Weighted Coverage based Reviewer Assignment

Ngai Meng Kou *1 Leong Hou U *2 Nikos Mamoulis #3 Zhiguo Gong *4

*Department of Computer and Information Science, University of Macau, Macau SAR

#Department of Computer Science, University of Hong Kong, Hong Kong SAR

1yb27406@umac.mo 2ryanlhu@umac.mo 3nikos@cs.hku.hk 4fstzgg@umac.mo

ABSTRACT

Peer reviewing is a standard process for assessing the quality of submissions at academic conferences and journals. A very important task in this process is the assignment of reviewers to papers. However, achieving an appropriate assignment is not easy, because all reviewers should have similar load and the subjects of the assigned papers should be consistent with the reviewers' expertise. In this paper, we propose a generalized framework for fair reviewer assignment. We first extract the domain knowledge from the reviewers' published papers and model this knowledge as a set of topics. Then, we perform a *group assignment* of reviewers to papers, which is a generalization of the classic Reviewer Assignment Problem (RAP), considering the relevance of the papers to topics as weights. We study a special case of the problem, where reviewers are to be found for just one paper (Journal Assignment Problem) and propose an exact algorithm which is fast in practice, as opposed to brute-force solutions. For the general case of having to assign multiple papers, which is too hard to be solved exactly, we propose a greedy algorithm that achieves a 1/2-approximation ratio compared to the exact solution. This is a great improvement compared to the 1/3-approximation solution proposed in previous work for the simpler coverage-based reviewer assignment problem, where there are no weights on topics. We theoretically prove the approximation bound of our solution and experimentally show that it is superior to the current state-of-the-art.

Keywords

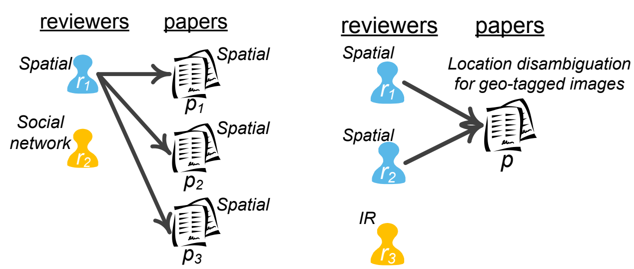
Paper Reviewer Assignment; Group Coverage; Stage Deepening Greedy

1. INTRODUCTION

Peer reviewing is a widely accepted process for assessing submitted papers to academic venues. One of the most challenging tasks in this process is to perform the assignment of reviewers to papers in a way that would maximize the quality of reviews. This problem is known as the *Reviewer Assignment Problem* (RAP) [4, 6, 7, 10, 14, 15, 18, 19, 21, 23, 29]. In order to improve the quality of the assignment, existing conference management systems (e.g., Conference Management Toolkit[1] and EasyChair[2]) ask reviewers to bid on their preferred papers; the assignment is then conducted based on the bids, as a classic matching problem [20]. Still, there are well-known drawbacks of this approach. For example, reviewers could be too lazy to go through the complete list of paper titles and abstracts. Alternatively, the reviewer assignment can be performed *automatically*, based on similarity models between the papers and the reviewers.

SIGMOD'15, May 31–June 4, 2015, Melbourne, Victoria, Australia.

http://dx.doi.org/10.1145/2723372.2723727.



(a) Retrieval-based (b) Assignment-based

Figure 1: Drawbacks of previous RAP definitions

1.1 Previous Work

RAP was first regarded as a retrieval problem by Dumais and Nielsen [10]. The retrieval-based RAPs [4, 10, 19, 23] attempt to retrieve qualified reviewers for each paper based on their similarity to the paper, where similarity is measured using topic discovery models [10, 18, 21, 23], Bayesian probabilistic matrix factorization and linear regression [6], interval fuzzy ontologies [29], root mean square error of bids [7], or vector space models [4]. This category of approaches produce imbalanced assignments, where some reviewers may receive too many papers to review. An example is given in Figure 1(a), where reviewer r_1, whose research interest is mainly in spatial databases, receives 3 related papers to review but another reviewer (r_2) receives no paper by this assignment. Another line of work [6, 7, 14, 18, 21, 27, 29] regard RAP as an assignment problem subject to a maximum workload per reviewer and the number of required reviews per paper. These assignment-based RAPs aim at

[1]http://cmt.research.microsoft.com/cmt/

[2]http://www.easychair.org/

finding the best assignment (i.e., an assignment objective is optimized) subject to the aforementioned constraints. The quality of each assignment pair is individually considered; however, it may turn out that an interdisciplinary paper is reviewed by a group of reviewers with too narrow expertise. For instance, assuming that each paper is supposed to be assigned to 2 reviewers in Figure 1(b), the review quality of p may improve if one of the reviewers (e.g., r_1 or r_2) is replaced by r_3.

The drawback of assigning reviewers independently was first stated in [15], where papers and reviewers are modeled as term vectors and the expertise of a reviewer group on a term is the sum of term values of the reviewers in the group. The assignment objective is to minimize the difference between the papers and their assigned groups for each term in their vectors. A hill-climbing algorithm is proposed in [15] to find a local optimal value for this problem, without a quality guarantee; in addition, [15] does not consider how to automatically construct reviewer groups. Recently, Long et al. [22] formulated and solved RAP as a Set-coverage Group-based Reviewer Assignment Problem (SGRAP). The intuition is that the quality of the reviews on a paper should be measured based on the entire group of reviewers assigned to it. In other words, a paper is well-reviewed only if the assigned reviewers have the expertise to cover every single topic of the paper. Long et al. [22] transform the expertise of reviewers and the content of papers into sets of topics and assess the quality of assigning a group of reviewers $\{r_i, \cdots, r_j\}$ to paper p by the *set coverage ratio*, i.e., $|\mathbb{T}_{r_i} \cup \cdots \cup \mathbb{T}_{r_j} \cap \mathbb{T}_p|/|\mathbb{T}_p|$, where $\mathbb{T}_{r_i}$ is the set of topics in the expertise of r_i and $\mathbb{T}_p$ is the set of p's topics. This assessment well considers every topic of a paper in the assignment process. However, all topics of a paper are assumed to have identical importance, which is not always true in the real world. For instance, a paper could be related to many topics, but only one of them is the main subject of the paper.

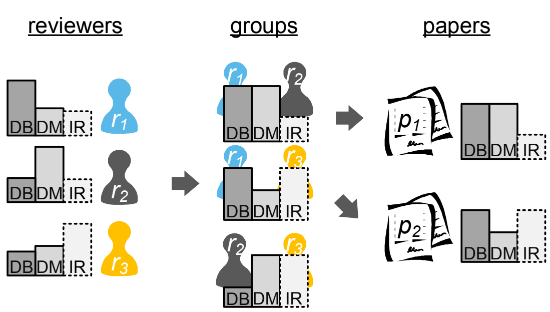


Figure 2: Weighted-coverage GRAP

1.2 Contributions

In this work, we revise the quality measure for a group of reviewers assigned to a paper from *set coverage* to *weighted coverage*. We model the expertise of reviewers and the content of papers as T-dimensional *topic vectors* (instead of topic sets). The quality of an assignment pair is estimated by considering the T-dimensional vectors of reviewers and the paper. In Figure 2, for example, the best group to review paper p_1 is $\{r_1, r_2\}$ because the topic vectors of these two reviewers, when taken together, best match the topic vector of p_1 (i.e., high relevance to DB and DM, lower relevance to IR). Our *weighted coverage* (to be defined in detail in Section 2) well addresses the topic equilibrium problem mentioned above, where the topics are no longer viewed as equally important. Based on the above, we define and solve RAP as a Weighted-coverage Group-based Reviewer Assignment Problem (WGRAP). Our contributions are as follows.

Introduction and generality of WGRAP. We introduce WGRAP as an appropriate definition of RAP that considers (i) the load balancing of reviewers to papers and (ii) the coverage of the topics of the assigned papers by the reviewers' expertise, proportionally to the importance of the paper topics. We show that all previous RAP definitions (retrieval-based RAP, assignment-based RAP, SGRAP) are special cases of WGRAP.

Journal Reviewer Assignment (JRA). As a case of WGRAP, we first study the problem of assigning a single journal paper p to a set of δ_p reviewers, chosen from a pool R of candidates. We propose an exact Branch-and-Bound Algorithm (BBA), which finds the best reviewer group based on a well-designed execution order. BBA finds the result much faster compared to a brute force method that examines all reviewer combinations, an integer linear programming approach, and a commercial constraint programming solver.

An approximation algorithm for WGRAP. The general WGRAP, where multiple papers are to be assigned to multiple reviewers, is an NP-hard problem; it is a generalization of SGRAP, the hardness of which was shown in Long et al. [22]. Since finding the exact solution for WGRAP is too expensive, we propose a polynomial-time approximation algorithm. Compared to the greedy algorithm of [22], which has 1/3 approximation ratio, our algorithm improves the approximation ratio significantly (to at least 1/2) and it is applicable to both WGRAP and SGRAP. The main idea of our approach is to divide the assignment into stages, such that exactly one reviewer is assigned to each paper at each stage (solved in PTIME as a linear assignment problem). More importantly, our approximation guarantee holds for any submodular objective function (we investigate the applicability of alternative objective functions in Appendix B).

Stochastic refinement of a WGRAP solution. We propose a stochastic process for refining the assignment computed by our approximation algorithm. In a nutshell, our approach first builds a probability model that captures the suitability of reviewers to be assigned to papers. Then, it iteratively attempts to swap assignment pairs, following the probability model, until the process converges to a solution that cannot be further improved.

Experimental evaluation. We conduct an experimental evaluation using real data from DBLP. Our results show our approximation algorithm paired with the stochastic refinement postprocessing finds an assignment of much better quality compared to previous work, within reasonable time.

The rest of this paper is organized as follows. Section 2 provides the necessary definitions, formulates WGRAP, and analyzes its relationship to previous RAP definitions. Section 3 includes our solution to JRA. In Section 4, we study the general case of WGRAP, by reviewing the greedy algorithm of [22], proposing our solution, analyzing its approximation ratio, and presenting our stochastic refinement process. Our experiments are presented in Section 5. Finally, Section 6 concludes with a discussion about future work.

2. PRELIMINARIES AND DEFINITIONS

In this section, we first define fundamental concepts, including topics, reviewer groups, and weighted coverage. Subsequently, we give the formal definition of our WGRAP and discuss how we can specialize WGRAP into three RAP definitions studied before.

2.1 Topic Coverage and Assignment Quality

As introduced in Section 1, we assume that there are T research topics (i.e., subjects) and that the expertise of reviewers and the content of papers are modeled as T-dimensional topic vectors, denoted as $\overrightarrow{r_i} = (\overrightarrow{r_i}[1], \cdots, \overrightarrow{r_i}[T])$ (for reviewer r_i) and $\overrightarrow{p_j} = (\overrightarrow{p_j}[1], \cdots, \overrightarrow{p_j}[T])$ (for paper p_j), respectively. For instance, $\overrightarrow{r_i}[1]$ refers to the relevance (i.e., expertise) of reviewer r_i to the first topic, while $\overrightarrow{p_j}[1]$ is the relevance of paper p_j to the first topic. For simplicity, we sometimes refer to $\overrightarrow{r_i}[t]$ (resp. $\overrightarrow{p_j}[t]$) as the *weight* of reviewer r_i (resp. paper p_j) on topic t. Given the publications record of r_i and the abstract of p_j, $\overrightarrow{r_i}$ and $\overrightarrow{p_j}$ can be extracted by topic modeling, e.g., Latent Dirichlet Allocation (LDA) [5]. Even though topic extraction is not the main focus of our work, it remains an important component of our framework. Thus, we will discuss its details in Section 2.4.

The topic vector of a reviewer $\overrightarrow{r}$ can be viewed as the confidence of the reviewer over different topics. Intuitively, if topic t of a paper p is well covered by an assigned reviewer r (i.e., $\overrightarrow{r}[t] \geq \overrightarrow{p}[t]$), then it is safe to assume that r is fully confident to review p on topic t. Specifically, we can measure how well a reviewer r *covers* topic t by $\min\{\overrightarrow{r}[t], \overrightarrow{p}[t]\}$. This quantity, also encapsulates the relative importance (i.e., *weight*) of t's coverage compared to other topics. For example, if a topic t' is more relevant to paper p than t is (i.e., $\overrightarrow{p}[t'] > \overrightarrow{p}[t]$), then the coverage of reviewer r on t' should be more important than the coverage of r on t. Thereby, the quality of assigning r to review p can be calculated by the *weighted coverage* of the two vectors as follows.

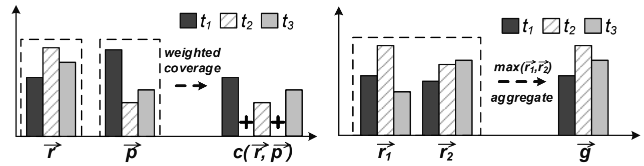
DEFINITION 1 (WEIGHTED COVERAGE SCORE $c(\overrightarrow{r}, \overrightarrow{p})$). *The quality of assigning $\overrightarrow{r}$ to review $\overrightarrow{p}$ is the summation of the coverage weights over different topics.*

$$c(\overrightarrow{r}, \overrightarrow{p}) = \frac{\sum_{t=1}^{T} \min\{\overrightarrow{r}[t], \overrightarrow{p}[t]\}}{\sum_{t=1}^{T} \overrightarrow{p}[t]} \quad (1)$$

The denominator $\sum_{t=1}^{T} \overrightarrow{p}[t]$ normalizes $c(\overrightarrow{r}, \overrightarrow{p})$ to take values from 0 to 1.[3] As an example, Figure 3(a) shows the quality of assigning r to review p, calculated based on the coverage weights over three topics. Besides Definition 1, we study alternative scoring functions for the quality of a reviewer-paper assignment in Appendix B.

