The New Automated IEEE INFOCOM Review Assignment System

Baochun Li and Y. Thomas Hou

Abstract

In academic conferences, the structure of the review process has always been considered a critical aspect of ensuring the quality of the conferences. Assigning reviews manually, by either the TPC chairs or Area Chairs, is time-consuming, and the process does not scale well to the number of submitted papers. Organizing a conference with multiple symposia (or tracks) helps its scalability, but predetermined boundaries between tracks may lead to inefficient use of reviewer expertise and suboptimal review assignment. Inspired by a rich literature on the problem of automated review assignment, we have designed and implemented a new review assignment system, called Erie, and successfully deployed it for IEEE INFOCOM 2015 and INFOCOM 2016. Implemented in Python, Erie is designed to use Latent Semantic Indexing to compute the suitability score between a submitted paper and a reviewer's representative papers, and to solve an optimization problem that maximizes the total suitability score across all submitted papers to the conference. Anecdotal evidence shows that Erie outperformed the accuracy of manual assignments by Area Chairs, and helped to improve the percentage of expert reviewers by a substantial margin.

As a high-quality international conference in the field of networking, IEEE INFOCOM receives over 1600 paper submissions every year. With a Technical Program Committee (TPC) consisting of more than 40 Area Chairs and more than 400 TPC members, it is both important and challenging that each submitted paper be assigned with some degree of accuracy to the most suitable Area Chair who oversees its review process, and to the most suitable set of TPC members for review. Review assignments serve as a good starting point of a rigorous and fair review process.

To maximize the amount of time for TPC members to review the submitted papers, it is desirable that the task of finalizing all review assignments is completed as quickly as possible. To cope with the potential scalability problem, the conventional wisdom used by large academic conferences is to divide a conference into multiple *tracks* or *symposia*, each handled separately by its own program committee. INFOCOM, however, has historically been organized as a *single-symposium* conference. This allows a submitted paper to be reviewed by any TPC member in the conference with the best match of expertise, free of boundaries between tracks/symposiums — a practice that has many benefits over a multi-symposium conference.

For a long time, INFOCOM has assigned submitted papers to Area Chairs and TPC members using a set of pre-defined research topics. Before the submission deadline, all Area Chairs, TPC members, and authors of submitted papers are provided with a pre-defined list of approximately *50 topics* (e.g., *cloud computing*) and 10 *methodologies* (e.g., *theory*). For each of these topics and methodologies, the Area Chairs and TPC members are asked to indicate their review preferences. The authors of each submitted paper are also asked to choose the topics and methodologies that are most representative of the paper. These choices are made in the online paper submission system used by INFOCOM, called EDAS.

To assign each submitted paper to Area Chairs in EDAS, the review assignment algorithm heavily relies on the idea of *paper claims*. Each TPC member is presented with a subset of submitted papers that fit into the topics and methodologies that he/she has previously indicated. For each submitted paper in the subset, one of four choices needs to be selected in a paper claim process, ranging from *want to review* to *cannot review*. After the process of claiming papers concludes, the review assignment algorithm in EDAS attempts to make final review assignments, favoring paper claim preferences. For papers with the same number of claims, EDAS assigns them to reviewers randomly.

Unfortunately, the process of claiming papers has always been both time consuming and error prone. To make sure that the EDAS algorithm is able to find feasible solutions, TPC members are typically asked to claim a much larger subset of papers than their maximum review load, which is laborious and unpleasant. As the conference became larger over the years, more and more TPC members failed to find the time to enter claims for their subsets of papers. It became apparent that as fewer paper claims were received, the review assignment process became more random, since the EDAS algorithm is designed to work best with paper claims. Such randomness was bound to deteriorate the quality of review assignment, as a substantial portion of TPC members failed to complete the paper claim process.

The paper claim process, while crucial for the EDAS algorithm to automate review assignment, is also problematic in that it is not resilient to collusion, when a small group of close-knit TPC members attempt to claim each other's papers. This problem ultimately motivated the process of manually assigning one of the reviews for each submitted paper by the Area

Baochun Li is with the University of Toronto.

Y. Thomas Hou is with Virginia Tech.

0890-8044/16/$25.00 © 2016 IEEE

Chairs. However, such manual assignment was again time consuming, as Area Chairs scrambled to manually select the most qualified TPC members for each of the 30–40 papers in their batch. Making matters worse, some popular or well-known TPC members may have quickly reached their review capacity, making it even more challenging for other Area Chairs in this manual assignment process.

The crux of the problem lies in the fact that the idea of associating a simple list of topics with both papers and TPC members, and then using some measure of overlap as a reflection of *suitability* between each submitted paper and TPC member, is crude at best. It is necessary to use additional information to help compute such suitability. The partial manual assignment by the Area Chairs, however, has only shifted the time-consuming workload from the TPC members to the Area Chairs. To keep INFOCOM as the largest networking conference with high quality, it is imperative to start to design a new system from scratch, automating each step in the process and minimizing any human factor.

