Published in final edited form as:
Inf Process Manag. 2012 July 1; 48(4): 725–740. doi:10.1016/j.ipm.2011.09.004.

Integer Linear Programming for Constrained Multi-Aspect Committee Review Assignment

Maryam Karimzadehgan and **ChengXiang Zhai**
Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, IL, 61801

Maryam Karimzadehgan: mkarimz2@illinois.edu; ChengXiang Zhai: czhai@cs.illinois.edu

Abstract

Automatic review assignment can significantly improve the productivity of many people such as conference organizers, journal editors and grant administrators. A general setup of the review assignment problem involves assigning a set of reviewers on a committee to a set of documents to be reviewed under the constraint of review quota so that the reviewers assigned to a document can collectively cover multiple topic aspects of the document. No previous work has addressed such a setup of *committee review assignments* while also considering matching *multiple aspects* of topics and expertise. In this paper, we tackle the problem of committee review assignment with multi-aspect expertise matching by casting it as an integer linear programming problem. The proposed algorithm can naturally accommodate any probabilistic or deterministic method for modeling multiple aspects to automate committee review assignments. Evaluation using a multi-aspect review assignment test set constructed using ACM SIGIR publications shows that the proposed algorithm is effective and efficient for committee review assignments based on multi-aspect expertise matching.

Keywords

Topic Models; Review Assignment; Algorithms; Combinatorial Optimization; Evaluation Metrics

1. Introduction

Review assignment is a common task that many people such as conference organizers, journal editors, and grant administrators would have to do routinely. The task usually has to do with assigning a certain number of experts to review a research paper or a grant proposal and judge its quality. Currently such assignments are mostly done manually by editors or conference organizers. Some conference systems support bidding on papers by reviewers, shifting some of the assignment task to the reviewers, but it is still mostly a time-consuming manual process. Manual assignment is not only time-consuming, but also often biased. For example, in bidding, a popular paper may attract more bids than a non-popular paper. Also, the matching of expertise would presumably be more accurate for reviewers whose expertise is largely known to an editor or a conference program chair, but less accurate for reviewers not so familiar to the review assigner.

Publisher's Disclaimer: This is a PDF file of an unedited manuscript that has been accepted for publication. As a service to our customers we are providing this early version of the manuscript. The manuscript will undergo copyediting, typesetting, and review of the resulting proof before it is published in its final citable form. Please note that during the production process errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

To help people who manage review assignments to improve their productivity and to potentially correct any bias in review assignment, researchers have studied automatic review assignment. In most of the studies (e.g., [2, 5, 11, 15, 26]), the problem is considered as a retrieval problem, where the query is a paper (or a grant proposal) to be reviewed and a candidate reviewer is represented as a text document. One main drawback of all these works is that a paper or proposal is matched as a *whole* component without taking into account the multiple subtopics.[1] As a result, if a document contains multiple subtopics, existing methods would not attempt to assign reviewers to cover all the subtopics; instead, it is quite possible that all the assigned reviewers would cover the major subtopic quite well, but not covering any other subtopic. Since a paper or proposal often contains multiple subtopics, such simplistic view of matching often leads to non-optimal assignments. For example, if a paper is about applying machine learning techniques to develop new retrieval models for Web search, ideally it should be reviewed by reviewers with expertise to cover three subtopics, i.e., “machine learning”, “retrieval models”, and “Web search”. However, the existing methods may assign three reviewers that are all experts on machine learning but do not know well about Web search.

In our previous study [19], we have proposed three general strategies for multi-aspect review assignment: *Redundancy Removal*, *Reviewer Aspect Modeling*, and *Paper Aspect Modeling*. These proposed methods for multi-aspect review assignment are shown to increase the aspect coverage for automatic review assignment. However, in this work, the assignment of reviewers to each paper is done *independently* without considering the whole committee. This makes it hard to balance the review load among a set of reviewers, and may result in assigning too many papers to a reviewer with expertise on a popular topic. In many applications of review assignment, such as assigning conference papers to the reviewers on the program committee and assigning grant proposals to a panel of reviewers, there is a limit on the number of reviews to be done by each single reviewer. To balance the review load and conform to the review quota of each reviewer, it is necessary to set up the problem as to simultaneously assign papers to all the reviewers on a committee with consideration of review-load balancing. We call this problem Committee Review Assignment (CRA).

Although the CRA problem has been previously studied by a few researchers [3, 14, 25, 35], none of them has considered multiple aspects of topics and expertise in matching papers with reviewers.

In this paper, we study a novel setup of the CRA problem where the goal is to assign a pool of reviewers on a committee to a set of papers based on multi-aspect expertise matching (i.e., the assigned reviewers should cover as many subtopics of the paper as possible) and with constraints of review quota for reviewers. We call this problem Constrained Multi-Aspect Committee Review Assignment (CMACRA). Given the limited review capacity, limited reviewer expertise, and the need for matching papers with reviewers based on multiple aspects, the CMACRA problem is quite challenging.

We propose to solve the CMACRA problem by casting it as an integer programming problem and present an integer linear programming formulation of the problem. In our optimization setup, matching of reviewers with a paper is done based on matching of multiple aspects of expertise. Therefore, the assigned reviewers would not only have the required expertise to review a paper but also can cover all the aspects of a paper in a complementary manner. In addition, such an assignment does not unduly burden any individual reviewer with too many papers and at the same time a full set of reviewers is assigned to each paper. All these preferences and requirements are captured through a set of

[1] Aspect and subtopic will be used interchangeably throughout the paper.

constraints in the integer programming formulation, and the objective function maximizes average coverage of multiple aspects. The proposed algorithm is quite general; it allows us to set a potentially different review quota for each reviewer and can naturally accommodate any probabilistic or deterministic method for modeling multiple aspects to automate committee review assignments.

To evaluate the effectiveness of our algorithm, we use the measures and gold standard data introduced in our previous work [19]. Our experiment results show that the proposed committee review assignment algorithm is quite effective for the CMACRA task and outperforms a heuristic greedy algorithm for assignment. The results also show that with a reasonable number of aspects, our algorithm is sufficiently efficient to handle the review assignments for a relatively large conference. Although we mainly evaluated our algorithm in the context of conference review assignment, it is general and can be applied to other review assignment tasks such as assigning grant proposals to a group of panelists for reviewing.

The rest of the paper is organized as follows. We first discuss related work in Section 2 and the problem of constrained multi-aspect committee review assignment in Section 3. We then propose our algorithm to optimize the committee assignment task in Section 4. In Section 5, we describe our experimental design and evaluation measures. In Section 6, we present our evaluation results. We conclude the paper in Section 7.

2. Related Work

Review assignment has been studied in several previous works. Dumais and Nielsen [11] did an early study using Latent Semantic Indexing (LSI). A recent work by Hettich and Pazzani [15] is a recommendation system which recommends panels of reviewers for NSF grant applications by using TF-IDF weighting measure. Basu et al.[2] use Web to find abstracts of papers written by reviewers and then TF-IDF weighting is used for ranking. Biswas and Hasan[5] use topics defined based on a domain-ontology to represent papers and reviewers and use TF-IDF weighting to rank reviewers. Mimno and McCallum [26] propose an Author-Persona-Topic (APT) model, in which topical components are learned and then each author’s papers are divided into many “personas”. Rodriguez and Bollen [29] present a method that propagates a particle swarm over the co-authorship network, starting with the authors cited by the submitted paper. Karimzadehgan et al.[19] consider multiple aspects of papers and multiple expertise of reviewers and they match papers with reviewers based on aspect matching. Our work is different from all these works in that we consider the CMACRA problem, i.e., considering constraints associated with papers and reviewers and maximizing the coverage of multiple topics in a paper by the assigned reviewers.

Committee review assignment has been studied in [3, 14, 25, 34, 35]; Benferhat and Lang [3] present a heuristic approach in which they define regulations to reduce the set of feasible solutions, and they use preferences and constraints to order the feasible assignments and select the best one. Authors of [25] present another heuristic evolutionary algorithm by maximizing the match between the reviewers’ expertise and paper topics. Hartvigsen and Wei [14] use minimum cost network flow. For each reviewer and paper, they define a weight denoting the degree of expertise of a reviewer for that paper. They then adopted the idea of finding the assignment by solving a maximum weighted-capacitated transportation problem on the network. Taylor [35] solves the global constrained optimization problem using Linear Programming. The information needed for optimization is based on the preferences indicated by the area chairs and affinities. However, their optimization framework can not solve the constrained multi-aspect matching. Considering the constrained multi-aspect in the optimization framework makes the problem more challenging. Sun et al.

[34] solve a review assignment task for assigning experts to review projects for funding. Their method is a combination of mathematical decision models with knowledge rules. Their aim is to maximize the total expertise levels of the reviewers assigned to the project with heuristic rules they define. They ask each individual reviewer to express their level of expertise with an integer value in {0, …, 3}for the discipline areas they have defined. These works cannot maximize the coverage of topics in the paper and reduce the redundancy in covering each topic by assfigned reviewers.

Integer linear programming has been applied to solve several other information management tasks. Roth and Yih [31] use integer programming for the inference procedure in conditional random field in order to support general structures. In [13, 24], integer programming is used to secure sensitive knowledge from being shown in patterns extracted by frequent pattern mining algorithms. Clarke and Lapata [8] use integer programming to compress sentences while preserving the meaning of the sentences. Berretta and et al. [4] use integer programming to set up a unified framework for feature selection process for molecular classification of cancer.