We now discuss how we measure the quality of assigning a *group of reviewers* $g = \{r_i, \cdots, r_j\}$ to a specific paper p.

[3]The normalization is not necessary if we already have $\sum_{t=1}^{T} \overrightarrow{p}[t] = 1$ for each paper p, i.e., the relevance of each paper to the topics is already a normalized vector. Besides, we can also assume that the reviewer vectors are normalized, i.e., $\sum_{t=1}^{T} \overrightarrow{r}[t] = 1$ for each reviewer r. Otherwise, we might allow one reviewer r_1 to dominate another reviewer r_2 with respect to expertise (i.e., r_1's expertise in each topic is greater than the corresponding expertise of r_2), which means that r_1 could be considered more qualified than r_2 in general. Regardless the situation, we keep the denominator in Eq. 1 to make the definition of weighted coverage general.



(a) Weighted coverage (b) Group topic vector

Figure 3: Weighted coverage and group topic vector

Given the topic vectors of the group (i.e., $\{\overrightarrow{r_i}, \cdots, \overrightarrow{r_j}\}$), the expertise of the reviewer group $\overrightarrow{g}$ is a vector which stores for every topic t the *maximum expertise of any reviewer in the group* in t. The rationale is that the reviewer who is the most expert in t will have the highest confidence in reviewing the aspects of the paper related to t and her opinion will most probably dominate the opinions on these aspects from other reviewers in the group. Formally,

DEFINITION 2 (EXPERTISE OF A REVIEWER GROUP). *Given* $g = \{r_i, \cdots, r_j\}$, $\overrightarrow{g} = (\max_{r \in g} \overrightarrow{r}[1], \cdots, \max_{r \in g} \overrightarrow{r}[T])$.

As an example, Figure 3(b) shows a reviewer group $g = \{r_1, r_2\}$. The expertise of g is $\overrightarrow{g} = (\overrightarrow{r_1}[1], \overrightarrow{r_1}[2], \overrightarrow{r_2}[3])$. The quality of assigning group g to review a paper p can now also be conveniently calculated using Equation 1; i.e., as $c(\overrightarrow{g}, \overrightarrow{p})$, after computing $\overrightarrow{g}$.

2.2 Weighted-coverage Group-based Reviewer Assignment Problem (WGRAP)

Without loss of generality, we assume that there are P papers (i.e., $\mathbb{P} = \{p_1, \cdots, p_P\}$) and R reviewers (i.e., $\mathbb{R} = \{r_1, \cdots, r_R\}$), where each paper should be reviewed by δ_p reviewers (group size constraint) and each reviewer should be given at most δ_r papers (reviewer workload). In addition, we assume that there are enough reviewers for the given papers, i.e., $R \cdot \delta_r \geq P \cdot \delta_p$.

The goal of our *Weighted-coverage Group-based Reviewer Assignment Problem* (WGRAP) is to find the best assignment $\mathbb{A} \subseteq \mathbb{P} \times \mathbb{R}$ such that the *objective* (i.e., the summation of the weighted coverage scores per topic) is maximized subject to the workload constraints. For the ease of discussion, we use $\mathbb{A}[x]$ to denote the assignment pair(s) of x. For instance, if $\mathbb{A} = \{(r_1, p_1), (r_2, p_1)\}$, then $\mathbb{A}[r_1] = \{(r_1, p_1)\}$ and $\mathbb{A}[p_1] = \mathbb{A}$. WGRAP is formally defined as follows:

DEFINITION 3 (WGRAP). *Given a set of papers $\mathbb{P}$ and a set of reviewers $\mathbb{R}$, WGRAP finds an assignment $\mathbb{A} \subseteq \mathbb{P} \times \mathbb{R}$, such that*

$$\begin{aligned} \max \quad & \sum_{p \in \mathbb{P}} c(\overrightarrow{g}, \overrightarrow{p}) \\ \textit{where} \quad & g = \{r | (r,p) \in \mathbb{A}[p]\} \\ \textit{s.t.} \quad & |\mathbb{A}[r]| \leq \delta_r \qquad \forall r \in \mathbb{R} \\ & \textit{(reviewer workload)} \\ & |\mathbb{A}[p]| = \delta_p \qquad \forall p \in \mathbb{P} \\ & \textit{(group size constraint)} \end{aligned}$$

In WGRAP, the assignment quality of each paper is calculated by the weighted-coverage of the assigned reviewers (cf. Definition 1); i.e., we want the assigned group of reviewers per paper to cover as much as possible every single topic of the paper. The quality of an assignment $\mathbb{A}$ is measured by the objective function of Definition 3, i.e., $\sum_{p\in\mathbb{P}} c(\overrightarrow{g}, \overrightarrow{p})$, where $g = \{r|(r,p) \in \mathbb{A}[p]\}$. We sometimes refer to this quality measure as the *coverage score* of assignment $\mathbb{A}$, simply denoted by $c(\mathbb{A})$. Table 1 summarizes the frequently used notations in this paper.

Symbol	Description	Symbol	Description
p	paper	r	reviewer
g	reviewer group	$\mathbb{T}$	set of topics
$\mathbb{P}$	set of papers	$\mathbb{R}$	set of reviewers
$\mathbb{A}$	assignment	$\mathbb{O}$	optimal assignment
δ_p	group size constraint	δ_r	reviewer workload

Table 1: Notations

2.3 Relationship to other RAPs

In this section, we briefly introduce three popular types of RAPs and discuss their relationship to our WGRAP.

Retrieval-based RAP (RRAP). This type of RAP is first proposed by Dumais and Nielsen [10]. Their model firstly suggests the top-δ_r most relevant papers to each reviewer and then each reviewer picks $\delta_r/2$ papers (out of δ_r) to review. As discussed in Section 1, RRAP may end up in imbalanced assignments, where some paper may be assigned to no reviewers. This is due to the lack of the group size constraint (i.e., the number of reviewers of a paper, δ_p) in RRAP. The formal definition of RRAP is shown as follows.

Definition 4 (RRAP). *Given $\mathbb{P}$ and $\mathbb{R}$, RRAP finds an assignment $\mathbb{A} \subseteq \mathbb{P} \times \mathbb{R}$ such that*

$$\begin{aligned} \max \quad & \sum_{p\in\mathbb{P}} \sum_{r\in\mathbb{A}[p]} c(\overrightarrow{r}, \overrightarrow{p}) \\ s.t. \quad & |\mathbb{A}[r]| = \delta_r \qquad \forall r \in \mathbb{R} \end{aligned}$$

WGRAP can be reduced to RRAP by two steps, (1) removing the group size constraint and (2) revising the objective function, i.e., making $\sum_{r\in\mathbb{A}[p]} c(\overrightarrow{r}, \overrightarrow{p})$ equal to $c(\overrightarrow{g}, \overrightarrow{p})$. For the second step, we can simply extend the topic vectors from T dimensions to $R \cdot T$ dimensions such that: (i) the original topic vectors of papers are repeated R times, (ii) for the i-th reviewer, the i-th T dimensions include its original topic vector and the others are 0. This reduction can be done in polynomial time.

Assignment-based RAP (ARAP). The second type of RAP addresses the workload problem: every paper is reviewed by a certain amount of reviewers [18]. The formal definition of ARAP is as follows.

Definition 5 (ARAP). *Given $\mathbb{P}$ and $\mathbb{R}$, ARAP finds an assignment $\mathbb{A} \subseteq \mathbb{P} \times \mathbb{R}$ such that*

$$\begin{aligned} \max \quad & \sum_{p\in\mathbb{P}} \sum_{r\in\mathbb{A}[p]} c(\overrightarrow{r}, \overrightarrow{p}) \\ s.t. \quad & |\mathbb{A}[r]| \le \delta_r \qquad \forall r \in \mathbb{R} \\ & |\mathbb{A}[p]| = \delta_p \qquad \forall p \in \mathbb{P} \end{aligned}$$

Since ARAP already considers reviewer size constraint, WGRAP can be reduced to ARAP by simply revising the objective function (similarly to the reduction to RRAP).

Set-coverage GRAP (SGRAP). We omit the formal definition of SGRAP [22], since it is identical to our WGRAP except for how the coverage function $c(\overrightarrow{g}, \overrightarrow{p})$ is defined for a group of reviewers assigned to a paper. WGRAP can easily be reduced to SGRAP if we transform each topic set $\mathbb{T}$ into a binary T-dimensional topic vector. The i-th value of a topic vector (e.g., $\overrightarrow{p}[i]$) is set to 1 if the i-th topic exists in the topic set (e.g., $\mathbb{T}_p$). Otherwise, it is set to 0. This conversion makes the coverage functions of SGRAP and WGRAP become identical, i.e.,

$$c(\mathbb{T}_g, \mathbb{T}_p) = \frac{|\mathbb{T}_g \cap \mathbb{T}_p|}{|\mathbb{T}_p|} = \frac{\sum_{t=1}^{T} \min\{\overrightarrow{g}[t], \overrightarrow{p}[t]\}}{\sum_{t=1}^{T} \overrightarrow{p}[t]} = c(\overrightarrow{g}, \overrightarrow{p})$$

where $\overrightarrow{g}[t], \overrightarrow{p}[t] \in \{0, 1\}$. Thus, SGRAP is a special case of WGRAP (where all scores are integral); this means that our solutions for WGRAP can be applied to SGRAP as well.

Summary. To the best of our knowledge, WGRAP is the first RAP formulation that assesses the reviewer assignment quality by a group-based objective function under weighted coverage of topics. Our evaluation approach well addresses certain drawbacks of previous approaches, including the load balancing problem (using the *group size constraint*), the diversity of reviewer groups (using a *group-based objective function*), and the topic equilibrium problem (using *weighted-coverage*).

	RRAP	ARAP	SGRAP	WGRAP
Constraint				
Group size	X	√	√	√
Objective				
Group-based	X	X	√	√
Obj. function	Weight	Weight	Set	Weight

Table 2: Comparison of different RAPs

2.4 Topic Vector Extraction

We now briefly introduce how we can extract the topic vectors (i.e., $\overrightarrow{p}$ and $\overrightarrow{r}$) from the corresponding papers and reviewers. A naïve method is to collect the vectors manually; e.g., collecting paper topics from keywords specified by the authors or asking the reviewers to declare their expertise by some check boxes. However, such information is not always available. As an example, in a journal review process, the expertise of potential reviewers is typically not declared by them in advance, or the pool of potential reviewers is dynamic. A more reasonable approach would be to extract the topic vectors of potential reviewers automatically from their publication records.

In this work, we first use the Author-Topic Model (ATM) of [25] to extract the topic set $\mathbb{T}$ and the topic vector of reviewers $\{\overrightarrow{r_1}, \cdots, \overrightarrow{r_R}\}$ based on their publication records. The topic vector of papers $\{\overrightarrow{p_1}, \cdots, \overrightarrow{p_P}\}$ is estimated by Expectation-Maximization (EM) [30] based on $\mathbb{T}$. For completeness, we show the details of our adapted ATM in Appendix A.

3. JOURNAL REVIEWER ASSIGNMENT

In this section, we study a special case of WGRAP, where only a single paper has to be reviewed (i.e., $\mathbb{P} = \{p\}$). This case finds practical application in the scenario of *Journal Reviewer Assignment* (JRA), where the editor is looking for δ_p qualified reviewers for a single submission. Thus, we can ignore the reviewer workload δ_r, since each assigned reviewer assesses the given paper just once. Formally, Definition 3 is reduced to JRA as follows:

Definition 6 (JRA). *Given a journal paper p and a set of reviewers $\mathbb{R}$, the journal reviewer assignment finds an assignment $\mathbb{A} \subseteq \{p\} \times \mathbb{R}$ such that*

$$\begin{aligned} &\max && c(\overrightarrow{g}, \overrightarrow{p}) \\ &where && g = \{r|(r,p) \in \mathbb{A}[p]\} \\ &s.t. && |\mathbb{A}[p]| = \delta_p \end{aligned}$$

The best reviewer group is one of the δ_p-combinations out of R reviewers and there are $C^R_{\delta_p}$ such combinations. The following lemma proves the NP-hardness of JRA.