In this article, we present our design, implementation, and experiences with *Erie*,[1] a new review assignment system that is designed to assign 1600 submitted papers for INFOCOM to over 500 TPC members and Area Chairs in a fully automated fashion. With *Erie*, we were able to make over 5000 review assignments within a day or two, relieving the Area Chairs from making manual assignments. Based on statistics from a new double-blind review process used in INFOCOM 2015 and 2016, we find that review expertise ratings have been dramatically improved. For example, the percentage of expert reviewers has increased from 20 percent without using *Erie* to 36 percent with its use.

State of the Art

The general problem of assigning submitted papers to the most suitable reviewers, typically called the *paper review assignment* problem, has been extensively studied. There are two aspects of this problem that stand out as the most important. *First*, to find the best reviewers for a specific paper, we need to find the best way to compute a numerical *suitability score* (also called *affinity* or *expertise* in the literature) between a reviewer and a submitted paper. *Second*, knowing such suitability scores between reviewers and papers, we need to design the best way to come up with an assignment of papers to reviewers for the entire conference, such that the sum of suitability scores is maximized, subject to a number of operational constraints.

In the context of INFOCOM, it is most desirable for the process of computing suitability scores to be fully automated, such that they can be computed in a batch on a scale of hundreds of reviewers and thousands of papers. There is a rich literature on this topic, and the general consensus was to take advantage of a set of published papers authored by the reviewer to compute his/her suitability score with a submitted paper. With published papers as representatives of the reviewer, computing a suitability score is essentially an information retrieval problem, in which a model needs to be established to analyze the degree of matching between the reviewer's representative papers and the submitted paper.

The paper by Mimno *et al.* [1] was one of the first to evaluate three alternatives for measuring the suitability score between a reviewer and a submitted paper, including a language model with Dirichlet smoothing for information retrieval (Ponte and Croft [2]), an author-topic model, and an author-persona-topic model. The evaluation was performed by comparing these methods with human annotation by the NIPS 2006 Program Committee as the ground truth. The three alternative methods were rather similar in their accuracy, and the author-persona-topic model with about 200 topics enjoyed a slight edge over the other two alternatives.

The second problem of coming up with an optimal assignment of submitted papers to the most qualified reviewers is, quite intuitively, a constrained optimization problem. Indeed, Taylor, in his technical report published in 2008 [3], proposed the first widely used formulation of such an optimization problem, with an objective of maximizing the overall sum of suitability scores globally, subject to constraints that each submitted paper should be assigned to no more than a certain number of reviewers, and no reviewer should be assigned more than a maximum workload of submitted papers. With binary assignment variables representing whether a paper is assigned to a reviewer, this is an integer programming problem, which is difficult to solve. However, Taylor pointed out that the constraint matrix is *totally unimodular*. This implies that the problem can be solved as a linear program using an off-the-shelf solver, and the values of assignment variables in the optimal solution are guaranteed to be binary (0 or 1).

Other papers in the literature have investigated many other detailed aspects of the paper review assignment problem. For example, in the context of the first problem of computing suitability scores, Hettich *et al.* [4] presented empirical experiences at the National Science Foundation using *Revaide*, which used the term frequency-inverse document frequency (TF-IDF) vector space model to annotate submitted proposals with a vector of top 20 representative terms. In the context of the optimal assignment problem, Tang *et al*. [5] formulated the problem as one that looked for feasible flows with minimum costs, based on which the optimal assignment can be computed. In addition, Long *et al*. [6] studied the effects of conflict of interest on fairness when topics were used for matching papers to reviewers. Similar to [4], a TF-IDF weighted vector space model was used for topic extraction in their experiments. Their proposed algorithm tried to maximize the total number of distinct topics covered by the assigned reviewers.

Toward a real-world implementation and deployment for academic conferences, the most visible work was Charlin *et al*.'s implementation [7], called the Toronto paper matching system. It has been used in machine learning and computer vision conferences since 2010, such as NIPS, ICML, and ICCV. It was also integrated into the Microsoft Research Conference Management Toolkit, a web-based system for handling paper submissions and review processes. The system adopted the language model proposed in Ponte and Croft [2], as well as the constrained optimization formulation proposed by Taylor [3].

Motivation

With a rich literature on the topic of automated review assignment, the conventional wisdom is to adopt one of the existing practices, such as the Toronto paper matching system, which already includes some of the well documented problem formulations and algorithms in the literature. However, after a few careful rounds of literature review, we decided to design and implement a new review assignment system from scratch for the following reasons.

First, we were concerned with the real-world performance of existing approaches and their ability to scale to thousands of papers and hundreds of reviewers. Such real-world performance was not well documented in the existing literature, and we were not confident that the entire assignment process could be completed in a short period of time without much manual intervention. If we were to design and implement a new system from scratch, we would be able to enjoy the flexibility of eval-

[1] The name comes from Lake Erie, one of the Great Lakes that is close to Toronto. TPC members also believed that the new system is eerily accurate.

uating design trade-offs and implementing various code-level optimizations to improve performance.

Second, rolling our own design and implementation affords us the freedom of experimenting, evaluating, and even switching between different design choices.

Finally, existing systems were not designed to work with EDAS, which enjoys many excellent features such as built-in similarity checking, a large CoI database, and automated formatting for exporting to the IEEE Digital Library (Xplore), among others. Seamlessly integrating with EDAS is important to take advantage of the features it offers, instead of duplicating these efforts.