Expert Finding is another area related to our work. In Expert Finding, a ranked list of experts with expertise on a given topic is retrieved. Several probabilistic models and language models are presented in [1, 12, 28]. Some other systems use graph-based ranking algorithms for determining the connections between people. These methods [7, 10, 23, 33] are applied on corpora of email communications to locate experts. Another method for expert finding is “Voting” [22]. Using the ranked list of retrieved documents, Macdonald and Ounis [22] model the ranking of candidates as a voting process using the retrieved documents and documents in experts’ profile. They then apply data fusion techniques to generate the final ranking for experts with this model.

Our work is also related to topic modeling [6, 16, 21, 30, 36]. Topic models are generative probabilistic models for collections of text. A collection of data is often modeled as a finite mixture of an underlying set of topics, and each topic is modeled with a multinomial word distribution. We use topic models to discover multiple aspects of expertise of reviewers, which are then used in the proposed optimization framework to enable matching of reviewers and papers based on multiple aspects.

The basic idea of our work has been published in a short paper of CIKM [18]. However, this short paper does not have a complete description of the proposed algorithms due to the page limit. This paper is a significant expansion of the previous short paper to add a comprehensive review of related work, more complete description of the algorithms, new experiment results for inferred topics, complexity analysis of the algorithms, and a new set of scalability experiments and results.

3. Constrained Multi-Aspect Committee Review Assignment

Informally, the problem of Constrained Multi-Aspect Committee Review Assignment (CMACRA) is to reflect a very common application scenario such as conference review assignment where the goal is to assign a set of reviewers to a set of papers so that (1) each paper will be reviewed by a certain number of reviewers; (2) each reviewer would not review more than a specified number of papers; (3) the reviewers assigned to a paper have the expertise to review the paper; and (4) the combined expertise of the reviewers assigned to a paper would cover well all the subtopics of the paper. To the best of our knowledge, no previous work on review assignment has considered all these requirements. Indeed, in most previous work, the problem has been simplified as using each paper as a query to retrieve a set of right reviewers with expertise matching the paper, e.g. [2, 5, 26], which mostly addresses (3), and may also address (4) if multiple aspects are considered. Some other works have addressed (1) and (2), but failed to consider (4).

We propose to solve this problem by casting it as an optimization problem. As a computation problem, CMACRA takes the following information as the input:

- A set of n papers: $\mathcal{P} = \{p_1, \ldots, p_n\}$ where each p_j is a paper.
- A set of m reviewers: $\mathcal{R} = \{r_1, \ldots, r_m\}$ where each r_i is a reviewer.
- A set of reviewer quota limits: $NR = \{NR_1, \ldots, NR_m\}$ where NR_i is the maximum number of papers a reviewer r_i can review.
- A set of numbers of reviewers to be assigned to a paper: $NP = \{NP_1, \ldots, NP_n\}$ where NP_j is the number of reviewers that should be assigned to paper p_j.

And the output is a set of assignments of reviewers to papers, which can be represented as an $n \times m$ matrix M with $M_{ij} \in \{0, 1\}$ indicating whether reviewer r_i is assigned to paper p_j. ($M_{ij} = 1$ means that reviewer r_i has been assigned to review paper p_j.)

To respect the reviewer quota limits and to ensure that each paper gets the right number of reviewers, we require M to satisfy the following two constraints:

$$\forall i \in [1, m], \sum_{j=1}^{n} M_{ij} \leq NR_i \quad (1)$$

$$\forall j \in [1, n], \sum_{i=1}^{m} M_{ij} = NP_j \quad (2)$$

Naturally, we assume that there are sufficient reviewers to review all the papers subject to the quota constraints. That is,

$$\sum_{j=1}^{n} NP_j \leq \sum_{i=1}^{m} NR_i \quad (3)$$

In addition, we would also like the review assignments to match the expertise of the assigned reviewers with the topic of the paper well, and ideally, the reviewers can cover all the subtopics of the paper. Formally, let $\boldsymbol{\tau} = (\boldsymbol{\tau}_1, \ldots, \boldsymbol{\tau}_K)$ be a set of K subtopics that can characterize the content of a paper as well as the expertise of a reviewer, and $\boldsymbol{\tau}_k$ is a specific topic. These subtopics can either be from the list of topic keywords that are typically provided in a conference management system to facilitate review assignments or automatically learned via statistical topic models such as Probabilistic Latent Semantic Analysis (PLSA) [16] from the publications of reviewers as done in the previous work [19].

The subtopics in the first case are usually designed by human experts that run a conference such as program chairs, and both the authors and reviewers would be asked to choose some specific keywords to describe the content of the paper and the expertise of the reviewer, respectively. Thus we would have access to *deterministic* assignments of subtopics to the papers and reviewers. In the second, a subtopic can be characterized by a word distribution, and in general, a paper and a reviewer would get a *probabilistic* assignment of subtopics to characterize the content of the paper and the expertise of the reviewer. Since deterministic assignments can be regarded as a special case of probabilistic assignments when the probability is either 1.0 or 0.0, we thus only need to consider the probabilistic assignments of subtopics.

Thus we assume that we have two matrices P and R available, which represent our knowledge about the subtopics of the content of a paper and the subtopics of the expertise of a reviewer, respectively. P is a $n \times K$ matrix where P_{jk} is a probability indicating how likely subtopic τ_k represents the content of paper p_j. R is a $m \times K$ matrix where R_{ik} is a probability indicating how likely subtopic τ_k represents the expertise of reviewer r_i. Clearly, when P_{jk} and R_{ik} take binary values, we would end up having deterministic assignments of subtopics to papers and reviewers.

Since there are potentially a very large number of possible committee assignments, our solution space is huge and a bruce force enumeration of all the possible solutions would not be feasible. Thus a main technical challenge in solving the CMACRA problem is to develop a tractable algorithm that can respect all the constraints and optimize the matching of the expertise of reviewers and the topics of papers with consideration of multiple aspects of topics and expertise. We discuss how we solve this challenge in the next section.

4. Algorithms for CMACRA

Our main idea for solving the CMACRA problem is to cast it as a tractable optimization problem, i.e., an integer linear programming problem. Before we present such an optimization algorithm, we first present a non-optimal heuristic greedy algorithm, which only works for the scenario of deterministic subtopic assignments to papers and reviewers. We would also use this algorithm as a baseline to evaluate our integer linear programming algorithm.

4.1. A greedy algorithm

A straightforward way to solve the CMACRA problem is to use a greedy algorithm, in which we would optimize the review assignments for each paper iteratively. The algorithm only works for the scenario of deterministic assignments of topics to papers and reviewers. It works as follows:

First, the papers are decreasingly sorted according to the number of subtopics they contain, i.e., the paper with the largest number of subtopics is ranked first. We then start off with this ranked list of the papers. At each assignment stage, the best reviewer that can cover most subtopics of the paper is assigned. In addition, the review quota and paper quota are checked, i.e., the number of papers assigned to each reviewer and the number of reviewers assigned to each paper. If the review quota is reached, that reviewer is removed from our reviewer pool; the same is done when the paper quota is satisfied. This process is repeated until reviewers are assigned to all the papers. Using notations and input data described in section 3, Figure 1 describes the algorithm more formally.

This greedy algorithm does not always lead to an optimal solution. Intuitively, since at each assignment stage, it greedily assigns the best reviewer that can cover most aspects of the paper, it may “consume” all the reviewers with rare expertise on a subtopic quickly when processing the top-ranked papers in the ranked list. As a result, such reviewers would no longer be available later when we encounter a paper that really needs reviewers with such rare expertise. In the next subsection, we formulate the problem in a more principled way as an integer linear programming problem which can achieve optimized review assignments.

4.2. An integer linear programming algorithm

In this subsection, we propose a formulation to use Integer Linear Programming (ILP) [20, 27] to solve the CMACRA problem. The use of ILP, in particular, Binary Integer Programming (BIP), for solving this problem is natural because what we want to compute is binary assignments of papers to reviewers, and BIP can naturally compute such binary assignments to optimize a linear objective function subject to linear constraints. The main motivation for framing the problem with linear constraints and linear objective function is to ensure that the optimization problem can be solved efficiently.

4.2.1. The ILP formulation—Linear Programming (LP) is a way to optimize a linear objective function, subject to linear equality and linear inequality constraints. It is a way to achieve the best outcome (maximum profit or lowest cost) given a list of linear constraints. In Linear Programming, if all the unknown variables are required to be integers, then the problem is called an *Integer Linear Programming* (ILP). *Binary Integer Programming*(BIP) is the special case of integer linear programming where variables are required to be zero or one.

We now show how the problem of CMACRA can be formulated as a Binary Integer Programming problem. First, we observe that in our formal definition of the CMACRA problem in Section 3, we have already naturally introduced several constraints, and the problem can be cast as an optimization problem where we seek an optimized assignment matrix M that would satisfy all the constraints as well as optimize the multi-aspect matching of expertise of reviewers and the content of each paper. Thus M_{ij} would naturally become variables in the definition of the ILP problem.

How should we define the objective function to be optimized? Intuitively, the function must capture our desire to match the expertise of reviewers with the content of a paper based on multiple subtopics. Unfortunately the variables M_{ij} cannot help us directly because they are not defined on subtopics. Thus, we need to introduce auxiliary variables to connect M_{ij} with subtopic assignments. However, it is not immediately clear how to define such auxiliary variables and formulate a linear objective function.

We propose to introduce the following set of auxiliary variables:

$$\{t_{jk}\}_{j\in[1,n],k\in[1,K]}$$

where $t_{jk} \in [0, NP_j]$ is an integer indicating the number of assigned reviewers that can cover subtopic τ_k for paper p_j. This allows us to define the following linear objective function to maximize:

$$Maximize(\sum_{j=1}^{n}\sum_{k=1}^{K} t_{jk}) \quad (4)$$

Intuitively, this objective function says that we would prefer assignments that maximize the coverage of all the subtopics (the inner sum) for all the papers (the outer sum).