Lemma 1. *JRA is an NP-hard problem.*

Proof. We will show that another NP-hard problem, i.e., the *maximum coverage problem* [11], can be reduced to an instance of JRA in polynomial time. Given a collection of sets $\mathbb{S} = \{s_1, s_2, ...\}$, where each set s_i is a subset of a domain set $\mathbb{D} = \{e_1, \cdots, e_n\}$, and a number k, the maximum coverage problem finds a subset $\mathbb{S}' \subseteq \mathbb{S}$ of sets such that $|\mathbb{S}'| = k$ and the number of covered elements from $\mathbb{D}$, i.e., $|\cup_{s_i \in \mathbb{S}'} s_i|$, is maximized. A special case of JRA is that the target paper p is only relevant to n topics and each topic has the same importance, i.e., $\overrightarrow{p}[t] \in \{0, 1/n\}$ and $\sum_{1 \leq t \leq T} \overrightarrow{p}[t] = 1$. We transform each subset $s_i \in \mathbb{S}$ to a T-dimensional vector $\overrightarrow{s_i}$, where $\overrightarrow{s_i}[e] = 1/n$ if element e is in s_i, otherwise $\overrightarrow{s_i}[e] = 0$. We also set $k = \delta_p$ (i.e., the group size constraint). All these transformations take $O(|\mathbb{S}|n)$ time. The problem now becomes to find a subset $\mathbb{S}' \subseteq \mathbb{S}$ of k collections such that $c(\overrightarrow{g}, \overrightarrow{p})$, where $g = \{s_i | s_i \in \mathbb{S}'\}$ is maximized. This problem is equivalent to the maximum coverage problem as every newly covered (missing) topic of $\mathbb{S}'$ increases (decreases) by exactly $1/n$ of the total coverage score. □

Even though JRA is NP-hard (it considers $C^R_{\delta_p}$ reviewer combinations), we can still solve it within acceptable time if R and δ_p are not very large. For example, in practice $\delta_p = 3$ and R is in the order of a few hundreds. To the best of our knowledge, Brute Force Search (BFS) (i.e., enumerating every possible reviewer group) and Integer Linear Programming (ILP) [24] can be used to compute JRA exactly. However, these two solutions do not scale well. BFS is very sensitive to R and δ_p, because it examines every possible reviewer group and ILP suffers from floating-point precision issues, when there are too many constrained functions and variables (these are proportional to R and δ_p). We develop a novel algorithm, Branch-and-Bound Algorithm (BBA), which finds the best reviewer group based on a well-designed execution order. A promising result is that BBA finds the exact solution almost in real time in practical cases. For example, BBA finds the best set of 5 reviewers out of 200 candidates within 2.2 seconds on a commodity machine while BFS and ILP take 5.1 hours and 45.6 minutes, respectively, to solve the same problem.

BBA operates on the search space of JRA (cf. Figure 4), that can be viewed as a tree structure. Each *non-root* node represents a reviewer r_i and it has up to $R-1$ children nodes (i.e., all reviewers except r_i). The depth of the tree is determined by the group size constraint δ_p; the nodes along a path from the root to a leaf indicate one possible reviewer group. For instance, path $root \rightarrow r_1 \rightarrow r_3$ indicates reviewer group $\{r_1, r_3\}$.

To search for the best reviewer group, we apply the classic backtracking paradigm which partitions the search process

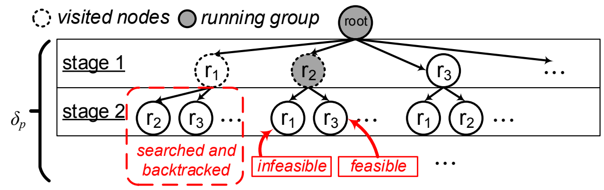


Figure 4: Search space of JRA

into δ_p stages. When the search moves from stage s to stage $s+1$, a *feasible* reviewer (see Definition 7) is inserted into the running reviewer group g. The feasibility condition ensures that every group combination is only examined at most once in the entire search process. When search reaches the last stage (i.e., stage δ_p), the coverage score of g is calculated and the best-so-far result is updated, if applicable. Search backtracks to the previous stage if there are no more feasible reviewers at the current stage.

Definition 7 (Feasibility of reviewer). *Given a running group g and the visited information, a reviewer r is feasible only if r is not yet visited along the execution path (i.e., along the first stage to the running stage).*

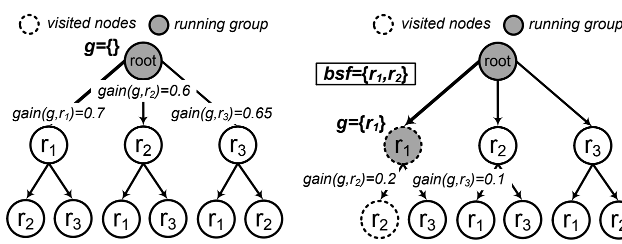
In Figure 4, when examining the branches under r_2 (at stage 1), r_1 (at stage 2) is infeasible since r_1 has been visited at stage 1 already. In other words, every possible reviewer group containing r_1 has been considered as soon as r_1 has been selected at stage 1, therefore r_1 should be omitted from the subsequent processes. As another example, r_3 is still feasible since the visited information of the branches under r_1 has been reset when it backtracks to stage 1.

	t_1	t_2	t_3
$\overrightarrow{p}$	0.35	0.45	0.2
$\overrightarrow{r_1}$	0.15	0.75	0.1
$\overrightarrow{r_2}$	0.75	0.15	0.1
$\overrightarrow{r_3}$	0.1	0.35	0.55

(a) Topic vectors

SL_1	SL_2	SL_3
$0.75(r_2)$	$0.75(r_1)$	$0.55(r_3)$
$0.15(r_1)$	$0.35(r_3)$	$0.1(r_1)$
$0.1(r_3)$	$0.15(r_2)$	$0.1(r_2)$

(b) T-sorted lists, SL_i



(c) Picking r_1 at stage 1 (d) Early termination at stage 2

Figure 5: Searching for the best reviewer group

To efficiently locate the best reviewer group, we need to prioritize the traversal order (*branching*) and terminate search branches that cannot lead to a better solution as early as possible (*bounding*). Given the reviewer vectors $\{\overrightarrow{r_1}, \cdots, \overrightarrow{r_R}\}$, we prepare T sorted lists where the t-th sorted list SL_t keeps the t-th values of the reviewer vectors in an descending order. An example of three reviewers and their corresponding sorted lists is shown in Figure 5(a) and 5(b), respectively.

In the following, we show how to prioritize and bound the search process by accessing the T sorted lists back-and-forth. At each stage s, we construct a set of T running cursors, $\Pi^s = \{\pi_1^s, \cdots, \pi_T^s\}$ and every cursor always points at a feasible reviewer. Initially, every cursor (e.g., π_i^1) points at the beginning of the corresponding sorted list (e.g., SL_i).

Branching. Intuitively, a reviewer r is a good candidate for the running group g if adding r into g has large *marginal gain*, which is defined as follows:

DEFINITION 8 (MARGINAL GAIN, $gain(g, r, p)$). *Given a running reviewer group g and a reviewer candidate r, the marginal gain of adding r into g is calculated by*

$$gain(g, r, p) = c(\overrightarrow{g \cup \{r\}}, \overrightarrow{p}) - c(\overrightarrow{g}, \overrightarrow{p}) \tag{2}$$

We use the running cursors to locate the best yet feasible reviewer to be considered in the branching process. More specifically, we add r into g only if r has the maximum marginal gain among the reviewers pointed by the cursors at current stage s. After having added r into g, we move *forward* every affected cursor (i.e., pointing to r) to the next *feasible reviewer* and continue to the next stage. The initial positions of the next stage cursors Π^{s+1} are cloned from Π^s since every reviewer group is necessarily examined once (cf. the feasibility condition in Definition 7). In our running example, r_1 is the first picked reviewer since its marginal gain (i.e., $gain(g, r_1, p) = c(\overrightarrow{r_1}, \overrightarrow{p}) = 0.7$) is the maximum among all three cursors (see Figure 5(c)). r_1 is marked as visited at stage 1 and we move forward the affected cursor (i.e., π_2^1) to the next feasible reviewer (i.e., $0.35(r_3)$).

Bounding. We use the running cursors to estimate the upper bound of the running group g as follows:

$$UB(g, \Pi^s) = c(\overrightarrow{ub}, \overrightarrow{p}) \tag{3}$$

where $\overrightarrow{ub}[i] = \max(\overrightarrow{g}[t_i], \pi_i^s), \forall_{1 \le i \le T}$. Equation 3 is the upper bound of g since the cursors always move forward so that every cursor always point to the best yet feasible reviewer. If the upper bound of g is no longer promising (i.e., smaller than the score of the best reviewer group found so far), then we backtrack to the previous stage and reset the visited information at the current stage.

In our running example (cf. Figure 5(d)), assume that we already have examined a reviewer group $bsf = \{r_1, r_2\}$ (e.g., $c(\overrightarrow{bsf}, \overrightarrow{p}) = 0.9$) and bsf is our best result so far. The running cursors at stage 2 point to $0.1(r_3), 0.35(r_3)$, and $0.55(r_3)$, respectively. Before examining another feasible reviewer (e.g., r_3) at stage 2, we estimate the upper bound of g by Equation 3 where $\overrightarrow{ub} = \{0.15, 0.75, 0.55\}$ and $UB(g, \Pi^2) = 0.8$. Since the upper bound of the running group $g(=\{r_1\})$ is not better than the best-so-far result, we backtrack to stage 1 and reset the visited information at stage 2 (e.g., changing the dashed line of r_2 back to solid line at stage 2).

Algorithm 1 shows a pseudocode of BBA for clarity. Search is terminated when there is no feasible cursor at stage 1 (cf. Line 5 of Algorithm 1). BBA fully exploits the sorted lists to boost the brand-and-bound process by maintaining the running cursors properly. Even though BBA does not improve the worst-case computational cost of JRA, its branching prioritization (by the marginal gain estimation) and its early termination (by the upper bound estimation) significantly reduce the computational cost in practice. The space overhead of BBA is mainly due to the sorted lists that consume $O(R \cdot T)$ space (i.e., negligible space for modern commodity machines). Note that BBA can easily be adapted to return the top-k reviewer sets (i.e., by replacing bsf by a heap structure that keeps the k best-so-far reviewer groups).

Algorithm 1 Branch-and-Bound Algorithm (BBA)

Input: paper p, reviewer set $\mathbb{R}$, reviewer group size δ_p
Output: result bsf
Algorithm $bba(p, \mathbb{R}, \delta_p)$
1: construct T sorted lists
2: initialize the running cursor set for 1st stage, Π^1
3: set running group $g \leftarrow \emptyset$ and *best-so-far* group $bsf \leftarrow \emptyset$
4: set running stage $s \leftarrow 1$
5: **while** $\exists_{\pi_i^1 \in \Pi^1} \pi_i^1 \neq nil$ **do**
6: $\quad r \leftarrow \text{argmax}_{r \in \{\pi_1^s, \cdots, \pi_T^s\}} gain(g, r, p)$ ▷ by Eq. 2
7: $\quad$ estimate $UB(g, \Pi^s)$ ▷ by Eq. 3
8: $\quad$ **if** $r = \emptyset$ or $UB(g, \Pi^s) \le c(\overrightarrow{bsf}, \overrightarrow{p})$ **then** ▷ bounding
9: $\quad\quad$ reset visited information at stage s
10: $\quad\quad$ backtrack to previous stage (i.e., $s \leftarrow s - 1$)
11: $\quad\quad$ **goto** line 5
12: $\quad g \leftarrow g \cup \{r\}$ and mark r as visited at stage s ▷ branching
13: $\quad$ **if** $|g| = \delta_p$ **then** ▷ a complete assignment is found
14: $\quad\quad$ **if** $c(\overrightarrow{g}, \overrightarrow{p}) > c(\overrightarrow{bsf}, \overrightarrow{p})$ **then** $bsf \leftarrow g$
15: $\quad\quad g \leftarrow g \backslash \{r\}$ ▷ backtracking
16: $\quad$ **else**
17: $\quad\quad$ **for** $\pi_i^s \in \{\pi_1^s, \cdots, \pi_T^s\}$ **do**
18: $\quad\quad\quad$ move π_i^s *forward* when π_i^s is infeasible
19: $\quad\quad$ set $\Pi^{s+1} \leftarrow \Pi^s$
20: $\quad\quad$ move to next stage (i.e., $s \leftarrow s + 1$)
21: **return** bsf

4. CONFERENCE REVIEWER ASSIGNMENT

In this section, we study the general WGRAP (Definition 3), which applies in a scenario of *Conference Reviewer Assignment* (CRA), where P received papers must be assigned to R program committee members. The objective is to find an assignment $\mathbb{A} \subseteq \mathbb{P} \times \mathbb{R}$ such that the assignment maximizes the topic coverage scores subject to the reviewer group and workload constraints. Compared to JRA, the number of papers increases from 1 to P, therefore the search space increases from $C_{\delta_p}^R$ to $(C_{\delta_p}^R)^P$. Such an increase makes finding the exact solution infeasible even for small problems (e.g., when $R = 100$ and $P = 100$). Thus we develop a polynomial-time approximation algorithm for CRA.

4.1 Greedy Algorithm

We first investigate the application of the approximation algorithm [22], which was proposed for SGRAP, to WGRAP. This greedy algorithm finds a solution for SGRAP with 1/3 approximation ratio. The assignment $\mathbb{A}$ is constructed incrementally; at each iteration, the algorithm picks the feasible pair of reviewer and paper $(r, p)_{best}$ which has the largest marginal gain when added to the current $\mathbb{A}$ among all feasible pairs. The iterative process terminates only when all papers are fully assigned, i.e., $\mathbb{A}[p] = \delta_p, \forall p \in \mathbb{P}$. Formally:

$$\begin{aligned} (r, p)_{best} &= \underset{(r,p) \in \mathbb{F}}{\text{argmax}} \; gain(\mathbb{A}[p], r, p), \\ \text{where } \mathbb{F} &= \{(r, p) \in \mathbb{R} \times \mathbb{P} \mid (r, p) \notin \mathbb{A} \wedge \\ & \quad |\mathbb{A}[r]| < \delta_r \wedge |\mathbb{A}[p]| < \delta_p\} \end{aligned} \tag{4}$$

$\mathbb{F}$ indicates the set of feasible reviewer-paper pairs and $\mathbb{A}[p]$ (used in the $gain(\cdot)$ function) represents the running group of reviewers for p (i.e., reviewers already assigned to p).

The greedy algorithm needs $P \cdot \delta_p$ iterations[4] and each iteration takes $O(P \cdot R)$ time to locate a pair (i.e., evaluating the *gain* score). We can easily reduce the cost of this step to logarithmic time if we keep the feasible pairs into a heap that organizes them in descending order of their marginal gains. This can be done because the gain function is monotonically decreasing with the size of $\mathbb{A}$. Thus, the time complexity of Greedy is $O(P\delta_p \log(PR))$.