Design

Toward the design of Erie, our new review assignment system, we wish to first ensure that it works well with EDAS when preparing input data, and then make sensible design choices with respect to how suitability scores can be computed, and how the constrained optimization problem can be formulated and solved.

Preparing Input Data

Most of the input data that are needed to run *Erie* are available and can be readily exported from EDAS. The review assignment process is divided into two stages. In the first stage, submitted papers are assigned to the Area Chairs so that they can oversee the review process, identify papers that are not suitable for the conference so that they can be rejected without review, and manually assign additional reviewers if needed. After a small fraction of papers have been identified to be rejected, the remaining papers will need to be assigned to the TPC members. We wish to use *Erie* to assign papers in *both* stages, first to the Area Chairs and then to the TPC members.

As a starting point, *Erie* requires the following data as its input:

- A list of Area Chairs and TPC members as reviewers, including their EDAS IDs as unique identifiers
- A list of submitted papers, including their paper IDs and PDFs
- A conflict-of-interest (CoI) matrix between reviewers and the submitted papers
- For each reviewer (either an Area Chair or a TPC member), a repository that contains a list of his/her published papers that best represent his/her expertise, in the form of PDF documents

The first three pieces of data can be exported from EDAS. With respect to the repository of published papers, the list does not have to be complete; it merely needs to represent the research expertise of the reviewer. Unfortunately, EDAS does not offer a facility to collect these representative papers from reviewers; we had to explicitly construct a new website for the Area Chairs and TPC members to log into and upload the list of representative papers in the form of PDF documents.

To ease the burden of uploading papers manually, our initial plan was to populate the website with a default set of papers for each reviewer, for example, papers published in the most recent five years. Although such a default set can be downloaded using a script from Google Scholar,[2] many of the downloaded papers were unfortunately incorrect. To address this problem, each TPC member was asked to log into the website and replace any paper with a different one that better reflected his/her own expertise.

Computing Suitability Scores

Assume that there are R reviewers and P submitted papers. To compute the *suitability score*, s_{rp}, between a reviewer and a submitted paper, we first need to evaluate the similarity between each of the reviewer's papers and the submitted paper. For each submitted paper, the *maximum* similarity score among all of the reviewer's representative papers is chosen as the suitability score between the submitted paper and the reviewer, which is in the range of (0, 1). A suitability score close to 1 means that the reviewer is highly likely to be an expert reviewer for the submitted paper, while a score close to 0 implies otherwise.

Extracting Text from PDF Documents: Before we start computing suitability scores, the full text of both submitted papers and the representative papers of all reviewers need to be extracted from PDF documents. Assuming each reviewer places 30 papers in his/her list of representative papers and 500 reviewers, we are operating at a scale of 15,000 papers that need to be processed. A reasonably good option to automate such a process is to use Python's *PDFMiner* package. Largely depending on the complexity of embedded figures, it takes between a second and several minutes to complete the text extraction process for each paper using *PDFMiner*. Unfortunately, a small fraction — about 2 percent — of PDF documents cannot be successfully extracted using this package. For these papers, we need to resort to professional tools[3] to manually process each of them.

Removing Stopwords: After text has been extracted from all the PDF documents, we need to clean up the text by removing all the *stopwords* in the English language, such as "we," "our," "be," and "about." In typical papers, about 25 percent of the words are stopwords in English. In the process of cleaning up the text, we have removed all 127 stopwords as defined in nltk.corpus, part of the Natural Language Toolkit (NLTK) Python package. We have also removed abbreviations such as "Fig." and "Sec." commonly used in many papers. At the same time, we have also converted all the words in each paper to lower case.

Computing the Similarity Score between a Submitted Paper and One of the Reviewer's Papers: We first need to adopt an algorithm to compute the similarity score between a submitted paper and *one* of the published papers authored by a reviewer, with a high degree of fidelity. We have considered three design choices.

TF-IDF Vectorization: The first, and perhaps simplest, algorithm is to first transform the pair of papers into vectors in the TF-IDF vector space, and then compute the cosine similarity between the two vectors. The TF-IDF value increases proportionally with the number of times that a word appears in the paper, offset by the frequency of the same word in the corpus. Existing works in the literature, such as [4, 6], also proposed to use the TF-IDF vector space model to compute the similarity between papers.

Latent Semantic Indexing: The second alternative involves the use of *latent semantic indexing* (LSI), a widely used dimensionality reduction technique to evaluate the similarity between a pair of documents in information retrieval. Also called *latent semantic analysis*, LSI takes advantage of singular value decomposition (SVD) to identify patterns in the relationship between words and their underlying concepts. It excels in its ability to extract "topics" from the text in a paper by establishing the association between words across similar contexts. It is language-independent as the approach is purely mathematical.

[2] It turned out that Google Scholar restricted the ability of scripts (robots) to download papers to which it referred, even though these papers were hosted elsewhere in the Internet. It was time consuming to manually switch to different IP addresses for the script to finish downloading papers for more than 400 TPC members and Area Chairs.

[3] We used an online conversion tool: http://document.online-convert.com/convert-to-txt.