Now we still need to connect t_{jk} with the review assignment matrix M. Specifically, we need an upper-bound constraint for t_{jk} so that we will not end up having a trivial solution that simply gives each t_{jk} its maximum value NP_j. This upper-bound is intuitively the actual number of reviewers assigned to paper p_j that can cover subtopic τ_k according to M. Thus if the subtopic assignment is completely binary, which means that the element values of both matrices P and R are binary, it is relatively easy to see that we should have the following set of n inequality linear constraints, each for a paper:

$$\forall j \in [1,n], k \in [1,K], P_{jk} t_{jk} \le \sum_{l=1}^{m} R_{lk} M_{lj} \quad (5)$$

Recall that P_{jk} refers to whether paper p_j covers topic τ_k and here we assume that $P_{jk} \in \{0, 1\}$. Thus, this constraint can be interpreted as follows: For paper p_j, if the paper covers subtopic τ_k (i.e., $P_{jk} = 1$), t_{jk} can be as large as the actual number of reviewers assigned to paper p_j that can cover subtopic k. Note that $R_{lk} = 1$ iff the expertise of reviewer r_l covers subtopic τ_k, and $M_{lj} = 1$ iff reviewer r_l is assigned to paper j, thus $\sum_{l=1}^{m} R_{lk} M_{lj}$ captures the total amount of expertise from the reviewers assigned to paper p_j.

Since this upper-bound is itself an integer, and our objective function encourages choosing a larger value for t_{jk}, in our solution, we will end up having t_{jk}'s that would actually satisfy the equality.

What if our subtopic assignment is probabilistic or fuzzy? In such a case, the element values of P and R can be any positive real numbers. It turns out that the inequality above for t_{jk} would still makes sense, though our solution would unlikely satisfy the equality. Specifically, we may regard the right hand side of the inequality as to compute the weighted combined coverage of subtopic τ_k by all the assigned reviewers according to M, thus it still serves as a meaningful upper-bound. Similarly, the left hand side can also be interpreted as the desired coverage of subtopic τ_k since if P_{jk} is large, it would mean that paper p_j is very much likely about subtopic τ_k, and thus we would demand more coverage about τ_k. Thus the inequality constraint also makes sense in the case of non-deterministic assignments of subtopics to papers and reviewers as in the case of learning subtopics from the publications of reviewers.

Adding additional constraints introduced in Section 3, the complete ILP formulation of the CMACRA problem is shown in Figure 2.

Our objective function indicates that for each paper we want to maximize both the number of covered topics and the number of reviewers that can cover each topic in the paper. Constraint **C1** shows that each variable is either one or zero where one means reviewer r_i is assigned to paper p_j. Constraint **C2** indicates that t_{jk} is an integer with minimum zero and maximum NP_j, where NP_j is the number of reviewers that should be assigned to paper p_j. Constraint **C3** indicates that each paper p_j will be assigned precisely NP_j reviewers. Constraint **C4** indicates that each reviewer r_i can review up to NR_i papers. Finally, constraint **C5** requires that variable t_{jk} be constrained by the actual coverage of subtopic t_k by the assigned reviewers to paper p_j according to M.

If we have knowledge about conflict of interest of reviewers, we may further add the following additional constraint:

C6: $M_{ij} = 0$, if reviewer r_i has conflict of interest with paper p_j.

4.2.2. Solving the ILP—Once our problem is formulated as an ILP problem, we can potentially use many algorithms to solve it. In our experiments, we use the commercial ILOG CPLEX 11.0 package[2] to solve our CMACRA problem. ILOG CPLEX simplex optimizers provide the power to solve linear programs with millions of constraints. Specifically, CPLEX uses "Branch-and-Cut" [32] algorithm which is an exact algorithm consisting of a combination of a *cutting plane method* with a *Branch-and-Bound algorithm* to solve integer linear programs. The idea of the “Branch-and-Bound” algorithm [32] is to take a problem which is difficult to be solved directly and decompose it into smaller problems in such a way that a solution to a smaller problem is also a solution to the original problem. This decomposition is done recursively until it can be solved directly or is proven not to lead to an optimal solution. In order to solve integer linear programs, the integrality constraints on the variables are removed; this transforms the subproblem to a linear programming problem, called the *LP relaxation* of the subproblem, which is easy to solve to optimality. This linear program without integer constraints is solved using simplex algorithm [27].

The “Branch-and-Cut” algorithm is essentially a “Branch-and-Bound” algorithm with an additional *Cutting* step. When the optimal solution is found but the final value for an integer variable is non-integer, a cutting plane step tries to add additional constraints. In other words, the *Cutting* step is to generate valid inequalities for the integer hull of the current subproblem and add them to the LP relaxation of the subproblem. At this point, the problem is divided into two subproblems: one is to explore values greater than or equal to the smallest integer greater than the current value, and the other is to explore values less than or equal to the next lesser integer. These new linear programs are then solved using the simplex method and the process repeats until a solution satisfying all the integer constraints (or binary constraints) is found. For complete information about the algorithm, a reader is referred to [32].

4.3. Complexity analysis of the algorithms

The complexity of the greedy algorithm in the worst case is $O(n * \log n + n * m * \log m)$, where n and m are the number of papers and the number of reviewers, respectively. The $n * \log n$ part is the *sort time* for papers according to the number of topics they have. In line 3 of the algorithm, for each paper, we also sort the reviewers according to covering most topics of the papers. So, the complexity for that part is: $n * m * \log m$.

Integer linear programming (ILP) is classified as NP-complete problems [27]. The worst case complexity of our ILP algorithm is $O(2^{n*m} * N^{n*k})$, where k is the number of subtopics, and N is the maximum number of reviewers that should be assigned to a paper, i.e., $N = \max_{i \in [1,n]} NP_i$. For binary variables which are $n * m$, a node is split into two children based on rounding of a variable with a fractional part. So, at the worst case, we will see 2^{n*m} nodes. In addition, for a general integer variable x with domain $0 \ldots N$, we are typically branching using cuts such as $x \leq a$ or $x \geq a + 1$, where $0 \leq a \leq N$. So, we also get two children, but it takes approximately $\log_2 N$ levels to pin down a specific value for x in the worst case. So we have $2^{\log_2 N} = N$ paths in the tree from any node where we first start branching on x to any node where we are done branching on x (assuming we do all x branching before moving on to another variable). So the overall complexity is $O(2^{n*m} * (N)^{n*k})$. Although the worst case complexity of the optimized algorithm is exponential, it works empirically well in practice and as we will show later in Section 6.3, the ILP algorithm for CMACRA is sufficiently efficient to handle review assignments for reasonably large conferences.

4.4. Modeling and assigning subtopics

The proposed algorithms are based on the assumption that we have available a set of subtopics τ and the assignments of them to the papers and reviewers (i.e., P and R). This is a

[2] http://www.ilog.com/products/cplex/

realistic assumption for a conference review system that requires all authors and reviewers to choose subtopic keywords, in which case we generally would have a binary P and R.

In applications where we do not have such input from authors and reviewers, we may learn subtopics from the publications of reviewers and compute probabilistic assignments of subtopics to papers and reviewers as has been done in our previous work [19]. The proposed ILP algorithm can be easily applied in this case because constraint **C5** can take non-binary element values of P and R. Indeed, the algorithm is completely general to take any meaningful positive weights in P and R as long as these weights are comparable (e.g., when they are all probability values).

In our experiments, we experiment with both application scenarios: (1) We have complete knowledge about subtopics and their assignments to papers and reviewers. (2) We have no such knowledge but we have publications of reviewers so we can learn subtopics and their assignments. In the second scenario, we can use the Probabilistic Latent Semantic Indexing (PLSA)[16] approach to model and learn subtopics, i.e., to set matrices P and R as described in [19]. This method has been shown to outperform other general strategies in our previous work [19] for solving the multi-aspect review assignment.

We now describe this approach in more detail.

We assume that we have available some text documents to represent the expertise of each reviewer, which can be, e.g., the publications of the reviewer. Formally, let $T_R = \{Tr_1, \ldots, Tr_m\}$ to denote the set of text documents representing the expertise of m reviewers, respectively and $T_Q = \{Tq_1, \ldots, Tq_n\}$ to denote the text of the n papers to be reviewed.

PLSA *explicitly* models subtopics in the publications of reviewers with mixture language models. If we can somehow directly model the potential topic aspects and try to match a paper with reviewers based on their topic-aspect representation, we may potentially achieve better results.

To implement this idea, we assume that there is a space of K topic aspects, each characterized by a unigram language model. Let $\tau = (\tau_1, \ldots, \tau_K)$ be a vector of topics. τ_k is a unigram language model and $p(w|\tau_k)$ is the probability of word w according to topic τ_k. Consider that we have a set of m reviewer expertise documents (each representing a reviewer). We can then learn these K topic aspects from this set of reviewer expertise documents using a topic model such as PLSA [16], which is described below.

The log-likelihood of the whole collection according to PLSA is:

$$\log p(T_R|\tau) = \sum_{i=1}^{m}\sum_{w\in V} c(w, T_{r_i}) \log\left(\sum_{a=1}^{K} p(a|\theta_i)p(w|\tau_a)\right) \quad (6)$$

where V is the set of all the words in our vocabulary, $c(w, Tr_i)$ is the count of word w in Tr_i, and $p(a|\theta_i)$ is the probability of selecting topic aspect τ_a for document Tr_i. Intuitively, $p(a|\theta i)$ encodes the coverage of different topic aspects in the expertise document of reviewer i. $p(w|\tau_a)$ is the probability of word w according to topic τ_a.

We may use Expectation-Maximization (EM) algorithm [9] to compute the maximum likelihood estimate of all the parameters including τ and θ_i's.