The greedy algorithm is shown to provide an assignment with 1/3 approximation ratio for SGRAP [22]. Since SGRAP is a special case of WGRAP (cf. Section 2.3), the greedy algorithm finds at worst an 1/3-approximation solution for WGRAP.[5] The greedy algorithm simply splits the assignment process into $P \cdot \delta_p$ iterations and disregards how the selections between iterations are correlated. In the next section, we introduce a more sophisticated method that performs only δ_p iterations and significantly increases the approximation ratio up to $1 - 1/e$.

4.2 Stage Deepening Greedy Algorithm

Our intuition is to reduce the number of iterations and assign multiple pairs at each iteration. This way, we can improve the quality of the resulting assignment because we have higher flexibility in which pairs are added to $\mathbb{A}$ at each iteration. One way to apply our idea is to find the best reviewer group (i.e., δ_p reviewers) for any paper at each stage (i.e., iteration), which reduces the number of iterations to P (from $P \cdot \delta_p$). However, finding the best reviewer group for a specific paper (i.e., the JRA problem) is already NP-hard, therefore such a solution would not be a polynomial-time solution. Thus, our goal is to partition the stages such that (i) each stage requires polynomial time and (ii) the approximation ratio is guaranteed. We first discuss how we perform the assignment in δ_p stages and then analyze the approximation ratio of our method in Section 4.3.

To satisfy condition (i), we construct a sub-problem which assigns exactly one reviewer to every paper at each stage; in this way, at each stage we solve a PTIME linear assignment problem. To fulfil condition (ii), we confine the reviewer workload at each stage, i.e., every reviewer is assigned at most $\lceil \delta_r/\delta_p \rceil$ papers at each stage. The necessity of this confinement will be discussed in Section 4.3; the practical benefit is that every reviewer enters into the tail stages of the assignment process, as shown in the following example. Consider a WGRAP with 3 reviewers and 3 papers with topic vectors as in the tables below. Let $\delta_p = 2$ and $\delta_r = 2$. If we greedily assign r_1 to 2 papers (i.e., $c(\overrightarrow{r_1}, \overrightarrow{p_2}) = 0.6$ and $c(\overrightarrow{r_1}, \overrightarrow{p_3}) = 0.6$) at the first stage, then no reviewer (other than r_1) can cover t_3 at the second stage. On the other hand, the overall assignment score will be improved if we reserve one workload of r_1 for the second stage.

	t_1	t_2	t_3
$\overrightarrow{r_1}$	0.1	0.5	0.4
$\overrightarrow{r_2}$	1	0	0
$\overrightarrow{r_3}$	0	1	0

	t_1	t_2	t_3
$\overrightarrow{p_1}$	0.6	0	0.4
$\overrightarrow{p_2}$	0.5	0.5	0
$\overrightarrow{p_3}$	0.5	0.5	0

[4]if there are not enough reviewers, it needs $R \cdot \delta_r$ iterations.
[5]The proof in [22] is also applicable to WGRAP, as a greedy algorithm for the problem of maximizing a submodular function (i.e., $c(\cdot)$) over a 2-system (i.e., feasible set of assignments) achieves a 1/3-factor approximation [12].

DEFINITION 9 (STAGE-WGRAP). *Given the running stage s and previous stage results $\mathbb{A}_1, \cdots, \mathbb{A}_{s-1}$, Stage-WGRAP finds an assignment $\mathbb{A}_s$ such that*

$$\begin{aligned} \max \quad & \sum_{p \in \mathbb{P}} c(\overrightarrow{g}, \overrightarrow{p}) \\ \textit{where} \quad & g = \{r | (r,p) \in \mathbb{A}_1[p] \cup \cdots \cup \mathbb{A}_s[p]\} \\ \textit{s.t.} \quad & |\mathbb{A}_s[r]| \le \lceil \delta_r/\delta_p \rceil \qquad \forall r \in \mathbb{R} \\ & \textit{(reviewer workload)} \\ & |\mathbb{A}_s[p]| = 1 \qquad \forall p \in \mathbb{P} \\ & \textit{(group size constraint)} \end{aligned}$$

We denote the discussed sub-problem as Stage-WGRAP and define it in Definition 9. When $s = 1$, this sub-problem is clearly a linear assignment problem (cf. Definition 5), since each group includes only a single reviewer at the first stage.

LEMMA 2. *Stage-WGRAP can be computed incrementally from stage 1 to s in polynomial time.*

PROOF. So far we know that $\mathbb{A}_1$ can be computed in polynomial time. Given $\mathbb{A}_1$, the objective function at stage 2 can be rewritten as:

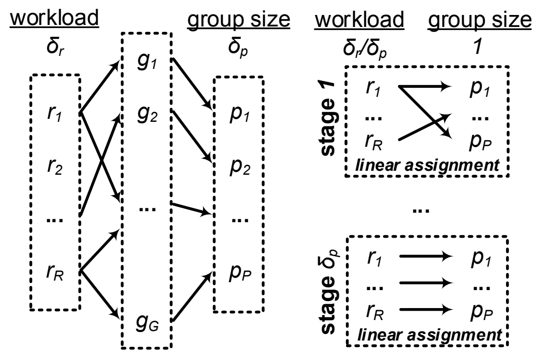
$$\begin{aligned} \sum_{p \in \mathbb{P}} c(\overrightarrow{g}, \overrightarrow{p}) &= \sum_{p \in \mathbb{P}} c(\overrightarrow{\mathbb{A}_1[p] \cup \mathbb{A}_2[p]}, \overrightarrow{p}) \\ &= \sum_{p \in \mathbb{P}} c(\overrightarrow{\mathbb{A}_1[p]}, \overrightarrow{p}) + \sum_{p \in \mathbb{P}} gain(\mathbb{A}_1[p], \mathbb{A}_2[p], p) \end{aligned}$$

As $c(\overrightarrow{\mathbb{A}_1[p]}, \overrightarrow{p})$ is fixed at stage 1 already, our objective is to find $\mathbb{A}_2$ such that the marginal gain $\sum_{p \in \mathbb{P}} gain(\mathbb{A}_1[p], \mathbb{A}_2[p], p)$ of the assignment is maximized. This is obviously another linear assignment problem. By simple induction, we can compute Stage-WGRAP incrementally from stage 1 to s in polynomial time, i.e.,

$$\begin{aligned} \sum_{p \in \mathbb{P}} c(\overrightarrow{g}, \overrightarrow{p}) &= \sum_{p \in \mathbb{P}} c(\overrightarrow{\mathbb{A}_1[p] \cup \cdots \cup \mathbb{A}_s[p]}, \overrightarrow{p}) \\ &= \sum_{p \in \mathbb{P}} c(\overrightarrow{\mathbb{A}_1[p] \cup \cdots \cup \mathbb{A}_{s-1}[p]}, \overrightarrow{p}) \\ &\quad + \sum_{p \in \mathbb{P}} gain(\mathbb{A}_1[p] \cup \cdots \cup \mathbb{A}_{s-1}[p], \mathbb{A}_s[p], p) \end{aligned} \tag{5}$$

□

Figure 6 illustrates WGRAP and Stage-WGRAP. Algorithm 2 is a pseudocode for our Stage Deepening Greedy Algorithm (SDGA). At each stage, we can apply a classic linear assignment algorithm (e.g., Hungarian algorithm [20], Minimum-cost flow assignment [3]) to compute the assignment in polynomial time. For instance, the Hungarian algorithm takes $O((\max\{P, R\})^3)$ to compute a linear assignment; therefore, the time complexity of SDGA using the Hungarian algorithm is $O(\delta_p(\max\{P, R\})^3)$ and the space requirement is $O((\max\{P, R\})^2)$. Both time and space complexities are acceptable (for commodity machines) as P and R are in the order of a few hundreds (or thousands) in academic conferences.



(a) WGRAP (b) Stage-WGRAP

Figure 6: WGRAP and Stage-WGRAP

Algorithm 2 Stage Deepening Greedy Algorithm

Input: paper set $\mathbb{P}$, reviewer set $\mathbb{R}$, workload constraint δ_r, group size constraint δ_p
Output: assignment $\mathbb{A}$
Algorithm $SDGA(\mathbb{P}, \mathbb{R}, \delta_r, \delta_p)$
for $s \leftarrow 1$ to δ_p **do**
    $\mathbb{A}_s \leftarrow$ *Stage-WGRAP*$(\{\mathbb{A}_1, \cdots, \mathbb{A}_{s-1}\}, \mathbb{P}, \mathbb{R})$ ▷ by Eq. 5
return $\mathbb{A}_1 \cup \cdots \cup \mathbb{A}_{\delta_p}$

4.3 Approximation Ratio of SDGA

In this section, we show that SDGA provides a solution of WGRAP with an approximation ratio $(1 - 1/e)$ (when δ_r is divisible by δ_p) and $1/2$ (otherwise).

4.3.1 Integral cases: δ_r/δ_p is an integer

Optimal assignment, $\mathbb{O}$. We first introduce some notations that are used throughout this section. For any WGRAP, we denote by $\mathbb{O}$ its optimal assignment. In the following, we show how we can split $\mathbb{O}$ into δ_p isolated sets, $\{\mathbb{O}_1, \cdots, \mathbb{O}_{\delta_p}\}$, such that $\mathbb{O} = \mathbb{O}_1 \cup \cdots \cup \mathbb{O}_{\delta_p}$, $\mathbb{O}_i \cap \mathbb{O}_j = \emptyset, \forall 1 \le i < j \le \delta_p$, and for every stage s,

$$|\mathbb{O}_s[r]| \le \lceil \delta_r/\delta_p \rceil, \forall r \in \mathbb{R}, \text{ and } |\mathbb{O}_s[p]| = 1, \forall p \in \mathbb{P} \quad (6)$$

This partitioning can be done in $O(|\mathbb{O}|^2)$ time in a nested loops fashion. First, we arbitrarily divide the reviewer-paper pairs in $\mathbb{O}$ into disjoint sets $\mathbb{O}_1, \cdots, \mathbb{O}_{\delta_p}$. Then, the outer loop scans every assignment pair from $\mathbb{O}_1, \cdots, \mathbb{O}_{\delta_p}$; if the current pair (e.g., in $\mathbb{O}_s$) violates the constraints of Equation 6, an inner loop is run to the partitions (e.g., $\mathbb{O}_{s+1}, \cdots, \mathbb{O}_{\delta_p}, \mathbb{O}_1, \cdots, \mathbb{O}_{s-1}$) to swap the pair with one that is valid.

Stage marginal gain and assignment gap. For the ease of our discussion, we use $c(\mathbb{O})$ to denote the *coverage score* (i.e., quality) of an assignment $\mathbb{O}$, i.e., $c(\mathbb{O}) = \sum_{p \in \mathbb{P}} c(\overrightarrow{g}, \overrightarrow{p})$ where $g = \{r | (r, p) \in \mathbb{O}[p]\}$ and we use $\mathbb{O}_{1 \cdots s}$ to denote the union set of $\mathbb{O}_1 \cup \cdots \cup \mathbb{O}_s$.

We define the stage marginal gain of the SDGA assignment at the running stage s as

$$\Delta(\mathbb{A}, s) = c(\mathbb{A}_{1 \cdots s}) - c(\mathbb{A}_{1 \cdots s-1}) \quad (7)$$

and the *assignment gap* (i.e., the difference) between the optimal score and the running SDGA score (from stage 1 to $s-1$) as

$$gap(\mathbb{O}, \mathbb{A}, s) = c(\mathbb{O}) - c(\mathbb{A}_{1 \cdots s-1}) \quad (8)$$

To show the approximation ratio, we first show that the marginal gain of SDGA at stage s is always larger than the $1/\delta_p$-assignment gap. This is a commonly used step in showing the approximation ratio of maximum coverage problems [11].

