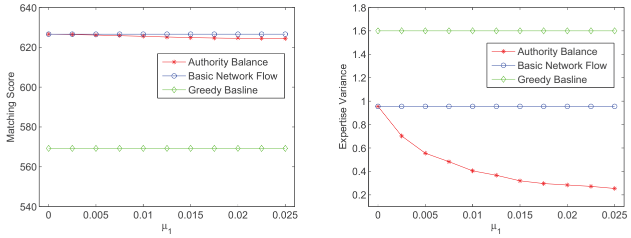
Then we set β to 0 to test the effects of authority balance. Experts are divided into 2 levels base on their H-index, and we set $\mu_2 = 0$ to consider the balance of the senior reviewers only. Figure 3 presents the accumulative matching score (a) and expertise variance (b) with μ_1 varied.

Further, we analyze the effects of different constraints. Specifically, we first remove all constraints (using Eq. (1) only), and then add the constraints one by one in the order (Load balance, Authority balance, Topic coverage, and COI). In each step, we perform expertise matching using our approach. Table II lists the accumulative matching score obtained in each step. We see that the load balance constraint will reduce the expertise matching score, while the other constraints have little negative effect. This is because senior experts are often good at many aspects, thus assigned with heavy load in traditional matching. In out approach the decrease of matching score in the load balance constraint is to balance the work load of senior experts.



(a) (b) (c)

Figure 2. Figure (a) and (b) illustrate how soft penalty function influences the matching score(MS) and load variance with different β respectively. Figure (c) gives a comparison between soft penalty function and strict constraint methods towards load balance.



(a) (b)

Figure 3. Matching score (MS) and expertise variance (EV) with μ_1 varied.

Table II
EFFECTS OF DIFFERENT CONSTRAINTS ON MATCHING SCORE.

Constraint	Matching Score
Basic objective function (Eq. 1)	635.51
+ Load Balance *soft penalty* with β = 0.02	592.83
+ Authority Balance with $\mu = (0.02, 0)^T$	599.37
+ Topic Coverage with $\tau_1 = \tau_2 = 0.08, \lambda = 0.1$	599.37
+ COI	590.14

Finally, we evaluate the efficiency performance of the proposed algorithm. We compare the CPU time of the original optimal algorithm and the version with *Arc-Reduction*. As shown in Figure 4, the *Arc-Reduction* process can significantly reduce the time consumption. For example, when setting $c = 12$ in this problem, we can achieve a $> 3\times$ speedup without any loss in matching score.

We further use a case study (as shown in table III and IV) to demonstrate the effectiveness of our approach. We see that the result is reasonable. For example, Lise Getoor, whose research interests include relational learning, is assigned with a lot of papers about social network.

Course-Teacher Assignment Experiment Figure 5 (a) shows the assignment precision in the course-teacher assignment task by our approach and the baseline method, and

Table IV
LIST OF REVIEWERS FOR 5 RANDOM PAPERS.

Paper	Assigned reviewers
Audience selection for on-line brand advertising: privacy-friendly social network targeting	C. Lee Giles, Jie Tang, Matthew Richardson, Hady Wirawan Lauw, Elena Zheleva
Partitioned Logistic Regression for Spam Filtering	Rong Jin, Chengxiang Zhai, Saharon Rosset, Masashi Sugiyama, Annalisa Appice
Structured Learning for Non-Smooth Ranking Losses	Xian-sheng Hua, Tie-yan Liu, Hang Li, Yunbo Cao, Lorenza Saitta
Unsupervised deduplication using cross-field dependencies	Chengxiang Zhai, Deepak Agarwal, Max Welling, Donald Metzler, Oren Kurland
The structure of information pathways in a social communication network	C. Lee Giles, Wolfgang Nejdl, Melanie Gnasa, Michalis Faloutsos,Cameron Marlow

Table V
CASE STUDY: PROFESSORS WITH MANY COURSES ASSIGNED IN UIUC(2008, FALL - 2010, SPRING)

Professor	Pub Papers	Courses assigned(baseline)	Courses assigned(our approach)
Jose Meseguer	237	23 courses Database Systems (2008,spring) Programming Languages and Compilers (2008,spring) Iterative and Multigrid Methods (2009,spring) Programming Languages and Compilers (2009,spring)	7 courses Programming Languages and Compilers (2008,spring) Programming Language Semantics (2008,spring) Programming Languages and Compilers (2008,fall) Programming Languages and Compilers (2009,spring)
ChengXiang Zhai	117	18 courses Computer Vision (2009,spring) Text Information Systems (2009,spring) Stochastic Processes and Applic (2009,fall) Computer Vision (2008,spring)	7 courses Text Information Systems (2008,spring) Stochastic Processes and Applic (2008,fall) Text Information Systems (2009,spring) Stochastic Processes and Applic (2009,fall)

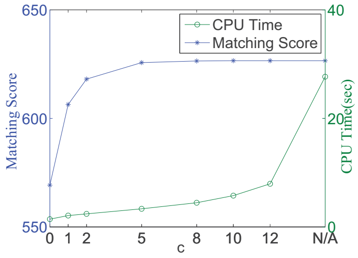
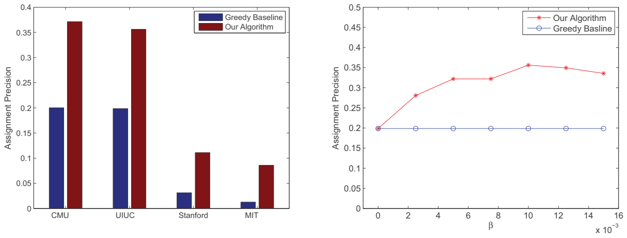


Figure 4. Efficiency performance (s).