Before LSI can be used, the paper needs to be first converted into a *bag of words*, which is simply a sparse vector of occurrence counts of the words used in the paper. To improve the performance of LSI, the information retrieval literature suggested the use of global weighting functions, such as *logarithmic entropy* or TF-IDF. In general, these functions all serve a similar purpose of promoting the importance of rare words. Previous results have shown that logarithmic entropy, as the global weighting function, is superior to TF-IDF [8]. As a result, we decided to apply logarithmic entropy as the weighting function before LSI was applied.

Latent Dirichlet Allocation: The third alternative is to apply latent Dirichlet allocation (LDA) [9], which has been mentioned and evaluated by Charlin *et al*. [7, 10] as well. Similar to LSI, LDA is another transformation from bag-of-words counts into a topic space of lower dimensionality. LDA can best be described as a probabilistic extension of LSI: its topics can be broadly interpreted as probability distributions over words. Similar to LSI, each paper is interpreted by LDA as a mixture of these topics. Due to the design of LDA, no global weighting functions were needed before using it to transform bag-of-words counts.

To make an informed decision on selecting one of these alternatives, we have implemented all three and gone through a detailed evaluation process. Among the three, the first we eliminated was the TF-IDF vectorization method, as our empirical observations — using a small set of papers with which we are familiar — clearly indicated that its accuracy was inferior to LSI. The comparisons between LSI and LDA were more extensive and involved, and we postpone detailed discussions on these evaluation results to the next section. The result was that LSI became our preferred choice when computing similarity scores between a reviewer's paper and a submitted paper. When applying LSI, the number of topics we selected was 400, which fell in the range of the commonly accepted "gold standard" of 200–500 topics. Although LSI was a standard method used to evaluate document similarity in information retrieval, we have not found existing discussions in the literature on using it in the context of review assignment systems.

Last but not least, both LSI and LDA require using a corpus of training documents to identify topics. We decided to use a corpus that includes 4992 papers that span 15 years of INFOCOM proceedings (2000–2014). After the full text has been extracted, stopwords have been removed, and the same set of transformations has been applied to these papers.

Optimizing Review Assignment

With the knowledge of a matrix of suitability scores between submitted papers and reviewers, the problem of matching papers to reviewers can be formulated as an integer programming problem to maximize the total suitability for all submitted papers, subject to a number of constraints:

$$
\begin{aligned}
&\max_{\alpha_{rp}} \sum_{r} \sum_{p} s_{rp} \alpha_{rp} \\
&s.t.\ \sum_{p} \alpha_{rp} \leq W_r^{\max}, \quad \forall r \\
&\qquad \sum_{p} \alpha_{rp} \geq W_r^{\min}, \quad \forall r \\
&\qquad \sum_{r} \alpha_{rp} = R_p, \quad \forall p \\
&\qquad \alpha_{rp} \in \{0,1\}.\forall r \quad \forall p
\end{aligned}
$$

where s_{rp} is the suitability score between the reviewer r and the paper p, α_{rp} is the binary assignment variable that has a value of either 0 or 1. $W_r^{\max}$ and $W_r^{\min}$ represent the minimum and maximum workload of a reviewer r, respectively, and R_p is the number of required reviewers for a paper p. This is essentially the same formulation as in Taylor's work [3] with minor variations, such as the addition of the minimum workload.

Even though integer programming problems can be computationally difficult to solve, as Taylor pointed out, the constraint matrix in this formulation is totally unimodular, which implies that we can treat it as a linear programming problem, and use a standard LP solver. The LP relaxation — allowing $\alpha_{rp} \in [0, 1]$ — does not affect the integrality of the optimal solution.

The only remaining issue is how CoI information can be incorporated into this formulation. Although one may consider setting the suitability score s_{rp} to $-\infty$, we have opted to add more equality constraints to guarantee that CoIs will never appear in the optimal solution. More specifically, we added the constraint $\alpha_{rp} = 0$ if a reviewer r and a submitted paper p have a CoI.

A final note is that in our formulation, the minimum and maximum workload for each reviewer can be different across reviewers, and the number of required reviews may also vary across different papers. This is quite handy, since such flexibility helped us to accommodate any manual review assignments exported from EDAS, perhaps by the Area Chairs or TPC Chairs, before solving the optimization problem. We have implemented this feature to accommodate an initial round of Area Chair manual assignments (one reviewer for each submitted paper) in INFOCOM 2015; INFOCOM 2016 did not use this feature since all the reviews were automatically assigned by *Erie*.

Implementation

Challenges and Solutions

We chose to use Python to implement *Erie*, mostly due to its built-in support for lists and dictionaries, the two most often used data structures in our implementation. With Python, it is quite straightforward to parse all the required data exported from EDAS. Thanks to a wide selection of third-party libraries in Python, we were able to implement most of the required components by using available tools in these libraries: we used *PDFMiner* to extract full text from PDF documents, NLTK to process the text extracted, *gensim* to implement all three alternative methods of computing similarity scores between the reviewers' and submitted papers, and *cvxopt* to solve the linear program for computing optimal assignment. Although all development, performance testing, and deployment were performed on a Macbook Pro with a 2.3 GHz quad-core Intel Core i7 "Haswell" processor (with 6 MB L3 cache and 16 GB physical memory), only one of the CPU cores was used by Python.