LEMMA 3. $\Delta(\mathbb{A}, s) \ge gap(\mathbb{O}, \mathbb{A}, s)/\delta_p$

PROOF. As the coverage score is always positive, we begin with the fact that the coverage score becomes higher when we have a larger assignment, i.e.,

$$\begin{aligned}
c(\mathbb{O}) \le & c(\mathbb{A}_{1 \cdots s-1} \cup \mathbb{O}) \\
gap(\mathbb{O}, \mathbb{A}, s) \le & c(\mathbb{A}_{1 \cdots s-1} \cup \mathbb{O}_{1 \cdots \delta_p}) - c(\mathbb{A}_{1 \cdots s-1}) \\
= & c(\mathbb{A}_{1 \cdots s-1} \cup \mathbb{O}_1) - c(\mathbb{A}_{1 \cdots s-1}) + \\
& c(\mathbb{A}_{1 \cdots s-1} \cup \mathbb{O}_{1 \cdots 2}) - c(\mathbb{A}_{1 \cdots s-1} \cup \mathbb{O}_1) + \\
& \cdots + \\
& c(\mathbb{A}_{1 \cdots s-1} \cup \mathbb{O}_{1 \cdots \delta_p}) - c(\mathbb{A}_{1 \cdots s-1} \cup \mathbb{O}_{1 \cdots \delta_p - 1})
\end{aligned}$$

where each term $c(\mathbb{A}_{1 \cdots s-1} \cup \mathbb{O}_{1 \cdots i}) - c(\mathbb{A}_{1 \cdots s-1} \cup \mathbb{O}_{1 \cdots i-1})$ indicates the marginal gain of adding $\mathbb{O}_i$ into the set $\mathbb{A}_{1 \cdots s-1} \cup \mathbb{O}_{1 \cdots i-1}$. Accordingly, there must exist an $\mathbb{O}_i$ such that the marginal gain is larger than $gap(\mathbb{O}, \mathbb{A}, s)/\delta_p$, i.e.,

$$c(\mathbb{A}_{1 \cdots s-1} \cup \mathbb{O}_{1 \cdots i}) - c(\mathbb{A}_{1 \cdots s-1} \cup \mathbb{O}_{1 \cdots i-1}) \ge gap(\mathbb{O}, \mathbb{A}, s)/\delta_p$$

Since $c(\cdot)$ is a submodular function[6], $c(\mathbb{A}_{1 \cdots s-1} \cup \mathbb{O}_i) - c(\mathbb{A}_{1 \cdots s-1}) \ge c(\mathbb{A}_{1 \cdots s-1} \cup \mathbb{O}_{1 \cdots i}) - c(\mathbb{A}_{1 \cdots s-1} \cup \mathbb{O}_{1 \cdots i-1})$. Thus,

$$c(\mathbb{A}_{1 \cdots s-1} \cup \mathbb{O}_i) - c(\mathbb{A}_{1 \cdots s-1}) \ge gap(\mathbb{O}, \mathbb{A}, s)/\delta_p$$

What remains is to show that $c(\mathbb{A}_{1 \cdots s-1} \cup \mathbb{A}_s) \ge c(\mathbb{A}_{1 \cdots s-1} \cup \mathbb{O}_i)$. This inequality is true since (i) we always find the optimal assignment at each stage independently and (ii) the stages are equally partitioned (i.e., workload=δ_r/δ_p) so that $\mathbb{A}_s$ and $\mathbb{O}_i$ always share the same sets of reviewers and papers. □

THEOREM 1. *SDGA is a $(1 - 1/e)$-approximation algorithm for WGRAP if δ_r is divisible by δ_p.*

PROOF.

$$\begin{aligned}
& c(\mathbb{O}) - c(\mathbb{A}_{1 \cdots \delta_p}) \\
= & c(\mathbb{O}) - c(\mathbb{A}_{1 \cdots \delta_p - 1}) - \Delta(\mathbb{A}, \delta_p) && \text{(by Eq. 7)} \\
= & gap(\mathbb{O}, \mathbb{A}, \delta_p) - \Delta(\mathbb{A}, \delta_p) && \text{(by Eq. 8)} \\
\le & gap(\mathbb{O}, \mathbb{A}, \delta_p)(1 - 1/\delta_p) && \text{(by Lemma 3)} \\
\le & gap(\mathbb{O}, \mathbb{A}, \delta_p - 1)(1 - 1/\delta_p)^2 && \text{(Eq.7, 8 and Lemma 3)} \\
\le & \cdots \\
\le & (1 - 1/\delta_p)^{\delta_p} c(\mathbb{O}) && \text{(after } \delta_p \text{ iterations)} \\
\le & (1/e) c(\mathbb{O}) \\
\therefore\ & c(\mathbb{A}_{1 \cdots \delta_p}) \ge (1 - 1/e) c(\mathbb{O})
\end{aligned}$$

□

4.3.2 General cases: δ_r/δ_p is a real number

$\mathbb{O}$ cannot be split into equal sets $\{\mathbb{O}_1, \cdots, \mathbb{O}_{\delta_p}\}$ if δ_r is not divisible by δ_p. As SDGA computes the approximate assignment stage by stage, for the first $\delta_p - 1$ stages, the claim of Lemma 3 is satisfied. Thereby, we can derive the

[6] For the sake of readability, this is shown in Appendix B.

following inequality from the proof of Theorem 1, by simply ignoring the contribution at the last stage:

$$c(\mathbb{O}) - c(\mathbb{A}_{1\cdots\delta_p}) \leq (1 - 1/\delta_p)^{\delta_p - 1} c(\mathbb{O})$$

Thus, the approximation ratio is at least $1 - (1 - 1/\delta_p)^{\delta_p - 1}$. We conclude the approximation ratio of SDGA in Theorem 2.

THEOREM 2. *SDGA is a 1/2-approximation algorithm for WGRAP.*

PROOF. Trivial. $1 - (1 - 1/\delta_p)^{\delta_p - 1}$ is monotonically increasing to δ_p. When $\delta_p \in \mathcal{I}$ and $\delta_p \geq 2$, the smallest value of this function is 1/2. □

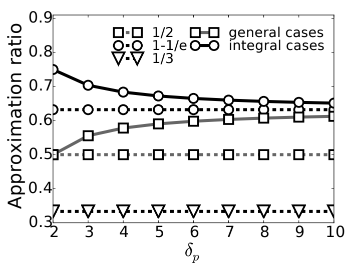


Figure 7: The effect of δ_p

As the approximation ratio of both integral (i.e., $1 - (1 - 1/\delta_p)^{\delta_p}$) and general cases (i.e., $1 - (1 - 1/\delta_p)^{\delta_p - 1}$) is sensitive to δ_p, we plot the ratio according to the value of δ_p in Figure 7 for clarity. In general cases, the approximation ratio increases to 5/9 for $\delta_p = 3$ (e.g., a typical setting in major database conferences) and to 0.5904 for $\delta_p = 5$ (e.g., a setting used in some IR conferences).

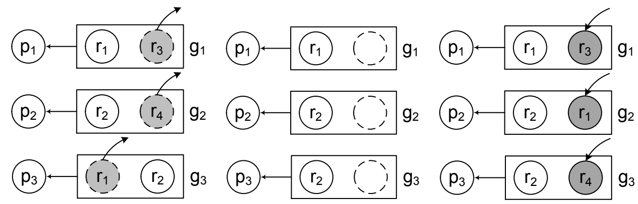
Supporting Conflicts of Interest (COIs). Our approximation algorithm can also seamlessly support COIs with the same approximation guarantee. The reason is that for any pair (r, p), which is a COI, the pair must not appear any stage of $\mathbb{O}_i$ and $\mathbb{A}_i$. Thus, $\mathbb{O}_i$ and $\mathbb{A}_i$ still share the same workload of reviewers and papers.

4.4 Stochastic Refinement

The approximate result $\mathbb{A}$ of SDGA can be further improved by postprocessing. *Local search* [17] is a widely used postprocessing technique for approximation algorithms. In our context, local search attempts to swap two or multiple assignment pairs such that the assignment quality (i.e., coverage score) gets improved. The search terminates when the improvement becomes not obvious (i.e., convergence) or the search time exceeds a pre-defined threshold.

However, a simple application of local search is not expected to be effective for WGRAP since the search space is huge, i.e., $O((C^R_{\delta_p})^P)$, even for small data instances. In such a huge space, local search is not likely to find a good result by only considering local information, e.g., greedily swapping two pairs (or a short chain of pairs) to improve the coverage score. In this section, we propose an effective refinement method, which swaps pairs based on a stochastic process.

Assume that the probability $\mathcal{P}(r|p)$ of the assignment pair (r, p) to be in the optimal assignment $\mathbb{O}$ can be estimated



(a) $\mathcal{P}$-based removal (b) Available slots (c) stage-WGRAP

Figure 8: An illustration of stochastic refinement

(how to estimate this will be discussed shortly). Our idea is to try the replacement of some pairs from $\mathbb{A}$ according to this probability. More specifically, we attempt to remove pairs (r, p) by a stochastic process (i.e., (r, p) has high chance to be removed when $\mathcal{P}(r|p)$ is low) and then add back pairs for papers with less reviewers than δ_p. In order to make this process systematic, we only remove one reviewer from each paper; this way, the result can be completed by a linear assignment (i.e., similarly to the process at the last stage of SDGA). The basic idea of the stochastic refinement is illustrated in Figure 8. In our approach, we iteratively apply stochastic refinement on the result of SDGA, until the process converges: if the assignment result does not improve in the last ω refinement rounds, then we assume that further refinement is not likely to improve the quality, therefore we terminate the process.

What remains is to estimate the probability $\mathcal{P}(r|p)$ of reviewer r correctly being assigned to paper p. A simple approach would be to consider the probabilities of all reviewers identical, i.e., $\mathcal{P}(r|p) = 1/R$. This uniformity assumption simply disregards the expertise of a reviewer to a paper. Thus, we consider a more data oriented approach which estimates the probability based on the coverage score of reviewers for each paper. The estimation can be done in $O(PR)$.

Assume that for each paper p, we have the coverage score of all reviewers (Definition 1). A reviewer r is more likely assigned to a paper p in $\mathbb{O}$ if $c(\overrightarrow{r}, \overrightarrow{p})$ is high. However, if r has high coverage score for many papers, then we should lower r's probability to be assigned to each of those papers.[7] Accordingly, we define the probability of reviewer r to be correctly assigned to paper p as follows:

$$\mathcal{P}(r|p) \propto c(\overrightarrow{r}, \overrightarrow{p}) / \sum_{p' \in \mathbb{P}} c(\overrightarrow{r}, \overrightarrow{p}'), \quad (9)$$

where the numerator indicates how good (r, p) is and the denominator penalizes reviewers that have high coverage scores in many papers.

Equation 9 models the probability used by our stochastic refinement process to remove assignment pairs from $\mathbb{A}$. After running several refinement iterations, the effect of the probabilities degrades dramatically since the process may get stuck in a local maximum. Thereby, we slightly modify Equation 9 to reflect this by an exponential decay function as follows:

$$\mathcal{P}(r|p) \propto \max\{1/R, e^{-\lambda I} \cdot c(\overrightarrow{r}, \overrightarrow{p}) / \sum_{p' \in \mathbb{P}} c(\overrightarrow{r}, \overrightarrow{p}')\} \quad (10)$$

[7] This shares the same intuition as the TF-IDF model in IR.

where I indicates the number of refinement iterations run so far. We use $1/R$ as a constant so that every reviewer has a chance to be involved into the refinement process. We summarize our stochastic refinement process in Algorithm 3.

Algorithm 3 Stochastic Refinement Algorithm

Input: assignment $\mathbb{A}$ (from SDGA), papers $\mathbb{P}$, reviewers $\mathbb{R}$, workload δ_r, group size δ_p
Output: refined assignment $\mathbb{A}$
Algorithm $SRA(\mathbb{A}, \mathbb{P}, \mathbb{R}, \delta_p, \delta_r)$

```
for all p ∈ ℙ do
    for all r ∈ ℝ do  computes c(r⃗, p⃗)
while c(𝔸) does not converge do
    for all (r, p) ∈ 𝔸 do
        computes 𝒫(r|p) by Eq. 10
    for all p ∈ ℙ do
        removes a reviewer r from 𝔸[p] based on 1 − 𝒫(r|p)
    𝔸_s ← Stage-WGRAP(𝔸, ℙ, ℝ)
    𝔸 ← 𝔸 ⋃ 𝔸_s
return 𝔸
```

5. EXPERIMENTS

	Data Mining	Databases	Theory
Venues	SIGKDD'08,'09 ICDM'08,'09 SDM'08,'09 CIKM'08,'09	SIGMOD'08,'09 VLDB'08,'09 ICDE'08,'09 PODS'08,'09	STOC'08,'09 FOCS'08,'09 SODA'08,'09
#Papers	545, 648	617, 513	281, 226
Reviewers	SIGKDD'08,'09	SIGMOD'08,'09	STOC'08,'09
#Reviewers	203, 145	105, 90	228, 222

Table 3: Data used in the evaluation

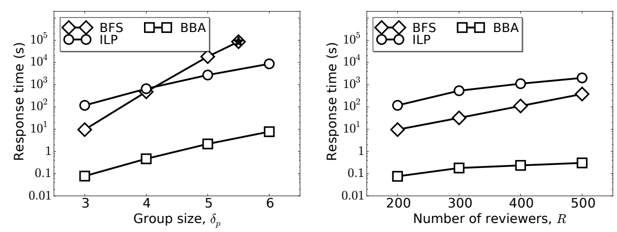
To evaluate our methods, we conducted experiments using academic publications in three research areas (Data Mining, Databases and Theory) over two years (2008 and 2009). We obtained the data from [26], since in this dataset the paper abstracts are available.[8] As set of reviewers, we take the program committee list from one venue of the corresponding area (e.g., the PC of SIGMOD 2008 is used to simulate the reviewers of the Databases area for year 2008). To extract the topic vectors of reviewers (cf. Section 2.4 and Appendix A), we collect the abstracts of their publication records from 2000 to 2009. We set the number of topics T to 30 (treated as a constant in this work). As we do not have records about the submitted papers at a conference, we simulate an artificial conference by taking the publications from 3 to 4 similar venues of the same year as the submitted papers (e.g., the submissions for Databases in 2008 are all considered to be all published papers in 2008 in SIGMOD, VLDB, ICDE, and PODS). The number of submitted papers and the set of reviewers for each area and year are summarized in Table 3.

All methods were implemented in C++, complied using VC2010, and the experiments were performed on a machine equipped with Intel Core 4-Cores (8-Threads) i7-2600 3.4GHz and 3.17GB main memory. The machine was running Windows 7 Enterprise 32bit.

[8]Abstracts are used in the process of extracting the topic vectors. See Section 2.4 and Appendix A for details.

5.1 Journal Reviewer Assignment

We first evaluate the efficiency of our Branch-and-Bound Algorithm (BBA) for JRA. We compare BBA with two competitors, Brute Force Search (BFS) and Integer Linear Programming (ILP). We use *lp_solve* [2] to solve ILP which is a C++ library based on the revised simplex method. In this section, we only evaluate the effect of δ_p (i.e., reviewer group size) and R (i.e., number of reviewer candidates) since the search space is only sensitive to the depth (i.e., δ_p) and the fan-out (i.e., R) of the tree. By default, the pool of candidate reviewers $\mathbb{R}$ includes all authors who published at least 3 papers in any of the three areas in 2005-2009 (a total of 1002 authors). In each experiment, we vary a single parameter, while setting the others to specific values. To evaluate the effect of R, we randomly select R reviewers candidates from the default pool $\mathbb{R}$. We report the average response time of all methods on 20 papers (i.e., p is randomly selected from the three areas).