Once we obtain θ_i which is a distribution over all the possible topic aspects $p(a|\theta_i)$ ($a = 1, \ldots, K$), we can think of it as a new way to represent our document Tr_i in terms of topic aspects. With a similar model to the one presented above, we can also estimate the subtopic coverage in our query papers, T_Q, with the main difference being that we have τ already known so we only need to estimate θT_Q which would give us a distribution over subtopics for each query paper (e.g., $p(a|\theta q_1)$ for the first query, where $a = 1, \ldots, K$). The distributions of topic aspects for each reviewer (i.e., $p(a|\theta_i)$) and for each query (i.e., $p(a|\theta q_i)$) are used to fill out matrices R and P, respectively.

5. Experiment Design

In this section, we describe the data set and evaluation measures.

5.1. Dataset

Evaluating of such a system is very challenging. Since the actual assignments of reviewers to papers are confidential, we cannot use such data to evaluate the effectiveness of our methods. Even if we had such information, it would not necessarily indicate the best matching. The data set in "Enterprise Track" in TREC cannot be used for our task since we need a collection for matching reviewers with papers based on multiple aspects not only for retrieving experts/reviewers. The only data set available for evaluating multi-aspect review assignment is the one we created in our previous work [19] [3]. We thus use this data set in our experiments. The details of the dataset is as follows:

This data set was constructed based on the abstract papers of ACM proceedings from years 1971–2006 from the ACM digital library[4]. We have only considered abstract papers in conferences such as World Wide Web, SIGIR and CIKM in order to consider only IR researches for evaluating purposes. Authors of these papers are considered as prospective reviewers. A profile for each author in this pool is created by concatenation of all papers written by that specific author. If a paper has more than one author, that paper is replicated, one for each author. Considering the authors having published more than three papers in these years as the prospective reviewers reduced the number of reviewers to 189. The papers in SIGIR 2007 are used to simulate papers that are to be reviewed. There are 73 papers with at least two aspects. We experimented with both abstracts and full papers.

To create a gold standard for evaluating our approaches, an information retrieval expert identified 25 major subtopics based on the topic areas in the Call for Papers of ACM SIGIR in most recent five years and session titles in the recent ACM SIGIR conferences. The expert then reads the abstracts of all the 73 test papers and all the 189 expertise profiles, and assign relevant expertise/topic aspects to each paper and each reviewer. This serves as a gold standard to evaluate our methods.

5.2. Evaluation Measures

We use the evaluation measures defined in the previous work [19]. For completeness, we include the description of these measures taken from reference [19].

It is desirable to include reviewers covering many topic aspects so that they can collectively cover all aspects in the query. This can be captured by the *Coverage* measure which tells us whether we can cover all aspects of the query by the assigned reviewers. Consider a query with n_A topic aspects $A_1, \ldots, A_{n_A}$ and let n_r denote the number of distinct topic aspects that these n assigned reviewers can cover. *Coverage* can be defined as the percentage of topic aspects covered by these n reviewers:

[3] Available at http://timan.cs.uiuc.edu/data/review.html

[4] http://www.acm.org/dl

$$Coverage \equiv \frac{n_r}{n_A} \quad (7)$$

We would also prefer an assignment where each aspect is covered by as many reviewers as possible. Given the same level of coverage, we would prefer an assignment where each aspect is covered by as many reviewers as possible. Intuitively, this is related to the overall confidence of the assigned reviewers in reviewing each topic aspect of the "query paper." Thus the *Confidence* measure is defined to capture the redundancy of reviewers in covering each aspect and is defined as follows:

Let $A_1^{'}, \ldots, A_{n_r}^{'}$ be the n_r distinct aspects covered by n retrieved reviewers and $n_{A_i'}$ be the number of reviewers that can cover aspect $A_i^{'}$, then *Confidence* measure is as follows:

$$Confidence \equiv \frac{\sum_{i=1}^{n_r} \frac{n_{A_i'}}{n}}{n_r} \quad (8)$$

The confidence values are normalized with the number of *covered* topic aspects n_r, thus a high confidence value may be obtained by intentionally covering fewer aspects. This is why confidence alone would not be so meaningful and it should be combined with the coverage. In this sense, the relation between coverage and confidence is similar to that between precision and recall. A perfect assignment should have both high coverage and high confidence. One way to combine coverage and confidence is to normalize confidence over *all* aspects rather than just the *covered* aspects to compute an *Average Confidence*. Since a missed aspect would decrease the average confidence, it serves as a combination of coverage and confidence. Using the notations introduced earlier, the *Average Confidence* measure is defined as follows:

$$Average\ Confidence \equiv \frac{\sum_{i=1}^{n_A} \frac{n_{A_i}}{n}}{n_A} \quad (9)$$

We normalize all three measures by considering their corresponding optimal values that can be gained from our gold standard data set. Since measures are different; we describe algorithms for finding optimal values specifically for each measure. In general, finding the exact optimal values for these measures is NP-hard [38], so we opt to use greedy algorithms to compute an approximate value.

For finding the *optimal Coverage*, we use an MMR-based approach when selecting the next wide-coverage reviewer; i.e., we first pick the reviewer that covers the most number of aspects of the paper, then we take away the covered aspects and pick the next reviewer to cover as many of the remaining aspects as possible. For finding the *optimal Confidence*, we first pick the aspect, e.g. A_1, of the paper that most of the reviewers in the collection can cover it, we then select other reviewers that can cover aspect A_1 but do not cover any of the remaining aspects of the paper (to minimize n_r). For finding *optimal Average Confidence*, we select each reviewer to work independently to cover most aspects of the paper. In all cases, we follow some previous work [17, 38] and normalize each value with its corresponding optimal value to make the values more comparable across different query papers.

6. Experiment Results

We first examine the question whether the ILP algorithm can effectively solve the CMACRA problem. We evaluate the ILP algorithm in the following two scenarios:

1. Known subtopic assignments: This is to simulate a common application scenario where reviewers are asked to choose from a set of pre-defined subtopics a subset to describe their expertise, and the authors are asked to do the same for their papers. In this case, we can set each element of matrices P and R to either 1 or 0 based on the selections made by reviewers and authors.

2. Inferring P and R through text mining: When we do not have access to the selections of subtopics by reviewers and authors, we can use Probabilistic Topic Models to infer subtopics and matrices P and R based on publications of reviewers as done in our previous work [19]. In this case, we can set P and R based on the inferred subtopic assignments for reviewers and papers, and the elements of both P and R will generally have a real value between 0 and 1.0.

Since our work is the first study of CMACRA, there is no existing baseline to be compared with. We thus want to see whether the ILP algorithm can achieve better aspect coverage than the heuristic greedy algorithm. Since the greedy algorithm only works for the scenario of known subtopics assignments, we only compare ILP with the greedy algorithm in the first scenario where we can use our gold standard data set to obtain subtopic assignments (i.e., matrices P and R).

6.1. Known subtopic assignments

Our data set has 189 reviewers and 73 papers, thus $n = 73$, and m=189. The total number of topics is 25, thus K=25. Average numbers of topics covered by a reviewer and a paper are 5 and 3, respectively. In all the experiments, we assign three reviewers to each paper, which is meant to resemble a typical setup of conference review assignment. Since there might exist multiple *optimal solutions*, i.e., different assignments of reviewers to papers will lead to the same *optimal value* for the objective function for the ILP algorithm, we generate 10 such solutions and average over all. We do the same for the greedy algorithm.

We compare the ILP algorithm with the greedy algorithm by varying parameter values in three different ways. The results are shown in Figures 3, 4 and 5 for the three measures, Coverage, Confidence, and Average Coverage, respectively. In all the figures, we plot the average performance over all the papers and also show standard deviations for different solutions with error bars. Please note that invisible error bars mean zero variance.

First, in Figure 3, we show the results from varying the number of reviewers and allowing each reviewer to review up to 5 papers. This is to simulate the variation of the size of a program committee. From the figure, we can see that as we increase the number of reviewers, the performance of both algorithms is getting better, which is expected because as the resources, i.e., the number of reviewers increase, better reviewers can be assigned to papers. In addition, the performance of the ILP algorithm is much better than the greedy algorithm.

Second, we fix the number of reviewers to 30, and vary the number of papers each reviewer can review. This is to simulate the variation of a reviewer’s review quota. In order to avoid bias, we repeat the sampling process (selecting 30 reviewers) for 10 times and get the average. The results are shown in Figure 4. As we increase the number of papers that each reviewer can get, we are also increasing the resources, and as a result, the performance of both algorithms becomes better. Also, comparing two algorithms shows that the ILP algorithm has a better performance than the greedy algorithm. The reason is that the greedy algorithm does not always lead to an optimal solution; since at each assignment stage, it greedily assigns the best reviewer that can cover most aspects of the paper, it may consume all the reviewers with rare expertise on a subtopic quickly. However, since our ILP algorithm is formulated to achieve the global optimization, the problem we just mentioned will not happen in the ILP algorithm.

Finally, we compare the two algorithms when we have very limited resources, i.e., the maximum number of reviewers is 10 for 73 papers. Again we randomly select 10 reviewers and we repeat the sampling process for 10 times and get the average. Each paper gets 3 reviewers and the number of papers that each reviewer can get is calculated according to the number of reviewers that we have. For example, if we have 5 reviewers, each should get 44 papers. The results are shown in Figure 5. As we increase the resources, i.e., the number of reviewers, the performance of both algorithms becomes better and the performance of the ILP algorithm is still better than the greedy algorithm even when the resources are *very limited* since it does global optimization.

Statistical Significant Tests—The performance results of the ILP algorithm shown in Figures 3, 4 and 5 are statistically significant for all cases compared to the greedy algorithm. We measured their statistical significance using a Wilcoxon Signed-Rank test [37] at the level of 0.05. The results indicate that the difference between the ILP algorithm and the greedy algorithm is statistically significant for all three evaluation measures (Coverage, Confidence and Average Confidence).