Table III
EXAMPLE ASSIGNED PAPERS TO THREE REVIEWERS.

Reviewer	Assigned papers
Lise Getoor	Evaluating Statistical Tests for Within-Network Classifiers of ... Discovering Organizational Structure in Dynamic Social Network Connections between the lines: augmenting social networks with text MetaFac: community discovery via relational hypergraph factorization Relational learning via latent social dimensions Influence and Correlation in Social Networks
Wei Fan	Mining Data Streams with Labeled and Unlabeled Training Examples Vague One-Class Learning for Data Streams Active Selection of Sensor Sites in Remote Sensing Applications Name-ethnicity classification from open sources Consensus group stable feature selection Categorizing and mining concept drifting data streams
Jie Tang	Co-evolution of social and affiliation networks Influence and Correlation in Social Networks Feedback Effects between Similarity and Social Influence ... Mobile call graphs : beyond power-law and lognormal distributions Audience selection for on-line brand advertising: privacy-friendly ...

(b) shows the effects of the parameter β on the precision on UIUC data. The precision is defined as the ratio of the number of correct assignments(consistent with the ground truth data) over total number of assignments. As Figure 5 (a) shows, in all the data sets we collect from top universities, our algorithm outperforms the greedy method greatly. And in Figure 5 (b), as the β increases, the precision of our approach increases in general and decreases slowly after it exceeds the peak value. The peak value is more than 50 percents larger than the initial precision, which validates the effectiveness of the soft penalty approach.



(a) Course assignment results

(b) Precision vs. β on UIUC data

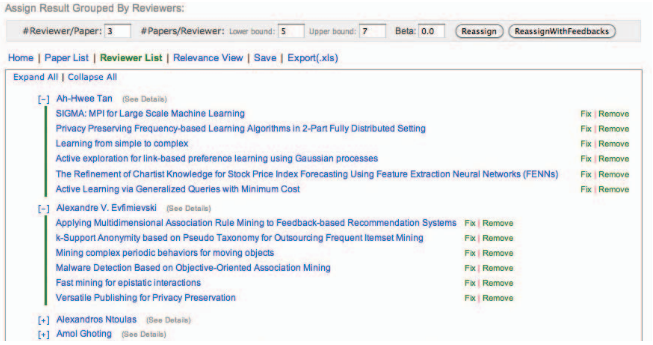
Figure 5. Course-Teacher Assignment performance(%).

We conduct a further analysis on the UIUC data set. As Table V shows, some professors with publications in various domains, are likely to be assigned with many courses in the baseline algorithm. But in real situation, most professors, though with various background, want to focus on several directions. Thus some courses should be assigned to younger teachers. While in our algorithm, the situation is much better. And we can see that each teacher is assigned with a reasonable load as well as a centralized interest.

C. Online System

Based on the proposed method, we have developed an online system for paper-reviewer suggestions, which is available at [3]. Figure 6 shows an screenshot of the system. The input is a list of papers (with titles, abstracts, authors, and organization of each author) and a list of conference program

[3] http:/review.arnetminer.org/


Home | Paper List | Reviewer List | Relevance View | Save | Export(.xls)
Expand All | Collapse All
Reassign
ReassignWithFeedbacks

Assign Result Grouped By Reviewers:

#Reviewer/Paper: 3 #Papers/Reviewer: Lower bound: 5 Upper bound: 7 Beta: 0.0

[-] Ah-Hwee Tan (See Details)
- SIGMA: MPI for Large Scale Machine Learning — Fix | Remove
- Privacy Preserving Frequency-based Learning Algorithms in 2-Part Fully Distributed Setting — Fix | Remove
- Learning from simple to complex — Fix | Remove
- Active exploration for link-based preference learning using Gaussian processes — Fix | Remove
- The Refinement of Chartist Knowledge for Stock Price Index Forecasting Using Feature Extraction Neural Networks (FENNs) — Fix | Remove
- Active Learning via Generalized Queries with Minimum Cost — Fix | Remove