(a) Effect of δ_p ($R = 200$) (b) Effect of R ($\delta_p = 3$)

Figure 9: Scalability evaluation of JRA

Figure 9(a) shows the response time of the methods as a function of group size δ_p, after setting R to 200. ILP and BBA are less sensitive to δ_p due to the effectiveness of the branching techniques (i.e., executing promising branches first). BFS cannot finish within 24 hours and ILP takes 2.4 hours to compute the result when $\delta_p = 6$, while BBA only takes 7.7 seconds to locate the optimal reviewer group. BBA is at least two orders of magnitude faster than ILP and BF.

Figure 9(b) shows the response time of the methods as a function of reviewer size R, after setting δ_p to 3. All methods are less sensitive to R than δ_p. Again, our method (BBA) is less sensitive to R than the other two methods due to the effect of branching and bounding. When $R = 500$, BBA is 1,252 and 6,661 times faster than BFS and ILP, respectively. For R=1,000 (not shown in Figure 9(b)), BBA is just 2.1 times slower (0.64 s) than BBA for $R = 500$ (0.3 s) while BFS and ILP are 33.9 and 3.4 times slower than their respective runs for $R = 500$. For completeness, we put two additional scalability experiments (i.e., changing the default value of R to 300 and of δ_p to 4) in Appendix C.

As JRA can be viewed as a general constraint programming problem [9], we also attempted to solve JRA by a commercial CP solver, IBM ILOG CPLEX Optimizer 12.6 (CPLEX) [8]. For a small problem instance (with $R = 30$ and $\delta_p = 3$), CPLEX takes 14.35 s to find the optimal assignment and uses 90 ms to return the first feasible assignment group. Our method (BBA) takes only 4 ms to return the optimal assignment. To our understanding, typical constraint programming techniques are not favorable to the group assignment problem due to the lack of a tight upper bound (cf. Equation 3).

5.2 Conference Reviewer Assignment

In the next set of experiments, we simulate the assignment process of two conferences (in the Databases and Data Mining areas of 2008)[9], for which the statistics (e.g., R and P) can be found in Table 3. We compare our proposed techniques, SDGA, and SDGA with stochastic refinement (denoted by SDGA-SRA), to other competitors, including Stable Matching (SM) [13] (i.e., a widely accepted approach in resource allocation problems), ILP (i.e., the objective is not a group coverage function), Greedy [22], and Best Reviewer Group Greedy (BRGG) (i.e., at each iteration, we find the best pair of group and paper (g, p) instead of best reviewer and paper (r, p), a method discussed in the beginning of Section 4.2). Regarding SM and ILP, they do not consider the quality of each assignment group at its entirety so that an interdisciplinary paper may be reviewed by a group of reviewers with too narrow expertise. Still, we include them in the experiments, for the sake of completeness. For SDGA-SRA, we set the convergence threshold $\omega = 10$. We also set the reviewer workload δ_r to the minimum possible value (i.e., $\delta_r = \lceil P \cdot \delta_p / R \rceil$). This setting is commonly used in the real world as the program chair would like to minimize the workload of each reviewer and make the assignment as balanced as possible. Moreover, this setting makes the problem more challenging as every reviewer must be involved in the assignment process.

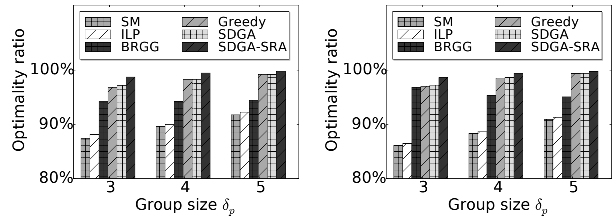
	SM	ILP	BRGG	Greedy	SDGA	SDGA-SRA
DB ($\delta = 3$)	0.1	7.6	11.6	0.1	5.9	46.3
DB ($\delta = 5$)	0.1	8.7	15.5	0.2	9.5	48.8
DM ($\delta = 3$)	0.1	16.0	30.2	0.3	5.5	44.3
DM ($\delta = 5$)	0.1	17.3	21.4	0.3	9.1	47.7

Table 4: Response time (s) of approximate methods

Even though the response time is not the main focus when evaluating approximation algorithms, it remains a significant performance factor in practice. We report the response time of all tested methods in Table 4. As expected, SDGA and SDGA-SRA are more costly than Greedy, due to the fact that these methods examine a larger part of the search space. Still, the response time, e.g., 5.9 s (SDGA) and 46.3 s (SDGA-SRA), is reasonable as the assignment is only computed once for a conference.

Optimality ratio. A reasonable approach to evaluate the quality of each assignment $\mathbb{A}$ is to compute its *approximation ratio* $c(\mathbb{A})/c(\mathbb{O})$ to the optimal assignment $\mathbb{O}$. However, computing $\mathbb{O}$ may take very long time even for small instances, due to the hardness of the problem. Instead, we compute the *optimality ratio* $c(\mathbb{A})/c(\mathbb{A}_{\mathbb{I}})$ of $\mathbb{A}$ to an *ideal assignment* $\mathbb{A}_{\mathbb{I}}$. To compute $\mathbb{A}_{\mathbb{I}}$, for each paper, we greedily assign to each paper the best set of δ_p reviewers, disregarding their workloads. $\mathbb{A}_{\mathbb{I}}$ may violate the constraint that each reviewer is assigned to at most δ_r, therefore, in general, $c(\mathbb{A}_{\mathbb{I}}) > c(\mathbb{O})$. Hence, $c(\mathbb{A})/c(\mathbb{A}_{\mathbb{I}})$ is a lower bound of $c(\mathbb{A})/c(\mathbb{O})$.

Figure 10 shows the optimality ratio of the methods as a function of reviewer group size δ_p. Although SDGA outperforms the simple methods SM, ILP and BRGG by a visible margin, SDGA performs similarly to Greedy. This is not



(a) Databases (b) Data Mining

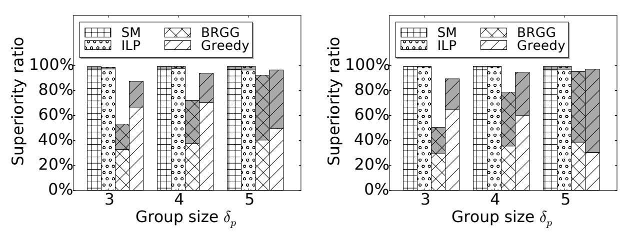
Figure 10: Optimality ratio

surprising, as Greedy is shown to provide good results in many NP-hard problems in practice. Still, in the majority of our tests (one exception can be found in Databases when $\delta_p = 5$), the optimality ratio of SDGA is better than that of Greedy due to the stage deepening technique. Given its stronger approximation guarantee and the better practical performance, we recommend SDGA as the best approach for WGRAP. In addition, after applying our stochastic refinement (SDGA-SRA), the optimality ratio becomes very close to 1, consistently outperforming the ratio of Greedy (from 0.39% for $\delta_p = 5$ on Data Mining to 1.91% for $\delta_p = 3$ on Databases). A substantial amount of papers get better coverage scores; for example, 389 (out of 617) papers, for $\delta_p = 3$ on Databases.

Superiority ratio. Next we investigate the assignment quality of each paper by a *superiority ratio* metric. Given two approximate result $\mathbb{A}_X$ and $\mathbb{A}_Y$, the ratio of X over Y is calculated as

$$ratio(X, Y) = \mid \{p \in \mathbb{P} \mid c(\overrightarrow{\mathbb{A}_X[p]}, \overrightarrow{p}) \geq c(\overrightarrow{\mathbb{A}_Y[p]}, \overrightarrow{p})\} \mid / \ P,$$

where the numerator indicates the number of papers, which get better or equal coverage score in $\mathbb{A}_X$ than $\mathbb{A}_Y$. The superiority ratio is an important quality factor, as it shows how many papers will get an assignment of better quality by the approximate solutions.



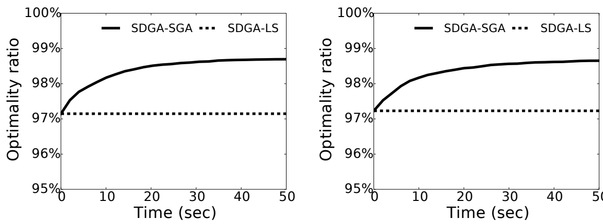
(a) Databases (b) Data Mining

Figure 11: Superiority ratio of SDGA-SRA

Figure 11 demonstrates the superiority ratio of SDGA-SRA over the three competitors, as a function of the reviewer group size δ_p. SDGA is omitted since SDGA-SRA is derived from SDGA. Each bar indicates the superiority ratio of SDGA-SRA over a competitor; the dark grey portion indicates the ratio of tie cases. For instance, almost every paper gets by SDGA-SRA a better or equal quality to SM and ILP, and at least 89.4% of the papers gets better or equal quality to Greedy. This result reveals that the stage execution

[9] As the results for different areas are similar, we moved the simulation for Data Mining/Database areas of 2009 and Theory area for 2008/2009 to Appendix C.

paradigm of SDGA and the stochastic process (SRA) help in finding a better reviewer group for each paper. SDGA-SRA is not better than BRGG since a portion of papers gets very good reviewer groups at the early stages of BRGG. However, this strategy harms the latter stage assignments, leading to a low coverage ratio (cf. Figure 10).



(a) Databases (b) Data Mining

Figure 12: Optimality ratio to stochastic refinement executions

Effectiveness of stochastic refinement. Next, we demonstrate the effect of our stochastic refinement process. We compare it with a standard, *Local Search* (LS) refinement approach. Our stochastic refinement can improve the optimality ratio by 1.4% and 1.2% in databases and data mining, respectively, after 20 seconds; while local search does not improve the overall quality since it gets stuck in a local maximum.

Summary. Our fully optimized approximate solution, SDGA-SRA, outperforms other methods by a wide margin considering different performance factors. More importantly, SDGA-SRA finds a solution with at least $1 - 1/e$ approximation ratio (for integral cases) and 1/2 ratio (for general cases), which is the best ratio achieved to the best of our knowledge. More comparisons, discussions, and case studies can be found in Appendix C. All paper assignment results are available via our project homepage (`http://degroup.cis.umac.mo/reviewerassignment/`).

6. CONCLUSION

In this paper, we formulated the automatic reviewer assignment to papers as a Weighted-coverage Group-based Reviewer Assignment Problem (WGRAP). Our formulation does not have the drawbacks of previous work. First, it balances the review load. Second, it uses a quality metric based on the coverage of the paper topics by the reviewers' expertise, giving weight to the topics according to their relevance to the papers. We studied the special case of Journal Reviewer Assignment and proposed an efficient algorithm that finds an exact solution for it. For the general WGRAP, we proposed a polynomial-time approximation algorithm, which improves the approximation ratio of previous work from 1/3 to 1/2. Finally, we proposed a stochastic refinement process that further improves the quality of the solution found by our algorithm. Our experimental results show that our algorithm paired with the stochastic refinement postprocessing achieves much better results compared to previous approaches and also runs within reasonable time. In the future, we plan to study alternative RAP formulations, e.g., where the quality of the assignment depends on both reviewer relevance to the paper topics and reviewer preferences based on available bids.

7. ACKNOWLEDGMENTS

This project was supported by grants MYRG188-FST11-GZG, MYRG109(Y1-L3)-FST12-ULH and MYRG2014-00106-FST from University of Macau RC, grant FDCT/106/2012/A3 from Macau FDCT, and grant HKU 715413E from Hong Kong RGC. We thank Y. Benjo for using his open source ATM project [1], the authors of [26] for the DBLP Citation network dataset, and the anonymous reviewers for their insightful comments.