The development process was not entirely smooth sailing, unfortunately. On our first attempt, the LP solver in the cvxopt package failed to solve the optimization problem, since the size of the constraint matrix did not fit into memory with around 1600 submitted papers and 500 reviewers. We first tried to rewrite the implementation in MATLAB using lp_solve as the LP solver, and it ran out of memory as well. It turned out that we had to revise our Python implementation and switch to the use of sparse matrices so that the constraint matrices may fit into memory; even with sparse matrices as input, the LP solver used a total of 60 GB of virtual memory. Although the solver was able to find the optimal solution, it was painfully slow due to the fact that most of its implementation was written in Python. It took 14 hours on average to converge to the optimal solution for our problem size, and due to bugs in the cvxopt implementation, a few values in the solution were around 0.5, which was neither 0 nor 1.

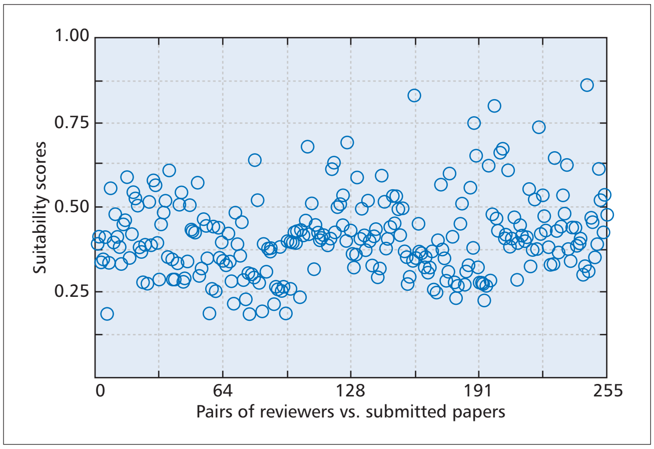

Figure 1. The scatter plot of suitability scores computed using latent semantic indexing, using a pilot pool of 51 submitted papers and five reviewers.

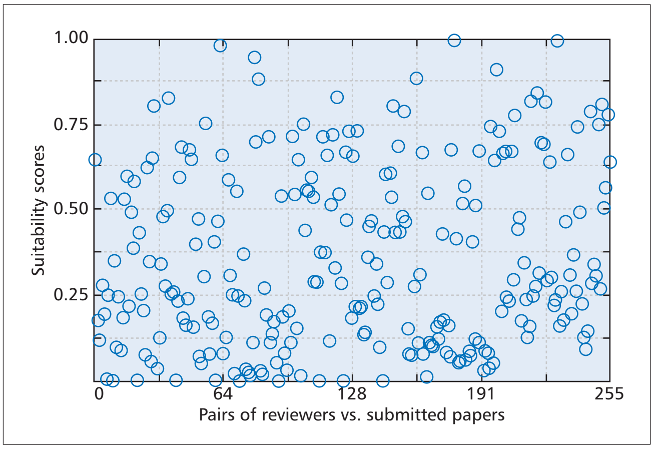

Figure 2. The scatter plot of suitability scores computed using latent Dirichlet allocation, using the same pilot pool of papers and reviewers as the LSI test.

Ultimately, we switched to a commercial LP solver, *Mosek*, that fortunately can be used by *cvxopt*. Mosek did not suffer from the same bugs as the default solver in *cvxopt*, and all values in the optimal solution were 0 or 1. In addition, Mosek implemented the interior point method, which was a vastly better choice for our problem. With Mosek, the optimal solution for our problem size could be computed within 30 s. Compared to the time needed for computing suitability scores, such a computation time was so short that it obliterated the need for designing new heuristics just for the sake of improving performance (e.g., [11]).

LSI vs. LDA: A Comparison

In order to compare the effectiveness of LSI and LDA, we have performed a careful evaluation at a small scale, with 5 reviewers and 51 randomly selected papers within the scope of INFOCOM. From a performance point of view, it was about 40 to 50 times slower to compute suitability scores with LDA than LSI. For example, with LSI, it took *Erie* 82 s to transform the dictionary from our corpus of around 5000 papers; with LDA, 3422 s were needed. To compute the similarity scores for 1000 pairs of papers, LSI needs 0.66 s, whereas LDA needs 35.8 s.

To illustrate the suitability scores computed (the maximum similarity score among a reviewer's batch of representative papers and a submitted paper), we show the scatter plot of suitability scores computed with LSI in Fig. 1, and those computed with LDA in Fig. 2. Our pilot pool contained 51 submitted papers and 5 reviewers (with 148 representative papers in total). The LSI-computed suitability scores had an average of 0.41, and a standard deviation of 0.12; and the LDA-computed scores had an average of 0.36, and a standard deviation of 0.26. It is apparent that, compared to LSI, LDA produced more diverging scores — either very low or very high — and far fewer in the borderline range (around 0.35–0.45 according to the subjective judgment of all the reviewers). In other words, LDA had a much more clear-cut "opinion" about many papers that LSI ranked as borderline. Unfortunately, this observation did not offer us conclusive evidence on which alternative performed better.