All these results confirm that the ILP optimization algorithm achieves better performance than the baseline greedy algorithm in terms of all the three measures.

6.2. Inferred subtopic assignments

In this subsection, we look at the scenario when the subtopics are unknown, and focus on studying how to optimize performance when we use PLSA to predict subtopics. We follow the work in [19] and learn the subtopics for both papers and reviewers using the PLSA model. We further look into cases when we only need to infer one of the two matrices, i.e., one of the matrices are learned with PLSA model and the other one is gained from gold standard data.

While our ILP algorithm can be directly applied to the probabilistic assignments of subtopics given by PLSA, intuitively, not all the predictions are reliable, especially the low-probability ones. Thus we also experimented with pruning low probability values learned with PLSA (i.e., setting low probability elements of P and R to zero). For example, in Figure 6 (left), $cutoff_5$ means when we only keep the top 5 probability values out of 25 learned topics and prune the rest. The figure shows the result of the Coverage measure. The figure suggests having more topics such as 15 and 25 for reviewers and fewer topics for paper, i.e., $K \geq 4$ and $K \leq 7$ would help improve the performance. Additional observation from the figure is, having more topics for reviewers such as 25 leads to a more stable curve.

Figure 6 (right) shows the performance of the ILP algorithm in Average Confidence. We can observe the same trend as in the curves for the Coverage. Since the variance of multiple solutions is small, to aid exposition, we do not show error bars in Figure 6.

Table 1 compares the results of applying ILP algorithm directly to the PLSA results (25 learned topics for both reviewer and paper) with the best result obtained from pruning low-probability topics. The best pruning result was obtained when reviewers' topics are set to 15 and papers' topics to 5. The results suggest that pruning is beneficial. Also, measuring their statistical significance using a Wilcoxon Signed-Rank test [37] indeed indicates that the difference between PLSA (Best Cutoff) and PLSA(No Cutoff) is statistically significant.

So far, we looked at the scenario when the subtopics for both papers and reviewers are unknown and used PLSA to predict subtopics. In order to have a better understanding of the our algorithm, we now further look into cases when we only need to infer one of the two matrices P and R.

6.2.1. Estimating subtopics for reviewers only with PLSA—

In the first scenario, we learn 25 topics (the same number of topics as in gold standard data) using PLSA and only predict topics for reviewers. That is, we assume that the authors of the papers choose the topic areas for their submitted paper, i.e., papers' topics are obtained from gold standard data. We use the special case where topics are binary and use a *Cutoff* to convert the topic weights learned with PLSA into binary values; for example, $Cutoff_5$ means that we would keep the top 5 probability values and set them to one and set the rest to zero.

The results are shown in Figure 7 (left). The figure also shows the precision and recall curves for the estimated topics based on PLSA. Since we know the true aspects from the gold standard data and we also have the estimated topics with PLSA model and Cutoff, we can easily calculate precision and recall at each given cutoff point. In these results, each paper gets 3 reviewers and each reviewer gets up to 5 papers to review. The interesting observation is that as Cutoff increases, the performance in the Coverage measure decreases. The reason can be explained based on precision and recall curves. When we have a perfect precision but low recall for reviewers, it means that we could not get true aspects with PLSA model but we also do not have noise for reviewers' topics. On the other hand, when we have a perfect recall but low precision, it means that we could get all the aspects for reviewers with PLSA model but we also have some noisy aspects which might mislead the optimized algorithm. In this figure, the performance is more sensitive to the precision, i.e., when the precision decreases the performance of the algorithm in the coverage measure also decreases, suggesting that we are probably introducing noise, i.e., we have considered a reviewer as an expert on a topic when the reviewer is not really an expert on the topic.

6.2.2. Estimating subtopics for papers only with PLSA—

In the second scenario, we predict the topics for papers with PLSA model and the topics for reviewers are obtained from gold standard data. In this experiment, we also assign 3 reviewers to each paper and each reviewer gets up to 5 papers to review. The results are shown in Figure 7 (right). An interesting observation is that as Cutoff increases, the performance in the Coverage measure also increases. This observation can be again explained with precision and recall curves for PLSA model. When we have a prefect recall but not prefect precision for papers, we may be overestimating the topics in the paper. In that case, the Coverage measure would be high because with PLSA model we could estimate not only the true aspects in the paper but also some extra aspects which are noise. However, these noises do not hurt the performance much when we have enough resources, i.e., enough reviewers. On the other hand, when we have a perfect precision but low recall, it means that we would minimize the noise in the topics for the paper but we also miss some aspects. That is why the coverage would be lower when the precision is high. As a result, the recall curve has more influence on the Coverage measure for this scenario, i.e., the performance trend for the Coverage measure is the same as for recall curve.

6.3. Scalability of ILP

Finally, we study the scalability of our ILP algorithm. In the results reported so far, we have only evaluated our algorithms using the 73 papers in gold standard data. In reality, the number of submitted papers to a conference is much larger. Unfortunately there is no data set with more papers available for us to use [5], so we opt to study the scalability by generating synthetic data to simulate the scenarios of conference review assignments with larger number of submissions.

Specifically, we first select the number of topic areas be 25 (the same number of topics as in the gold standard data). We then get the average number of expertise areas for reviewers from the gold standard data which is 5 and the average number of topics in our query papers (73 papers) which is 3. These are used to specify how many topics out of 25 should be one for each reviewer and paper. So, we randomly choose 5 and 3 topics out of 25 to be the ones assigned to each reviewer and paper, respectively. In our experiments, we also vary the number of topics, i.e., 50 and 100. When we have 50 topics, we randomly select 10 and 6 out of 50 topics to be one for each reviewer and paper, respectively. Since the number of topics is doubled, i.e., from 25 to 50, the number of expertise topics for reviewers and papers will be doubled too, i.e., from 5 to 10 for reviewers and 3 to 6 for papers. The same is done when we have 100 topics.

The number of reviewers to be assigned to each paper is 3 (this is what is usually done in real conferences) and each reviewer can get up to 6 papers. Then, we vary the number of papers and reviewers. Consider that n is the number of papers, the minimum number of reviewers which are needed is $n * 3/6$ (3 is the number of reviewers assigned to each paper and 6 is the maximum number of papers that each reviewer can review.)

Figure 8 (left) shows the runtime of both the ILP and greedy algorithm as the number of papers increase. Please note that the runtime is only to generate one optimal solution. The algorithms run on Intel (R) Xeon (R) CPU, 1.6 GHZ, with 8 GB memory. The runtime for both algorithms increases when we have a large number of papers and topics as expected. Since the runtime for greedy algorithm is less than a minute which is not visible in Figure 8 (left), to aid exposition, we show the runtime of the greedy algorithm separately in Figure 8 (right) as well. Given the computational complexity of the ILP algorithm, this observation is intuitively expected; indeed, as we increase the number of papers, more time is needed to find the optimal assignment, because the number of variables is increased, as a result, the algorithm behaves exponentially to the number of variables. Since in most real conferences, the keyword list used for authors and reviewers usually does not have more than 50 keywords, the ILP algorithm is sufficiently efficient for use in real conferences with large number of submissions, e.g., 1000.

As expected, the greedy algorithm takes less time to run (less than a minute) but for the ILP algorithm, it takes more time to find an optimal solution for a large number of papers and topics. However, as discussed in Section 6.1, the assignment accuracy and quality of ILP is significantly better than the greedy algorithm. Since assignment of reviewers in a conference management system is usually not required to be run in real time, spending more time to get an optimal solution is worthwhile. Thus we can expect ILP to be more useful than the greedy algorithm.

6.4. Summary

Overall, our experiment results demonstrate the value of the ILP algorithm for multi-aspect committee review assignment. The main findings are:

[5] The unavailability of larger test sets also makes it impossible to evaluate the effectiveness of the ILP algorithm for a large number of papers.

- The ILP algorithm is significantly better than the greedy algorithm according to Wilcoxon Signed-Rank test for all three evaluation measures, i.e., Coverage, Confidence and Average Confidence as we the results in Figures 3, 4 and 5 indicate. The reason is that the ILP globally optimizes the assignment however the greedy algorithm greedily assigns reviewer at each assignment stage which may consume all the reviewers with rare expertise on a subtopic quickly.
- The ILP algorithm is scalable and it can be used for a large number of paper submissions and topics according to the results in Figure 8. The greedy algorithm takes less time to run, but this is at the cost of having much worse review assignment quality. Since assignment of reviewers in a conference management system is usually not required to be run in a real time, spending more time to an optimal solution is worthwhile as in the case of the ILP algorithm.
- When predicting topics with PLSA model for both reviewers and papers and pruning the low-probability values helps improve the performance as shown in table 1. The reason is that when predicting subtopics with PLSA model, intuitively, not all the predictions are reliable, especially the low-probability ones, so they can pruned to further improve the performance.
- Due to the lack of resources (i.e., data set), we are planning, to further evaluate these algorithms with more data sets, ideally through applying the algorithms to review assignments in a real conference.

7. Conclusions and Future Work

Review assignment is an important but time-consuming task. Automatic assignment of reviews is interesting for multiple reasons, including reducing the labor of the assigners and potentially improving the quality of assignments. In this paper, we studied a novel setup of the problem, i.e., committee review assignment based on multiple subtopics, where the assigned reviewers would not only have the required expertise to review a paper but can also cover all the aspects of a paper in a complementary manner satisfying all constraints.