[-] Alexandre V. Evfimievski (See Details)
- Applying Multidimensional Association Rule Mining to Feedback-based Recommendation Systems — Fix | Remove
- k-Support Anonymity based on Pseudo Taxonomy for Outsourcing Frequent Itemset Mining — Fix | Remove
- Mining complex periodic behaviors for moving objects — Fix | Remove
- Malware Detection Based on Objective-Oriented Association Mining — Fix | Remove
- Fast mining for epistatic interactions — Fix | Remove
- Versatile Publishing for Privacy Preservation — Fix | Remove

[+] Alexandros Ntoulas (See Details)

[+] Amol Ghoting (See Details)

Figure 6. Screenshot of the online system.

committee (PC) members. We use the academic information stored in ArnetMtiner to find the topic distribution for each paper and each PC member [20]. With the two input lists and the topic distribution, the system automatically finds the match between papers and authors. As shown in Figure 6, there are 5-7 papers assigned to each PC member and the number of reviewers for each paper is set as 3. The system will also avoid the conflict-of-interest (COI) according to the coauthorship and co-organization relationship. In addition, users can provide feedbacks for online adjustment, by removing or confirm (fix) an assignment.

IV. Conclusion and Future Work

In this paper, we studied the problem of expertise matching in a constraint-based framework. We formalized the problem as a minimum convex cost flow problem. We theoretically proved that the proposed approach can achieve an optimal solution and developed an efficient algorithm to solve it. Experimental results on two different types of data sets demonstrate that the proposed approach can effectively and efficiently match experts with the queries. Also we present an algorithm to optimize the framework according to user feedbacks in real time. We are also going to apply the proposed method to several real-world applications.

Acknowledgment

The work is supported by the Natural Science Foundation of China (No. 60703059, No. 60973102), Chinese National Key Foundation Research (No. 60933013), National High-tech R&D Program (No. 2009AA01Z138).

References

[1] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. *Network Flows: Theory, Algorithms, and Applications*. Prentice Hall, 1993.

[2] K. Balog, L. Azzopardi, and M. de Rijke. Formal models for expert finding in enterprise corpora. In *SIGIR'2006*, pages 43–55, 2006.

[3] P. Beraldi, F. Guerriero, and R. Musmanno. Parallel algorithms for solving the convex minimum cost flow problem. *Computational Optimization and Applications*, 18(2):175–190, 2001.

[4] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

[5] D. Conry, Y. Koren, and N. Ramakrishnan. Recommender systems for the conference paper assignment problem. In *RecSys'09*, pages 357–360, 2009.

[6] S. T. Dumais and J. Nielsen. Automating the assignment of submitted manuscripts to reviewers. In *SIGIR'92*, pages 233–244, 1992.

[7] H. Fang and C. Zhai. Probabilistic models for expert finding. In *ECIR'07*, pages 418–430, 2007.

[8] T. L. Griffiths and M. Steyvers. Finding scientific topics. In *PNAS'04*, pages 5228–5235, 2004.

[9] D. Hartvigsen, J. C. Wei, and R. Czuchlewski. The conference paper-reviewer assignment problem. *Decision Sciences*, 30(3):865–876, 1999.

[10] C. B. Haym, H. Hirsh, W. W. Cohen, and C. Nevill-manning. Recommending papers by mining the web. In *IJCAI'99*, pages 1–11, 1999.

[11] S. Hettich and M. J. Pazzani. Mining for proposal reviewers: lessons learned at the national science foundation. In *KDD'06*, pages 862–871, 2006.

[12] T. Hofmann. Probabilistic latent semantic indexing. In *SIGIR'99*, pages 50–57, 1999.

[13] M. Karimzadehgan and C. Zhai. Constrained multi-aspect expertise matching for committee review assignment. In *CIKM'09*, pages 1697–1700, 2009.

[14] M. Karimzadehgan, C. Zhai, and G. Belford. Multi-aspect expertise matching for review assignment. In *CIKM'08*, pages 1113–1122, 2008.

[15] N. D. Mauro, T. M. A. Basile, and S. Ferilli. Grape: An expert review assignment component for scientific conference management systems. In *IEA/AIE'05*, pages 789–798, 2005.

[16] D. Mimno and A. McCallum. Expertise modeling for matching papers with reviewers. In *KDD'07*, pages 500–509, 2007.

[17] D. Petkova and W. B. Croft. Hierarchical language models for expert finding in enterprise corpora. *International Journal on Artificial Intelligence Tools*.

[18] Y.-H. Sun, J. Ma, Z.-P. Fan, and J. Wang. A hybrid knowledge and model approach for reviewer assignment. In *HICSS'07*, pages 47–47, 2007.

[19] J. Tang, R. Jin, and J. Zhang. A topic modeling approach and its integration into the random walk framework for academic search. In *ICDM'08*, pages 1055–1060, 2008.

[20] J. Tang, J. Zhang, L. Yao, J. Li, L. Zhang, and Z. Su. Arnetminer: Extraction and mining of academic social networks. In *KDD'08*, pages 990–998, 2008.

[21] C. Zhai and J. Lafferty. A study of smoothing methods for language models applied to ad hoc information retrieval. In *SIGIR'01*, pages 334–342, 2001.