Continuing our comparison study, we conducted a detailed empirical study by collecting subjective opinions about the suitability of the papers assigned by solving the optimization problem. In the batch assigned to one of the reviewers, there were only four instances where LSI and LDA diverged in their judgment (i.e., one of them ranked the submitted paper below 0.35 and the other ranked higher than 0.44). The results were mixed: LSI made a better judgment on two of the papers, whereas the reviewer agreed with the evaluation of LDA on the other two.

We then conducted a more in-depth investigation on one of these papers for which the reviewer favored LDA over LSI. It was a paper that LSI found to be similar to one of the reviewer's published papers, with a similarity score of 0.57. We examined both papers, and they were indeed quite similar, both working on the topics of location and mobility. However, to the reviewer, that paper was considered an outlier as it was not directly related to his current research interests. Our method of using the *maximum* similarity score as the suitability score between the reviewer and the submitted paper, however, made the assumption that all the papers in the representative list are true representatives of the reviewer's expertise, with equal weighting.

Although it was a close call, we were in favor of adopting LSI to compute our similarity scores after additional rounds of extensive evaluations based on subjective opinions on a wide variety of papers. One of the reasons was that LSI offered more "granularity" across borderline scores in the range of (0.35, 0.45). Since solving our global optimization problem required meeting all the constraints about CoIs and review workload, it was generally unlikely for all the papers to be assigned the required number of expert reviewers. If a reviewer with borderline expertise must be assigned, it may help to have a better understanding on who these borderline reviewers are, as well as their relative ranking to each other.

Results and Experiences

Erie was first used in the review assignment process of INFOCOM 2015. Since it was developed from scratch and has not been extensively tested, we continued the practice of having Area Chairs manually assign one review for each paper first, and afterward employing *Erie* to assign the remaining two reviews for each paper. We employed *Erie* to make all assignments to the Area Chairs.

As expected, extracting and cleaning up text from all the reviewers' and submitted papers was a time-consuming process. Approximately 10,000 papers have been processed, most within 5 s, with some of the papers taking as long as a few hours for *PDFMiner* to report a failure. It took us a total of about three days to complete the entire process, with manual intervention required for about 2 percent of the papers that needed to be processed with a professional tool.[4] Fortunately, except for the submitted papers (around 1600), most of the work can be done well before the paper submission deadline.

Computing the matrix of suitability scores using LSI turned

[4] We plan to purchase better text extraction tools that are suitable for batch processing on a large scale, such as PDFlib, so that this process can be fully automated.

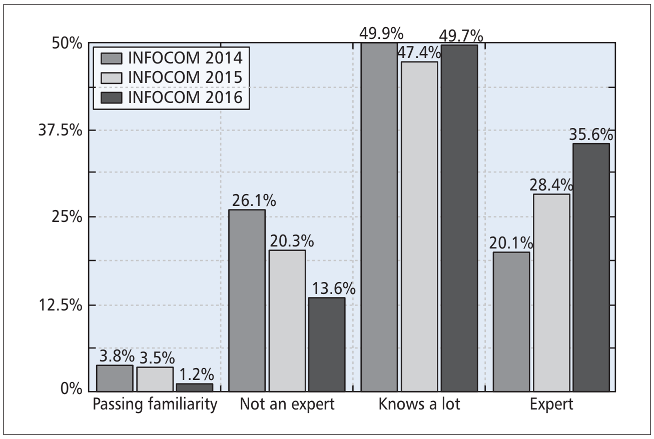

Figure 3. A histogram of the reviewer expertise distributions in INFOCOM 2014, 2015, and 2016.

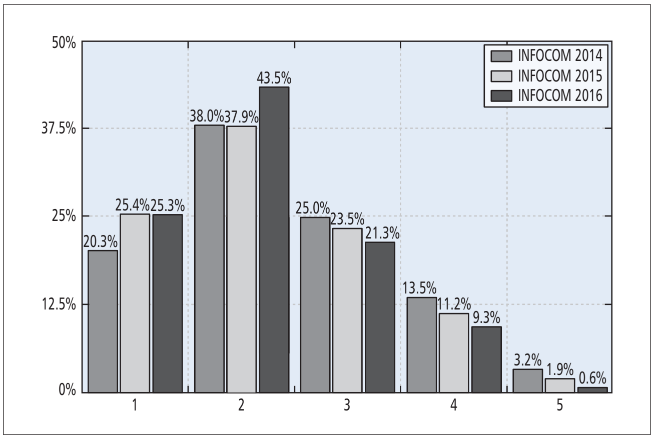

Figure 4. A histogram of the review rating distributions in INFOCOM 2014, 2015, and 2016.

out to be not as time-consuming as we expected. Recall that it took about 0.66 s on average to compute LSI-based similarity scores for 1000 pairs of papers. We had about 13 million pairs of papers that needed to be processed, and it took us about 2.5 hours to finish.

We experienced a rocky start due to bugs with CoI exported from EDAS and in our own implementation. But in the end, we succeeded in making around 3700 review assignments to TPC members, as computed by the LP solver in *Erie*, and avoiding all the CoIs. The anecdotal responses from the TPC members were very positive. For example, many applauded the system and commented that they have never seen such an accurate match to their research interests in the history of the conference. There was one incident where a TPC member was assigned a paper that plagiarized one of his own recently published papers with incremental and intentional changes, because *Erie* computed a very high similarity score between the two. Finally, we discovered that papers manually assigned to the TPC members by the Area Chairs did not match their research expertise as well as those assigned by *Erie*. This should not be surprising as a human's subjective perception cannot match *Erie*'s objective assignment once we adopt an objective metric.