We proposed two general algorithms for solving this problem, including greedy algorithm and ILP algorithm. We systematically tested the algorithms with previously created review-assignment data set. Experiment results show that the ILP algorithm is effective for increasing the coverage and confidence of topic aspects in committee assignment task, and outperforms the greedy algorithm significantly. The ILP algorithm is also sufficiently efficient to handle a large number of submissions in a normal conference.

The proposed algorithms are general, thus they can be applied to all review assignment tasks where we need to assign a set of reviewers to review a set of papers or proposals with review quota constraints, including, e.g., all conference review assignments and grant proposal assignments to a group of panelists.

Due to the lack of resources for evaluation, our conclusions are inevitably preliminary, thus an important future research direction is to further evaluate these algorithms with more data sets representing different application scenarios, ideally through applying the algorithms to review assignments for a real conference. Another interesting future research direction is to further extend our optimization formulation to include additional preferences such as the bids on papers entered by reviewers in a typical conference review system.

One limitation of our optimization formulation is that it maximizes only the coverage. Ideally, we would like to optimize both confidence and coverage, and further exploration of potentially better optimization algorithms would be a very interesting future work. Another limitation of our work is the lack of distinguishing possibly complementary expertise of multiple co-authors in both creation of the gold standard and automatic extraction of subtopics. Although this limitation is unlikely affecting much the hypotheses that we tested since the effectiveness of the proposed committee assignment algorithm is orthogonal to the optimization of topic inference, it would be interesting to further explore the use of author-topic model to model co-authors and infer topics more accurately [30].

References

1. Balog, K.; Azzopardi, L.; de Rijke, M. Formal models for expert finding in enterprise corpora. Proceedings of 29th International Conference on Research and Development in Information Retrieval; 2006. p. 43-50.
2. Basu, C.; Hirsh, H.; Cohen, W.; Nevill-Manning, C. Recommending papers by mining the web. Proceeding of 16th International Joint Conferences on Artificial Intelligence; 1999.
3. Benferhat S, Lang J. Conference paper assignment, In. Proceedings of International Journal of Intelligent Systems. 2001; 16:1183–1192.
4. Berretta, R.; Mendes, A.; Moscato, P. Integer programming models and algorithms for molecular classification of cancer from microarray data. Proceedings of the 28th Australasian conference on Computer Science ; 2005. p. 361-370.
5. Biswas, HK.; Hasan, MM. Using publications and domain knowledge to build research profiles: An application in automatic reviewer assignment. Proceedings of International Conference on Information and Communication Technology; 2007. p. 82-86.
6. Blei DM, Ng AY, Jordan MI. Latent dirichlet allocation. Proceedings of Journal of Machine Learning. 2003; 3:993–1022.
7. Campbell, C.; Maglio, P.; Cozzi, A.; Dom, B. Expertise identification using email communications. Proceedings of the 12th ACM CIKM International Conference on Information and Knowledge Management; 2003. p. 528-531.
8. Clarke J, Lapata M. Constraint-based sentence compression an integer programming approach. Proceedings of COLING/ACL. 2006:144–151.
9. Dempster AP, Laird NM, Rubin DB. Maximum likelihood from incomplete data via the em algorithm. Proceeding of Journal of the Royal Statistical Society. 1977; 39:1–38.
10. Dom, B.; Eiron, I.; Cozzi, A.; Zhang, Y. Graph-based ranking algorithms for e-mail expertise analysis. Proceedings of 22th ACM SIGMOD International Conference on Management of Data/ Principles of Database Systems; 2003. p. 42-48.
11. Dumais, S.; Nielsen, J. Automating the assignments of submitted manuscripts to reviewers. Proceedings of 15th International Conference on Research and Development in Information Retrieval; 1992. p. 233-244.
12. Fang, H.; Zhai, C. Probabilistic models for expert finding. Proceedings of 29th European Conference on Information Retrieval; 2007. p. 418-430.
13. Gkoulalas-Divanis, A.; Verykios, VS. An integer programming approach for frequent itemset hiding. Proceedings of ACM 15th Conference on Information and Knowledge Management; 2006. p. 748-757.
14. Hartvigsen D, Wei JC. The conference paper-reviewer assignment problem. Proceedings of Decision Sciences Journal. 1999; 30:865–876.
15. Hettich, S.; Pazzani, MJ. Mining for proposal reviewers: Lessons learned at the national science foundation. Proceeding of 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining; 2006. p. 862-871.
16. Hofmann, T. Probabilistic latent semantic indexing. Proceedings of 22nd International Conference on Research and Development in Information Retrieval; 1999. p. 50-57.
17. Jarvelin, K.; Kekalainen, J. Ir evaluation nethods for retrieving highly relevant documents. Proceedings of In Proceedings of the 23th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval; 2000. p. 41-48.

18. Karimzadehgan, M.; Zhai, C. Constrained multi-aspect expertise matching for committee review assignment. Proceeding of the 18th ACM conference on Information and Knowledge Management; 2009. p. 1697-1700.

19. Karimzadehgan, M.; Zhai, C.; Belford, G. Multi-aspect expertise matching for review assignment. Proceedings of ACM 17th Conference on Information and Knowledge Management; 2008. p. 1113-1122.

20. Korte, B.; Vygen, J. Combinatorial Optimization Theory and Algorithms. Springer; 2006.

21. Li, W.; McCallum, A. Pachinko allocation: Dag-structured mixture models of topic correlations. Proceedings of 23th International Conference on Machine Learning; 2006. p. 577-584.

22. Macdonald, C.; Ounis, I. Voting for candidates: Adapting data fusion techniques for an expert search task. Proceedings of the 15th ACM Conference on Information and Knowledge Management; 2006. p. 387-396.

23. Maybury M, D'Amore R, House D. Expert finding for collaborative virtual environments. Proceedings of Communications of the ACM. 2001; 44(12):55–56.

24. Menon S, Sarkar S, Mukherjee S. Maximizing accuracy of shared databases when concealing sensitive patterns. Proceedings of Information System Research. 2005; 16(3):256–270.

25. Merelo-Guervos, JJ.; Castillo-Valdivieso, P. Conference paper assignment using a combined greedy/evolutionary algorithm; Springer; 2004. p. 602-611.

26. Mimno, D.; McCallum, A. Expertise modeling for matching papers with reviewers. Proceeding of 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining; 2007. p. 500-509.

27. Papadimitriou, CH.; Steiglitz, K. Combinatorial Optimization, Algorithms and Complexity. Prentice Hall; 1982.

28. Petkova, D.; Croft, WB. Hierarchical language models for expert finding in enterprise corpora. Proceedings of the 18th IEEE International Conference on Tools with Artificial Intelligence; 2006. p. 599-608.

29. Rodriguez, MA.; Bollen, J. An algorithm to determine peer-reviewers. Proceedings of ACM 17th Conference on Information and Knowledge Management; 2008. p. 319-328.

30. Rosen-Zvi, M.; Griffiths, T.; Smyth, P.; Steyvers, M. The author-topic model for authors and documents. Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence; 2004. p. 487-494.

31. Roth, D.; tau Yih, W. Integer linear programming inference for conditional random fields. Proceedings of the 22nd International Conference on Machine Learning; 2005. p. 736-743.

32. Salkin, HM.; Mathur, K. Foundations of Integer Programming. Elsevier; 1989.

33. Sihn, W.; Heeren, F. Xpertfinder - expert finding within specified subject areas through analysis of e-mail communication. Proceedings of Euromedia; 2001. p. 279-283.

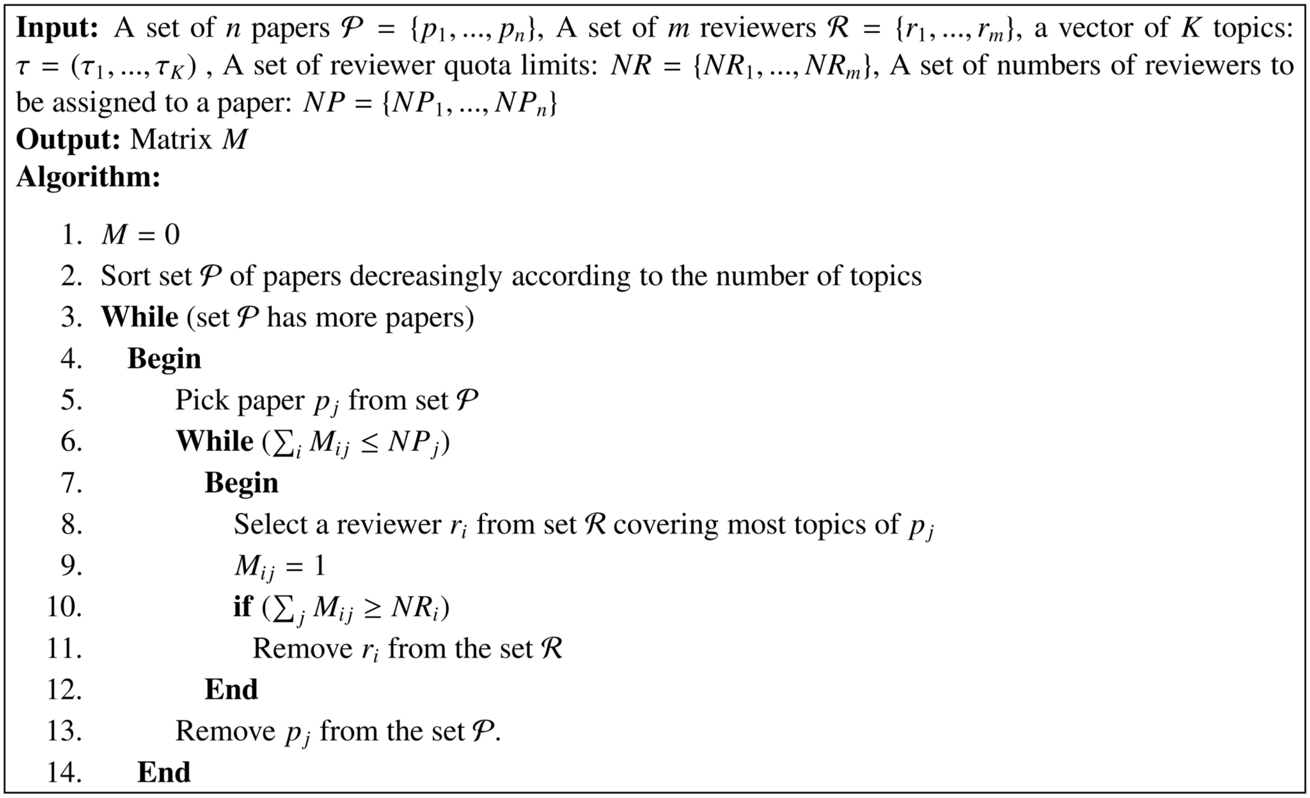
34. Sun YH, Ma J, Fan ZP, Wang J. A hybrid knowledge and model approach for reviewer assignment. Proceedings of Expert Systems with Applications: An International Journal. 2008; 34:817–824.