8. REFERENCES

[1] `https://github.com/ybenjo/LDA`. Accessed: 2013-09-01.
[2] lp_solve 5.5. `http://lpsolve.sourceforge.net/5.5/`. Accessed: 2014-11-11.
[3] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. *Network flows: theory, algorithms and applications*. Prentice Hall, 1993.
[4] H. K. Biswas and M. Hasan. Using publications and domain knowledge to build research profiles: An application in automatic reviewer assignment. In *ICICT*, 2007.
[5] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.
[6] L. Charlin, R. S. Zemel, and C. Boutilier. A framework for optimizing paper matching. In *UAI*, 2011.
[7] D. Conry, Y. Koren, and N. Ramakrishnan. Recommender systems for the conference paper assignment problem. In *RecSys*, 2009.
[8] I. Corp. Ibm cplex cp optimizer 12.6. `http://www-01.ibm.com/software/commerce/optimization/cplex-cp-optimizer/index.html`. Accessed: 2014-12-29.
[9] R. Dechter. *Constraint processing*. Morgan Kaufmann, 2003.
[10] S. T. Dumais and J. Nielsen. Automating the assignment of submitted manuscripts to reviewers. In *SIGIR*, 1992.
[11] U. Feige. A threshold of ln n for approximating set cover. *J. ACM*, 45(4):634–652, July 1998.
[12] M. Fisher, G. Nemhauser, and L. Wolsey. An analysis of approximations for maximizing submodular set functions—ii. In *Polyhedral Combinatorics*, volume 8 of *Mathematical Programming Studies*, pages 73–87. 1978.
[13] D. Gale and L. S. Shapley. College admissions and the stability of marriage. *The American Mathematical Monthly*, 120(5):386–391, 2013.
[14] D. Hartvigsen, J. C. Wei, and R. Czuchlewski. The conference paper-reviewer assignment problem. *Decision Sciences*, 30(3):865–876, 1999.
[15] S. Hettich and M. J. Pazzani. Mining for proposal reviewers: lessons learned at the national science foundation. In *KDD*, 2006.
[16] T. Hofmann. Probabilistic latent semantic indexing. In *SIGIR*, 1999.
[17] D. S. Johnson, C. H. Papadimitriou, and M. Yannakakis. How easy is local search? (extended abstract). In *FOCS*, 1985.
[18] M. Karimzadehgan and C. Zhai. Constrained multi-aspect expertise matching for committee review assignment. In *CIKM*, 2009.
[19] M. Karimzadehgan, C. Zhai, and G. G. Belford. Multi-aspect expertise matching for review assignment. In *CIKM*, 2008.
[20] H. W. Kuhn. Variants of the hungarian method for assignment problems. *Naval Research Logistics Quarterly*, 3(4):253–258, 1956.
[21] X. Li and T. Watanabe. Automatic paper-to-reviewer assignment, based on the matching degree of the reviewers. In *KES*, 2013.
[22] C. Long, R. C.-W. Wong, Y. Peng, and L. Ye. On good and fair paper-reviewer assignment. In *ICDM*, 2013.
[23] D. M. Mimno and A. McCallum. Expertise modeling for matching papers with reviewers. In *KDD*, 2007.
[24] G. L. Nemhauser and L. A. Wolsey. *Integer and combinatorial optimization*, volume 18. Wiley New York, 1988.

[25] M. Rosen-Zvi, T. Griffiths, M. Steyvers, and P. Smyth. The author-topic model for authors and documents. In *UAI*, 2004.
[26] J. Tang, J. Zhang, L. Yao, J. Li, L. Zhang, and Z. Su. Arnetminer: extraction and mining of academic social networks. In *KDD*, 2008.
[27] F. Wang, S. Zhou, and N. Shi. Group-to-group reviewer assignment problem. *Computers & OR*, 40(5):1351–1362, 2013.
[28] X. Wang and A. McCallum. Topics over time: a non-markov continuous-time model of topical trends. In *KDD*, 2006.
[29] N. Xue, J.-X. Hao, S. Jia, and Q. Wang. An interval fuzzy ontology based peer review assignment method. In *ICEBE*, 2012.
[30] C. Zhai, A. Velivelli, and B. Yu. A cross-collection mixture model for comparative text mining. In *KDD*, 2004.

APPENDIX

A. TOPIC DISCOVERY

Topic modeling is a widely accepted method in Text Mining [5, 16, 25, 28, 30]. The objective is to extract a set of relevant and representative topics from a set of articles. Recent work [18, 19, 23] on the reviewer assignment problem adopt topic modeling for the extraction of topic vectors for reviewers and papers. As the topics can also be defined manually, the quality of topic extraction is orthogonal and independent to the development and necessity of WGRAP and any new findings on topic extraction are readily applicable and beneficial to WGRAP.

In this work, we apply the Author-Topic Model (ATM) [25] to extract the topic vectors of reviewers based on their publication records. Figure 13 illustrates the adapted ATM for our problem, where each variable is represented by a node (solid nodes for observed variables and hollow nodes for unknown variables to be estimated) and their dependencies are indicated by the arrows. The number of times each variable is replicated is indicated by the right-bottom number of each box. In our figure, there are R reviewers and the total number of their publications is D (i.e., $\{d_1, \cdots, d_D\}$). ATM assumes that every word of a document can be generated independently using the following iterative procedures.

1. Generate $\overrightarrow{r_i} \sim Dir(\alpha)$, $i \in \{1, \ldots, R\}$, where $Dir(\alpha)$ is the Dirichlet distribution for parameter α.
2. Generate $\overrightarrow{t_i} \sim Dir(\beta)$, $i \in \{1, \ldots, T\}$.
3. Generate each word w in a publication d as follows.
   (a) Generate a reviewer r_i based on $p(r|d)$, where $p(r|d)$ is a uniform distribution over reviewers who are the authors of d.
   (b) Generate a topic t_j based on $p(t|\overrightarrow{r_i})$
   (c) Generate a word w_k based on $p(w|\overrightarrow{t_j})$

Our goal is to estimate the topic vector of reviewers $\{\overrightarrow{r_1}, \cdots, \overrightarrow{r_R}\}$ and the topic set $\{\overrightarrow{t_1}, \cdots, \overrightarrow{t_T}\}$ such that the generated documents are as similar as possible to the observed variables D (i.e., the generative probability is maximized). The estimation can be done by Gibbs Sampling, as suggested in [25].

Given the topic set, we can easily derive the topic vector $\overrightarrow{p}$ of each submitted paper by Expectation Maximization (EM) [30] as follows.

$$\overrightarrow{p} = \underset{\overrightarrow{p}}{\operatorname{argmax}} \prod_{i=1}^{W_p} \sum_{j=1}^{T} p(w_i|t_j)\overrightarrow{p}[t_j], \qquad (11)$$

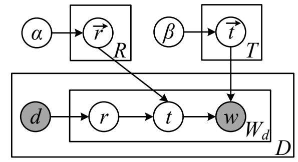


Figure 13: Author-Topic Model

where $\mathbb{W}_p$ indicates the set of words in p and $p(w_i|t_j)$ is estimated by ATM.

B. SUBMODULARITY AND ALTERNATIVE SCORING FUNCTIONS

According to the proofs of Lemma 3 and Theorem 1, the approximation ratio of SDGA still holds even when the objective function $c(\mathbb{A}) = \sum_{p\in\mathbb{P}} c(\overrightarrow{g}, \overrightarrow{p})$ (where $g = \{r|(r,p) \in \mathbb{A}[p]\}$) is replaced by alternative submodular functions. In this section, we first prove the submodularity of $c(\mathbb{A})$ subject to two conditions of the scoring function $c(\overrightarrow{r}, \overrightarrow{p})$. We then study 3 additional scoring functions and discuss their effect to WGRAP.

Proof of submodularity. $c(\mathbb{A})$ is submodular while (C.1) $c(\overrightarrow{r}, \overrightarrow{p})$ is a summation of topic scores, i.e. the contribution of different topics is independently counted, and (C.2) $c(\overrightarrow{r}, \overrightarrow{p})$ is monotonically increasing function w.r.t. $\overrightarrow{r}$, i.e., the assignment quality does not decrease when the expertise of a reviewer r (or a reviewer group g) increases. Obviously, our default scoring function (cf. Definition 1) fulfills these two conditions. A formal proof follows.

LEMMA 4. *The objective function $c(\mathbb{A})$ is submodular if (C.1) $c(\overrightarrow{r}, \overrightarrow{p})$ is a summation of topic scores and (C.2) $c(\overrightarrow{r}, \overrightarrow{p})$ is a monotonically increasing function w.r.t. $\overrightarrow{r}$.*

PROOF. $c(\mathbb{A})$ is submodular iff, $\forall (r,p), (r',p') \in \mathbb{R} \times \mathbb{P}$,

$$c(\mathbb{A} \cup \{(r,p)\}) - c(\mathbb{A}) \geq$$
$$c(\mathbb{A} \cup \{(r',p')\} \cup \{(r,p)\}) - c(\mathbb{A} \cup \{(r',p')\})$$

If $p \neq p'$, the marginal gains of (r,p) and (r',p') are independent. In other words, the gain of assigning (r,p) is not affected by whether we have assigned (r',p'), i.e.:

$$c(\mathbb{A} \cup \{(r,p)\}) - c(\mathbb{A}) =$$
$$c(\mathbb{A} \cup \{(r',p')\} \cup \{(r,p)\}) - c(\mathbb{A} \cup \{(r',p')\})$$

If $p = p'$, assignment pairs (r,p) and (r',p') refer to the same paper. In this case, r and r' belong to the same reviewer group g. According to the first condition (C.1), each topic t independently contributes to $c(\overrightarrow{r}, \overrightarrow{p})$. For the ease of our discussion, we use $f(\overrightarrow{r}[t], \overrightarrow{p}[t])$ to denote the contribution at topic t.

<u>When $\overrightarrow{r}[t] \geq \overrightarrow{r}'[t]$,</u>

$$\begin{aligned}
&f(\overrightarrow{g \cup \{r\}}[t], \overrightarrow{p}[t]) - f(\overrightarrow{g}[t], \overrightarrow{p}[t]) && \text{by Def. 2}\\
=&f(\overrightarrow{g \cup \{r, r'\}}[t], \overrightarrow{p}[t]) - f(\overrightarrow{g}[t], \overrightarrow{p}[t]) && \text{by C.2}\\
\geq&f(\overrightarrow{g \cup \{r, r'\}}[t], \overrightarrow{p}[t]) - f(\overrightarrow{g \cup \{r'\}}[t], \overrightarrow{p}[t]) && (12)
\end{aligned}$$

When $\overrightarrow{r}[t] < \overrightarrow{r}'[t]$,

$$f(\overrightarrow{g \cup \{r, r'\}}[t], \overrightarrow{p}[t]) - f(\overrightarrow{g \cup \{r'\}}[t], \overrightarrow{p}[t]) \quad \text{by Def. 2}$$
$$= f(\overrightarrow{g \cup \{r, r'\}}[t], \overrightarrow{p}[t]) - f(\overrightarrow{g \cup \{r, r'\}}[t], \overrightarrow{p}[t]) = 0 \quad (13)$$

Thereby,

$$f(\overrightarrow{g \cup \{r\}}[t], \overrightarrow{p}[t]) - f(\overrightarrow{g}[t], \overrightarrow{p}[t]) \quad \text{by C.2 and Eq. 13}$$
$$\geq f(\overrightarrow{g \cup \{r, r'\}}[t], \overrightarrow{p}[t]) - f(\overrightarrow{g \cup \{r'\}}[t], \overrightarrow{p}[t]) \quad (14)$$

Thus, we now prove the marginal gain of assigning (r, p) may decrease if we have assigned (r', p') first, i.e.:

$$c(\mathbb{A} \cup \{(r, p)\}) - c(\mathbb{A}) \quad \because g \text{ reviews } p$$
$$= \sum_{t=1}^{\mathbb{T}} f(\overrightarrow{g \cup \{r\}}[t], \overrightarrow{p}[t]) - f(\overrightarrow{g}[t], \overrightarrow{p}[t])$$

by Eq. 12 and 14

$$\geq \sum_{t=1}^{\mathbb{T}} f(\overrightarrow{g \cup \{r, r'\}}[t], \overrightarrow{p}[t]) - f(\overrightarrow{g \cup \{r'\}}[t], \overrightarrow{p}[t])$$
$$= c(\mathbb{A} \cup \{(r', p')\} \cup \{(r, p)\}) - c(\mathbb{A} \cup \{(r', p')\})$$

□

Alternative scoring functions. We study three alternative scoring functions, namely the reviewer coverage score, the paper coverage score, and the dot-product score. The reviewer (paper) coverage score employs a winner-takes-all strategy where the reviewer (paper) contributes to topic t if and only if $r[t] \geq p[t]$ (i.e., r is qualified to review p at topic t). The score computed by this strategy is based on only the reviewer expertise (paper contents). The dot-product score is widely used in many vector based similarity computations. All of these three functions are submodular, i.e., the satisfy the conditions of Lemma 4.

Table 6 is a toy example with two reviewers and one paper, demonstrating the effect of these scoring functions. The reviewer coverage score c_R prefers a reviewer with strong expertise on some specific topic(s) of the paper so that r_1 is preferred to r_2 due to r_1's expertise at these topic(s). The paper coverage score c_P focuses on whether a topic in the paper can be completely understood by the reviewer. Actually, we recommend to use the reviewer (paper) coverage score only when we are very confident about the reviewers' expertise information (paper topic distribution). The dot-product score c_D offers a fair evaluation based on both paper contents and reviewer expertise. However, it may overestimate the importance of some topic, e.g., topic t_1 of r_1 is 0.9, which likely returns a large score, although the paper's relevance to t_1 is 0.6 only. In this example, our default scoring function (i.e., weighted coverage c), is the only method that prefers r_2 to r_1. This result is more intuitive as r_2 is more similar to p than r_1 is. It should be noted that the main focus of this work is not to evaluate the appropriateness of the scoring functions. Instead, we provide flexibility in the choice of the objective function, as long as it is a submodular function, in which case our theoretical results (Theorem 1) hold.

name	symbol	function (numerator only)
weighted coverage (default)	c	$\sum_{t=1}^{T} \min\{\overrightarrow{r}[t], \overrightarrow{p}[t]\}$
reviewer coverage	c_R	$\sum_{t=1}^{T} \begin{cases} \overrightarrow{r}[t] & \text{if } \overrightarrow{r}[t] \geq \overrightarrow{p}[t] \\ 0 & \text{otherwise} \end{cases}$
paper coverage	c_P	$\sum_{t=1}^{T} \begin{cases} \overrightarrow{p}[t] & \text{if } \overrightarrow{r}[t] \geq \overrightarrow{p}[t] \\ 0 & \text{otherwise} \end{cases}$
dot-product	c_D	$\sum_{t=1}^{T} \overrightarrow{r}[t] \cdot \overrightarrow{p}[t]$

Table 5: Alternative scoring functions

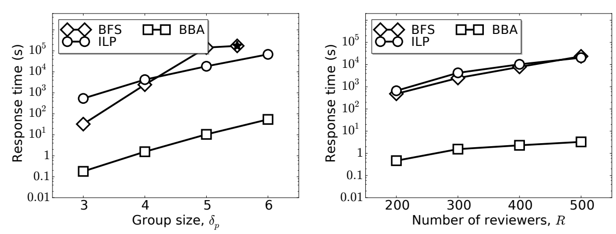
	t_1	t_2
$\overrightarrow{p}$	0.6	0.4
$\overrightarrow{r_1}$	0.9	0.1
$\overrightarrow{r_2}$	0.5	0.5

	$\overrightarrow{r_1}$	$\overrightarrow{r_2}$
$c_R(\overrightarrow{r}, \overrightarrow{p})$	0.9	0.5
$c_P(\overrightarrow{r}, \overrightarrow{p})$	0.6	0.4
$c_D(\overrightarrow{r}, \overrightarrow{p})$	0.58	0.5
$c(\overrightarrow{r}, \overrightarrow{p})$	0.7	0.9

Table 6: An example using the 4 scoring functions

C. ADDITIONAL EXPERIMENTS

Additional scalability evaluation of JRA. Figures 14(a) and 14(b) show the response time of the methods as a function of group size δ_p and number of reviewers R, respectively. The overall trend is similar to the result in Section 5.1. When $\delta = 6$, BF cannot finish in 48 hours and ILP takes 18.9 hours to compute the result while BBA only takes 53.8 seconds. Figure 15 evaluates the effect of k (i.e., top-k reviewer groups) under the default setting. BBA returns the best 1,000 reviewer groups within 2 seconds.