Thanks to its anecdotal success, the INFOCOM Steering Committee decided to use *Erie* to make all the review assignments for INFOCOM 2016 by entirely eliminating the first round of Area Chair assignments. In Fig. 3, we show a histogram of the reviewer expertise ratings from the review statistics that we collected from INFOCOM 2014 (1/3 manually assigned by Area Chairs, 2/3 assigned by EDAS based on topics declared by TPC members), INFOCOM 2015 (1/3 manually assigned by Area Chairs, 2/3 assigned by Erie), and INFOCOM 2016 (100 percent assigned by Erie).

It was clear that the percentage of reviewers who subjectively declared the second lowest rating ("Not an expert") has nosedived from 2014 to 2016, and the percentage of those who assigned the highest expertise rating ("Expert") has seen a substantial increase, from 20 percent in 2014 to 36 percent in 2016. Recall that the difference between 2015 and 2016 was that one third of the reviews were assigned manually by Area Chairs in 2015, and 100 percent by *Erie* in 2016 without any action by the Area Chairs. The fact that we have seen a substantial increase in reviewer expertise has strongly supported the anecdotal evidence we received from the TPC members: *Erie* was able to outperform the quality of manual review assignments by the Area Chairs.

An intriguing phenomenon, which can be observed clearly in Fig. 4, was that the average review ratings have also decreased from 2014 to 2016, presumably due to the natural inclination that experts tend to view new contributions in their own areas more critically, since they are knowledgeable of the state of the art in the literature. Ironically, although it is an orthogonal problem that needs to be addressed, such a "race to the bottom" phenomenon has indirectly validated the effectiveness of *Erie*.

Discussions

The conventional wisdom used by a large academic conference is to divide the conference into multiple *symposia* or *tracks,* each with a predefined scope or research area. Each symposium has its own group of TPC members, typically led by one or several senior members of the research community as symposium/track chairs. When an author wishes to submit a paper, it needs to be submitted to one of the predefined tracks, and then be reviewed by TPC members within that specific track of the conference. As we have shown in Table 1,[5] conferences with more than 300 paper submissions almost invariably opted to be organized as multi-symposium/track conferences.

With multiple symposia, a large conference is divided into smaller ones, making it easier to manually assign reviews to the TPC members in each symposium within a reasonable amount of time. However, the use of multiple symposia also introduces several significant drawbacks. From the perspective of the authors who need to select a symposium to which to submit, as a result of submitting to one of the symposia, TPC members in the other symposia will have no opportunity to review the paper. This is not too much of a problem if the scopes of symposia are relatively well defined and the overlap of scopes between symposia is minimal. However, since a large conference typically covers dozens of research topics, dividing these topics into symposia often introduces scope ambiguities and overlaps. If a paper is submitted to one of the suitable symposia, TPC members in the other symposia that are potential matches will have no opportunities to review it — a terrible mistake in some cases.

From the perspective of TPC members, many of them tend to work in multiple areas of research that may shift over time, but can typically participate in one particular symposium due to time constraints. The expertise of these TPC members will not be utilized in all the other symposia in which they do not participate, implying that a potentially significant capacity of reviewer expertise would be left untapped in a multi-symposium conference.

Since it is difficult to predict the number of submitted papers in each symposium, it is hard for program or symposium chairs to invite the corresponding number of TPC members, with an expected number of review assignments in

[5] Source: Networking Conferences Statistics, available online at https://www.cs.ucsb.edu/~almeroth/conf/stats/.

Conference	Submitted papers (approx.)	Number of symposia/ tracks	Average number of papers per symposium
ACM CoNEXT 2015	136	1	136
IEEE ICNP 2014	160	1	160
ACM MobiCom 2012	212	1	212
ACM SIGCOMM 2013	240	1	240
IEEE ICDCS 2014	500	11	45
IEEE ICC 2014	2608	21	124

Table 1. The scale of conferences and the number of symposia.

mind for each TPC member. It can often occur that a popular symposium does not have sufficient reviewing capacity to handle the influx of papers, while another symposium in the conference is under-subscribed. It is typically not appropriate to move either papers or TPC members across the boundary between symposia after the submission deadline.

The root cause of these problems lies in a trade-off between the quality of review assignment and the overhead of management. Having multiple symposia makes a conference potentially easier to manage, but the quality of review assignment for the overall conference suffers: although local optimality may be achieved within a particular symposium in a multi-symposium conference, it is highly unlikely for a multi-symposium conference to achieve global optimality at the conference level, simply due to additional artificial constraints of enforcing boundaries between symposia. From a management point of view, *Erie* is quite capable of operating at scales that involve thousands of submitted papers and hundreds of TPC members. Within a day or two after the submission deadline, it is able to compute a set of optimal review assignments. As discussed, it is far superior than multi-symposium conferences in terms of achieving conference-level global optimality in assigning all submitted papers to all the reviewers.