35. Taylor, CJ. Technical Reports. University of Pennsylvania; 2008. On the optimal assignment of conference papers to reviewers; p. 1-5.

36. Wang, X.; McCallum, A. Topics over time: a non-markov continuous-time model of topical trends. Proceedings of The 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining; 2006. p. 424-433.

37. Wilcoxon F. Individual comparisons by ranking methods. In Biometrics Bulletin. 1945; 1:80–83.

38. Zhai, C.; Cohen, W.; Lafferty, J. Beyond independent relevance: Methods and evaluation metrics for subtopic retrieval. Proceedings of 26th International Conference on Research and Development in Information Retrieval; 2003. p. 10-17.



Input: A set of n papers $\mathcal{P} = \{p_1, ..., p_n\}$, A set of m reviewers $\mathcal{R} = \{r_1, ..., r_m\}$, a vector of K topics: $\tau = (\tau_1, ..., \tau_K)$, A set of reviewer quota limits: $NR = \{NR_1, ..., NR_m\}$, A set of numbers of reviewers to be assigned to a paper: $NP = \{NP_1, ..., NP_n\}$

Output: Matrix M

Algorithm:

```
$M = 0$
Sort set $\mathcal{P}$ of papers decreasingly according to the number of topics
While (set $\mathcal{P}$ has more papers)
  Begin
    Pick paper $p_j$ from set $\mathcal{P}$
    While ($\sum_i M_{ij} \leq NP_j$)
      Begin
        Select a reviewer $r_i$ from set $\mathcal{R}$ covering most topics of $p_j$
        $M_{ij} = 1$
        if ($\sum_j M_{ij} \geq NR_i$)
          Remove $r_i$ from the set $\mathcal{R}$
      End
    Remove $p_j$ from the set $\mathcal{P}$.
  End
```

Figure 1.

A greedy Algorithm

$$Maximize\left(\sum_{j=1}^{n}\sum_{k=1}^{K} t_{jk}\right)$$

Subject to constraints:

C1: $\forall i \in [1,m],\ j \in [1,n],\ M_{ij} \in \{0,1\}$

C2: $\forall j \in [1,n], k \in [1,K],\ t_{jk} \in \left\{0,\ldots,NP_j\right\}$

C3: $\forall j \in [1,n],\ \sum_{i=1}^{m} M_{ij} = NP_j$

C4: $\forall i \in [1,m],\ \sum_{j=1}^{n} M_{ij} \leq NR_i$

C5: $\forall j \in [1,n], k \in [1,K]\ P_{jk}t_{jk} \leq \sum_{l=1}^{m} R_{lk}M_{lj}$

Figure 2.
ILP formulation

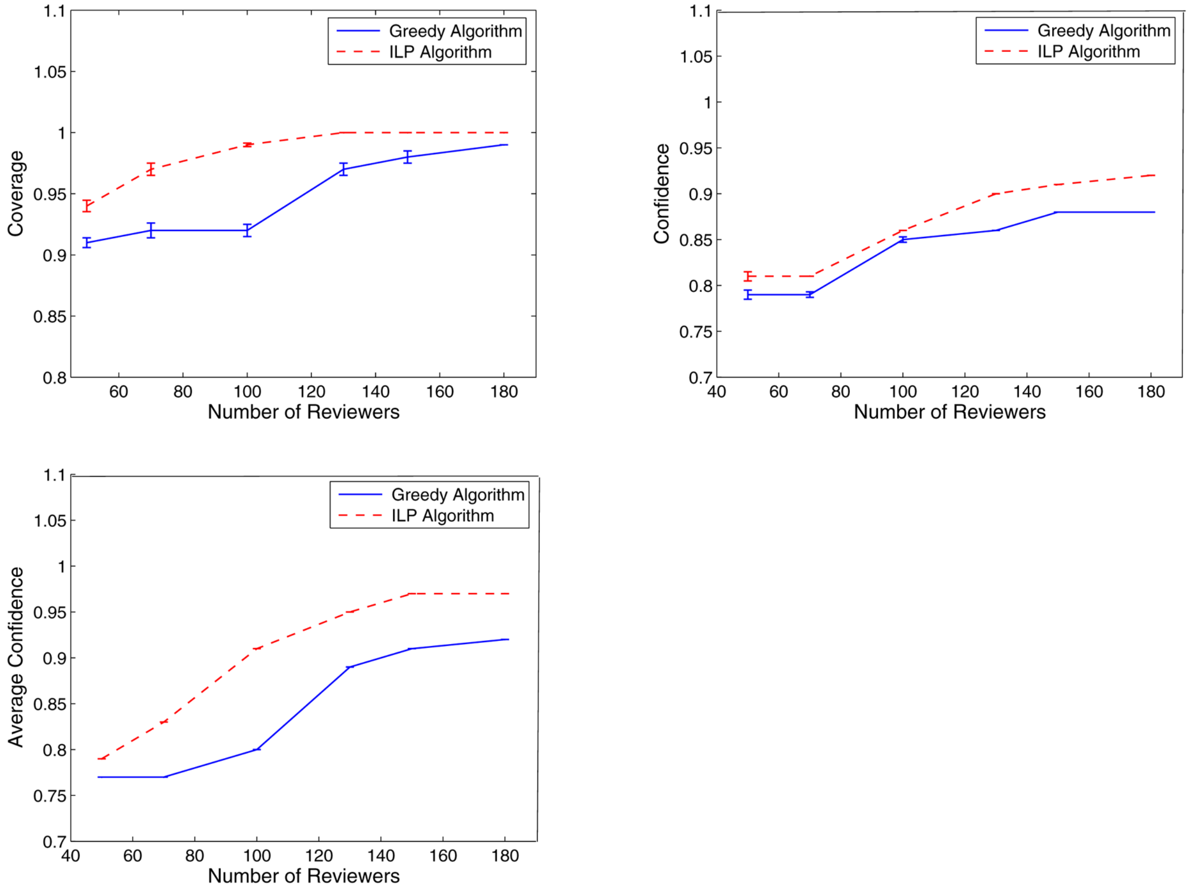


Figure 3.

Comparison of ILP and greedy algorithms according to all evaluation measures. Each paper gets 3 reviewers and each reviewer gets up to 5 papers. The number of papers is 73 and we vary the number of reviewers.

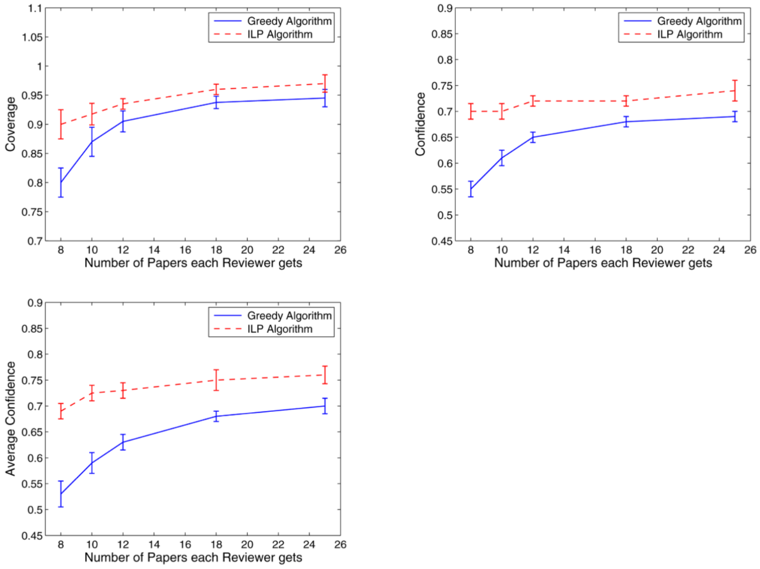


Figure 4.
Comparison of ILP and greedy algorithms according to all evaluation measures. The number of papers is 73, and the number of reviewers is 30. Each paper gets 3 reviewers and we vary the number of papers that each reviewer can review.