(a) Effect of δ_p ($R = 300$) (b) Effect of R ($\delta_p = 4$)

Figure 14: Additional scalability evaluation of JRA

Lowest coverage score. The lowest coverage score, $\min_{p \in \mathbb{P}} c(\overrightarrow{g}, \overrightarrow{p})$, models the quality of the worst assignment in a CRA approximate result. As Table 7 shows, SDGA-SRA significantly outperforms the other methods with respect to this metric, especially for low δ_p values. This is because the stage assignment of SDGA achieves a result closer to the optimal, when there are only few stages.

Effect of the convergence threshold ω in SRA. The convergence threshold ω is a parameter used to terminate the stochastic refinement process. For larger ω values, the execution time increases and we also expect the assignment quality to increase. Figure 16 demonstrates the effect of ω for $\delta_p = 3$, where the line indicates the response time and the bar represents the optimality ratio. Although the optimality ratio increases with ω, the response time decreases at a higher rate. We select $\omega = 10$ as our default setting

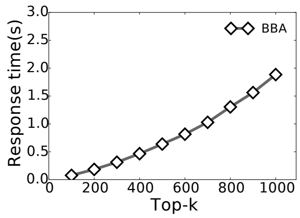
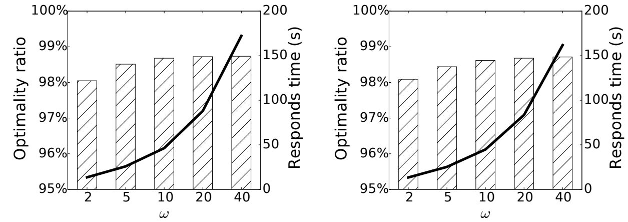


Figure 15: The effect of k on BBA

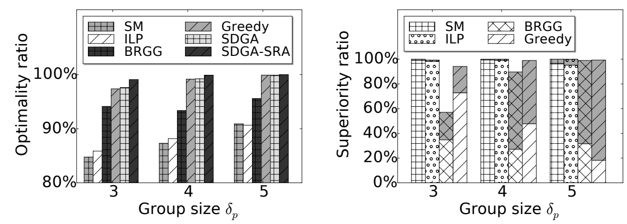


(a) Database (b) Data Mining

Figure 16: The effect of ω ($\delta_p = 3$)

	δ_p	SM	ILP	BRGG	Greedy	SDGA-SRA
DB08	3	0.03	0.42	0.16	0.56	0.59
	4	0.42	0.44	0.16	0.59	0.59
	5	0.53	0.46	0.16	0.59	0.59
DM08	3	0.45	0.45	0.40	0.77	0.75
	4	0.28	0.57	0.36	0.77	0.75
	5	0.44	0.58	0.36	0.77	0.76
T08	3	0.20	0.49	0.25	0.69	0.76
	4	0.59	0.49	0.48	0.79	0.79
	5	0.47	0.59	0.45	0.79	0.79
DB09	3	0.51	0.56	0.32	0.71	0.77
	4	0.33	0.56	0.26	0.78	0.78
	5	0.53	0.61	0.33	0.78	0.78
DM09	3	0.42	0.59	0.41	0.81	0.84
	4	0.36	0.62	0.30	0.86	0.86
	5	0.67	0.63	0.34	0.86	0.86
T09	3	0.16	0.50	0.13	0.66	0.70
	4	0.10	0.56	0.05	0.70	0.70
	5	0.23	0.63	0.19	0.70	0.70

Table 7: Lowest coverage score in $\mathbb{A}$



(a) Optimality ratio (b) Superiority ratio

Figure 17: CRA experiments in Theory (2008)

in this work as it offers good tradeoff between effectiveness and efficiency.

Additional quality evaluation of CRA. Figures 17 and 18 show the optimality ratio and the superiority ratio for the Theory area of 2008 and all three datasets for 2009. The overall trends have no difference to the results observed for the areas of Databases and Data Mining in 2008.

Case study (1). We take a closer look at the set of reviewers assigned to paper "Kun Liu, Evimaria Terzi: *Towards identity anonymization on graphs.* SIGMOD 2008: 93-106", in our Databases 2008 experiment. The result of different approaches are shown in Figure 19, where a bar indicates the coverage score for a specific topic and we only report the 5 most related topics in terms of their probability distribution (i.e., the probability of the remaining topics is low). For clarity, we show the related topics and their keywords in Table 8. Due to lack of space, we omit the result of SM (it performs similarly to ILP). As shown in Table 8 and Figure 19, the extracted topics (i.e., topic bars) accurately capture the contents of the paper (in fact, they also match the paper keywords provided by the authors, i.e., `Anonymity`, `Degree Sequence`, and `Dynamic Programming`).

We now analyze the expertise of the selected reviewers. Chris Clifton is an expert on data privacy, particularly with respect to analysis of private data, so he has relatively high weight on topic t_2 compared to the other topics. As another example, Philip Yu is an active scholar who has published at least 856 papers (according to DBLP) and works on diverse research topics. The majority of his research is related to data mining, therefore he has relatively high weight on topic t_5. We add a note that the expertise of Philip Yu only reveals that he offers strong support to review topic t_5 but this does not mean that he is not qualified for t_2 (since we treat the qualification of every committee member identically by normalization).

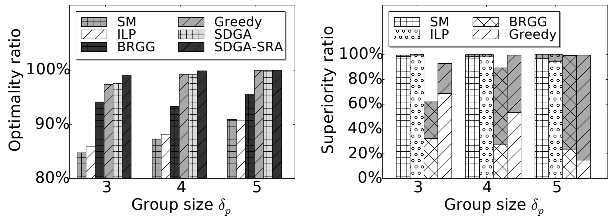
Regarding the assignment result, SDGA-SRA (i.e., our proposed implementation of WGRAP) returns the best assignment according to the coverage of the paper's topics. Note that SDGA-SRA is the only method which can find an expert (i.e., Philip Yu) to support topic t_5. Accordingly, the paper is expected to be reviewed in a more diverse way.

topic	keywords
t_1	algorithms, techniques, based, large, propose, efficient,...
t_2	privacy, access, control, security, sensitive, secure,...
t_3	stream, algorithm, approximation, online, string, traffic,...
t_4	graph, xml, similarity, matching, approximate, variety,...
t_5	clustering, stream, high, mining, classification, graph,...

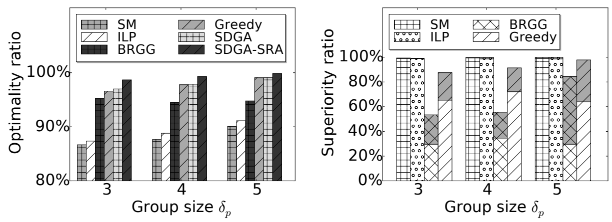
Table 8: Topics and keywords (for Case study 1)

Case study (2). Table 9 and Figure 20 analyze the assignment of another paper "Mirit Shalem, Ziv Bar-Yossef: *The Space Complexity of Processing XML Twig Queries Over Indexed Documents.* ICDE 2008: 824-832", having keywords `XML`, `computational complexity`, `indexing`, `query processing`. The coverage scores of all assignment results are relatively high compared to Case study (1). Again, SDGA-SRA is the only method that includes Christoph Koch, who has strong background in XML query processing, in the group of reviewers.

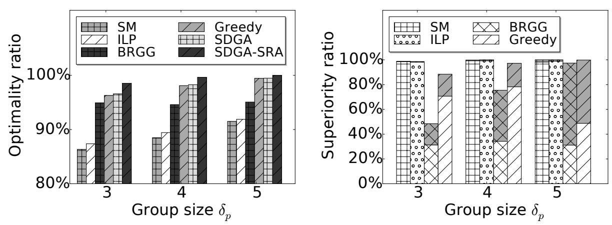
Note that our assignment results are computed solely based on the paper abstracts. The assignment quality is expected to be further improved if the topic extraction process uses the article's main text and the expertise on topics is self-tuned by the reviewers.



(a) Optimality ratio (T) (b) Superiority ratio (T)



(c) Optimality ratio (DB) (d) Superiority ratio (DB)



(e) Optimality ratio (DM) (f) Superiority ratio (DM)

Figure 18: CRA experiments in 2009 datasets

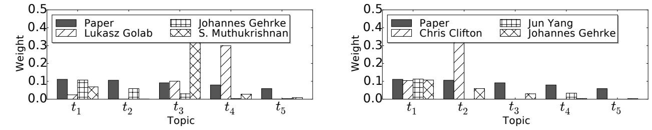
topic	keywords
t_1	algorithms, techniques, based, large, propose, efficient,...
t_2	query, processing, model, relational, optimization,...
t_3	xml, xquery, xpath, query, evaluation, tree, structure,...
t_4	graph, xml, similarity, matching, approximate, variety,...
t_5	stream, system, operators, plan, maintenance, processing,...

Table 9: Topics and keywords (for Case study 2)

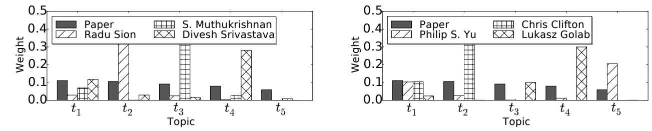
Quality evaluation of scoring functions. Figures 21(a), 21(b), and 21(c) show the optimality ratio of 3 alternative objective functions (cf. Table 5) on Databases area in 2008. The overall trends have no difference to those observed when using our default objective function (cf. Definition 1). Besides evaluating alternative scoring functions, we attempt to scale the expertise of the reviewers using their h-indices. Specifically, we scale the vector of each reviewer by a number in the range [1,2] as follows:

$$\overrightarrow{r} = (1 + \frac{h_r - h_{min}}{h_{max} - h_{min}})\overrightarrow{r} \quad (15)$$

where h_r denotes the h-index of reviewer r, while h_{max} and h_{min} denote the minimum and maximum h-indices of all reviewers, respectively. Similar to other experiments, SDGA-SRA performs well after the scaling (cf. Figure 21(d)).

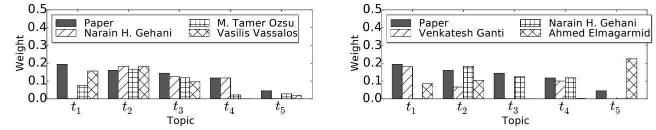


(a) ILP (Score = 0.77) (b) BRGG (Score = 0.64)

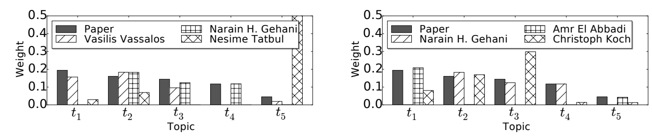


(c) Greedy (Score = 0.87) (d) SDGA-SRA (Score = 0.97)

Figure 19: Case study 1

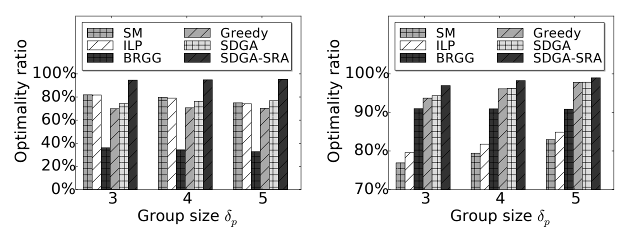


(a) ILP (Score = 0.91) (b) BRGG (Score = 0.85)

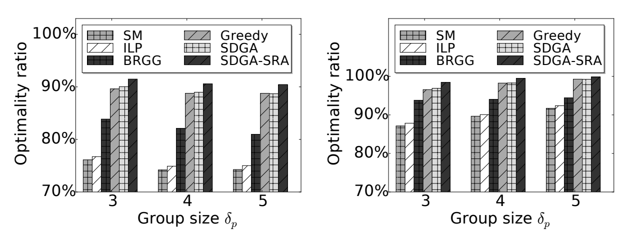


(c) Greedy (Score = 0.87) (d) SDGA-SRA (Score = 0.96)

Figure 20: Case study 2



(a) Optimality ratio by c_R (b) Optimality ratio by c_P



(c) Optimality ratio by c_D (d) Optimality ratio by h-index

Figure 21: The effect of alternative scoring functions