Concluding Remarks

Although the jury is still out on what the best possible review assignment may be, we have implemented a new review assignment system, *Erie*, based on best practices in the literature and our own improvements and experimental evaluations. For each submitted paper, *Erie* uses latent semantic indexing to compute its similarity with each of the reviewer's representative papers, and the maximum score is used as the suitability score (expertise) of the reviewer for this submitted paper. With a matrix of suitability scores, *Erie* solves a constrained optimization problem that maximizes the total suitability of all papers, subject to workload and CoI constraints.

Erie has been successfully used for two thirds of the reviews assigned for INFOCOM 2015, and all of the reviews assigned for INFOCOM 2016. We have observed convincing anecdotal and statistical evidence that *Erie* has quite dramatically improved the quality of review assignments in INFOCOM's new double-blind review process. As future work, it would be ideal if *Erie* can be used in a web application — such as one developed in Python using the Django framework — as a turn-key solution that is easy to use by the program chairs.

Acknowledgments

The authors would like to thank Prof. Lavy Libman, University of New South Wales, Australia; Prof. Wenjing Lou, Virginia Tech; Prof. Shiwen Mao, Auburn University; and Prof. Guoliang Xue, Arizona State University for their participation in our pilot evaluation studies during the development process. The authors would also like to thank Prof. Sergio Benedetto, past President of the IEEE Communications Society, for his encouragement to document our experiences in an article and share it with the research community. The authors are grateful to Dr. Harvey Freeman, current President of the IEEE Communications Society, for supporting our efforts in this article.

References

[1] D. Mimno and A. McCallum, "Expertise Modeling for Matching Papers with Reviewers," *Proc. 13th ACM SIGKDD Conf. Knowledge Discovery and Data Mining*, San Jose, CA, Aug. 2007, pp. 500–09.

[2] J. M. Ponte and W. B. Croft, "A Language Modeling Approach to Information Retrieval," *Proc. 21st ACM SIGIR Conf. Research and Development in Information Retrieval*, Melbourne, Australia, Aug. 1998, pp. 275–81.

[3] C. J. Taylor, "On the Optimal Assignment of Conference Papers to Reviewers," Dept. Computer and Info. Sci., Univ. of PA, tech. rep. MS-CIS-08-30, Jan. 2008; http://repository.upenn.edu/cis_reports/889

[4] S. Hettich and M. J. Pazzani, "Mining for Proposal Reviewers: Lessons Learned at the National Science Foundation," *Proc. 12th ACM SIGKDD Int'l. Conf. Knowledge Discovery and Data Mining*, Philadelphia, PA, Aug. 2006, pp. 862–71.

[5] W. Tang, J. Tang, and C. Tan, "Expertise Matching via Constraint-Based Optimization," *Proc. 2010 IEEE/WIC/ACM Int'l. Conf. Web Intelligence and Intelligent Agent Technology*, Toronto, Canada, Sept. 2010, pp. 34–41.

[6] C. Long *et al.*, "On Good and Fair Paper-Reviewer Assignment," *Proc. 13th IEEE Int'l. Conf. Data Mining*, Dallas, TX, Dec. 2013, pp. 1145–50.

[7] L. Charlin, R. Zemel, and C. Boutilier, "A Framework for Optimizing Paper Matching," *Proc. 27th Conf. Uncertainty in Artificial Intelligence*, Barcelona, Spain, July 2011, pp. 86–95.

[8] M. D. Lee, B. M. Pincombe and M. B. Welsh, "An Empirical Evaluation of Models of Text Document Similarity," *Proc. 27th Annual Conf. Cognitive Sci. Soc.*, Stresa, Italy, July 2005, pp. 1254–59.

[9] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet Allocation," *J. Machine Learning Research*, vol. 3, Mar. 2003, pp. 993–1022.

[10] L. Charlin and R. S. Zemel, "The Toronto Paper Matching System: An Automated Paper-Reviewer Assignment System," *Proc. Wksp. Peer Reviewing and Publishing Models, 30th Int'l. Conf. Machine Learning*, Atlanta, GA, June 2013.

[11] N. M. Kou *et al.*, "Weighted Coverage Based Reviewer Assignment," *Proc. 2015 ACM SIGMOD Int'l. Conf. Management of Data*, Melbourne, Australia, May 2015, pp. 2031–46.

References

BAOCHUN LI [F'15] received his B.Engr. degree from Tsinghua University in 1995, and his M.S. and Ph.D. degrees from the University of Illinois at Urbana-Champaign in 1997 and 2000, respectively. He is currently a professor in the Department of Electrical and Computer Engineering at the University of Toronto, and holds the Bell University Laboratories Endowed Chair in Computer Engineering. His research interests include cloud computing systems, large-scale distributed systems, applications of network coding, and wireless networks.

Y. THOMAS HOU [F'14] received his B.E. degree from the City College of New York in 1991, his M.S. degree from Columbia University in 1993, and his Ph.D. degree from New York University Polytechnic School of Engineering in 1998, all in electrical engineering. He is currently the Bradley Distinguished Professor of Electrical and Computer Engineering at Virginia Tech, Blacksburg. His research interests include solving complex cross-layer optimization problems in wireless networks. He serves as Chair of the IEEE INFOCOM Steering Committee.