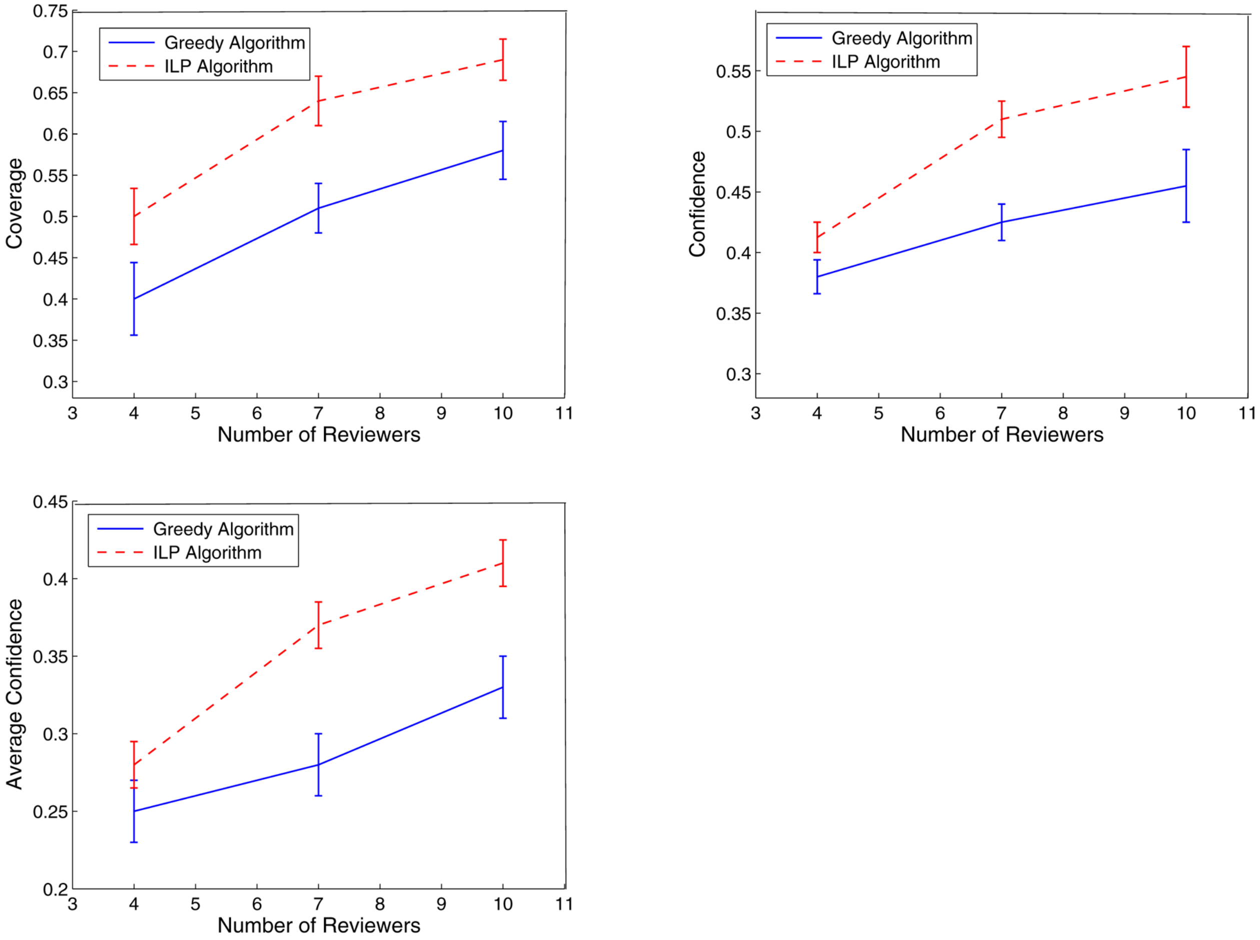


Figure 5.

Comparison of ILP and greedy algorithms according to all evaluation measures. Vary the number of reviewers available for reviewing and set the number of papers assigned to each reviewer accordingly (assuming equal review load for each reviewer).

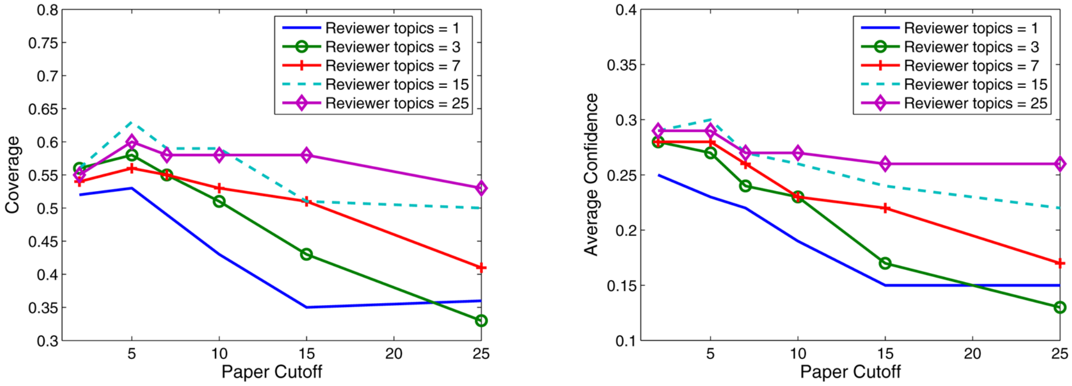


Figure 6.

Papers' topics and reviewers' topics are learned with PLSA model. Each paper gets 3 reviewers and each reviewer gets 5 papers. Performance of the ILP algorithm to Coverage measure (left) and to Average Confidence (right).

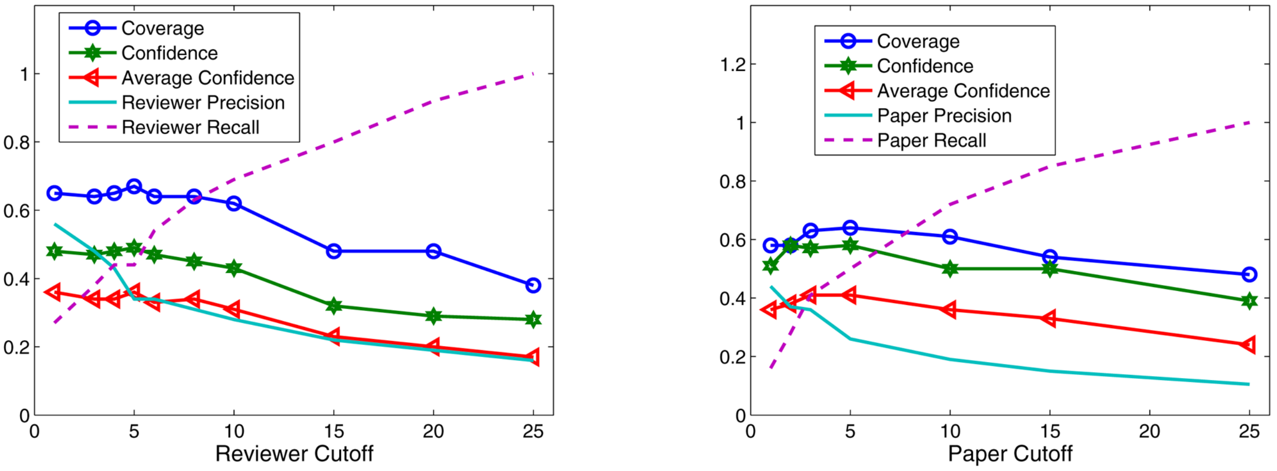


Figure 7.

Number of papers is 73. Each paper gets 3 reviewers and each reviewer gets 5 papers. Papers' topics are from gold standard data, reviewers' topics are learned with PLSA model and different cutoff thresholds are applied to vary topic assignments to reviewers (left). Papers' topics are learned with PLSA model and different cutoff thresholds are applied to vary the assignments of topics to papers, reviewers' topics are from gold standard data (right).

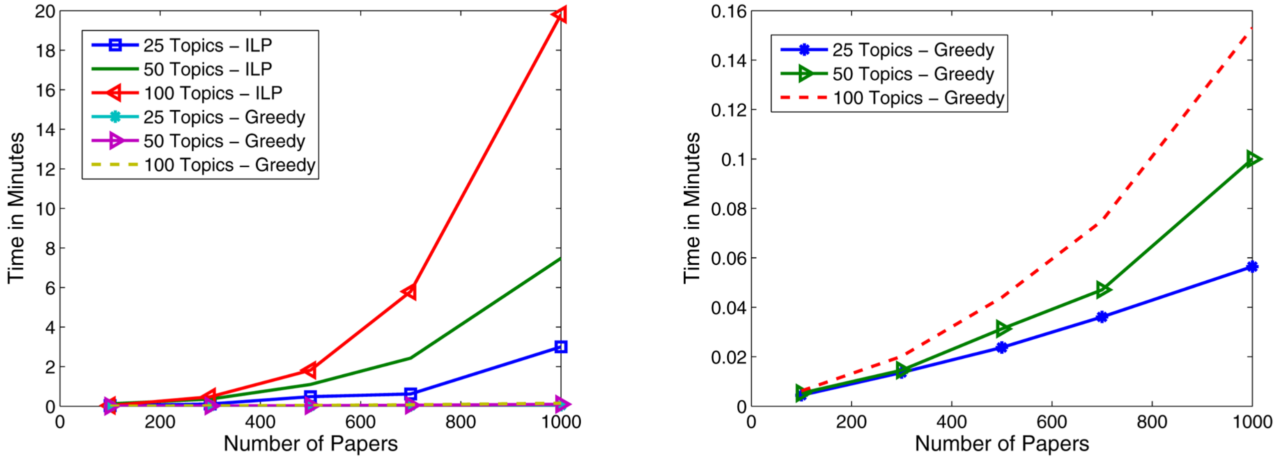


Figure 8.

Runtime for both greedy and ILP algorithm with different number of topics, i.e., 25, 50 and 100 (left) and runtime for greedy algorithm (right).

Table 1

Comparing the results of applying the ILP algorithm to the PLSA directly (No Cutoff) with the best results gained from pruning (Best Cutoff, reviewer topics are 15 and paper topics are 5).

Method	**Coverage**	**Avg Confidence**
No Cutoff	0.53	0.26
Best Cutoff	0.63 *	0.3 *

*shows statistically significant results at the level of 0.05.