Nicholas D. Sidiropoulos and Efthymios E. Tsakonas

Signal Processing and Optimization Tools for Conference Review and Session Assignment

How to ease the burden on technical chairs and improve review quality at the same time

Anyone who has served as a technical program committee (TPC) chair for a conference (or program manager for a funding agency) understands that paper (or proposal panel) review assignment is a demanding job that takes a lot of time, and reviewers are rarely satisfied with the end results. This article presents signal processing tools for two critical "mass assignment" tasks: assigning papers (or proposals) to reviewers in a way that matches reviewing expertise to scientific content while respecting the reviewers' capacity constraints and splitting accepted papers (or submitted proposals) to sessions (panels) while adhering to session (panel) capacity constraints. The basic idea is to use feature vectors to represent papers and reviewers. Features can be key words or phrases (e.g., *optimization* or *sensor networks*) or other types of attributes (e.g., *timeliness*). This viewpoint enables optimal assignment problem formulations that make sense from a scientific and practical point of view. While optimal solutions are hard to compute for a large number of papers and

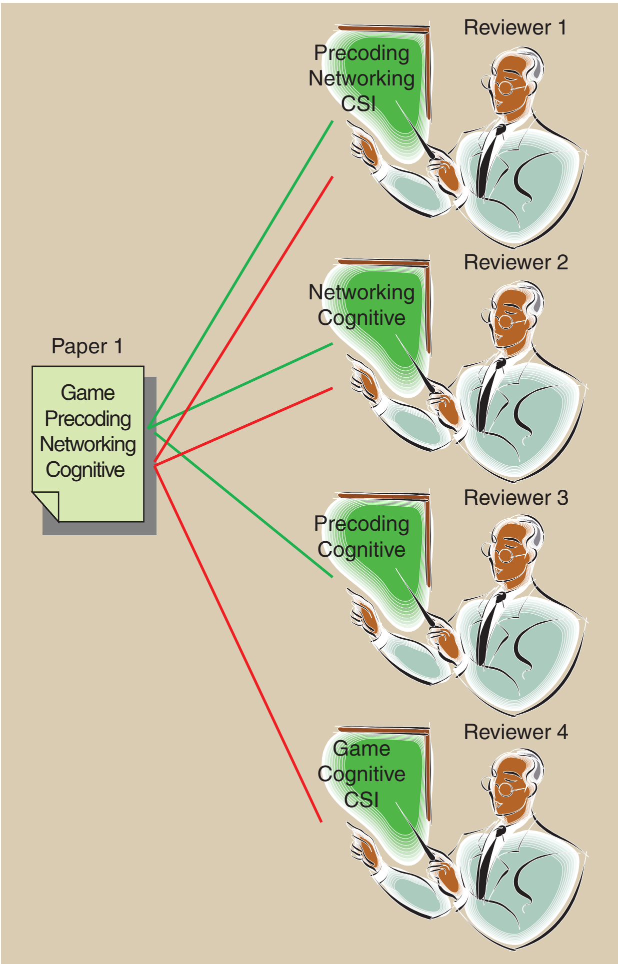
IMAGE LICENSED BY GRAPHIC STOCK

Digital Object Identifier 10.1109/MSP.2014.2359230
Date of publication: 6 April 2015

reviewers, high-quality approximate solutions of moderate complexity are developed here using familiar signal processing and optimization tools. These algorithmic solutions easily outperform days of expert manual work as demonstrated in experiments with real conference data.

The credibility of our scientific enterprise relies heavily on the peer-review system. Whereas many contributions are eventually still individually judged (e.g., when submitted for journal publication), there are at least two important modes of mass peer review at the center stage of scientific innovation: proposal review panels and conference reviewing. Paper and proposal review assignment is a difficult and tedious job that takes a lot of time, and, despite good intentions, often results in some awkward assignments.

A TPC chair's job includes 1) assigning reviewers to each paper, making every effort to match reviewing expertise to paper content while respecting the reviewer capacity constraints; 2) reading the submitted reviews and making an accept/reject decision for each paper, keeping in mind the target acceptance rate and the number of papers that can be presented at the conference; and 3) splitting the accepted papers into sessions, such that each session has a coherent theme, while adhering to session capacity constraints. The latter is the paper-to-session assignment problem. A program manager's job likewise includes 1) splitting the list of submitted proposals into smaller thematic batches to be assigned to review panels while adhering to panel capacity constraints (the proposal-to-panel assignment problem), 2) selecting reviewers to invite for each panel, and 3) assigning panelists to each proposal, trying to match reviewing expertise to the proposal content while respecting the panelist capacity constraints.



[FIG1] An example of what usually happens when one tries to maximize reviewer satisfaction (or similarity score) alone: Both green and red assignments have the same affinity score, but only the assignment in red ensures a scientifically sound paper review.

Given the difficulty and effort it takes to effectively solve these assignment problems, it is hard to believe that most TPC chairs and many program managers still operate without using the appropriate algorithmic aids to get the job done faster and better. There are two main reasons for this: 1) it is hard for a machine to nail down the essence of a submitted research paper or proposal and make a scientifically sound call on what is an appropriate set of reviewers, and 2) conference- or program-specific constraints require custom coding.

Generic computerized assignment algorithms (e.g., Cyberchair) are available, but these rely on reviewing "bids" or preference ratings, or a scalar similarity score between the contents of each paper and the expertise of each reviewer. Given the similarity (or affinity) paper-reviewer matrix, an assignment that maximizes the total affinity can be formulated as an integer linear programming problem. This formulation has been shown to be a totally unimodular program, which implies that an optimal solution can be computed at a modest complexity [1]; see also [2]. Using reviewer preferences for assignment certainly keeps the reviewers happy; however, it has two important pitfalls.

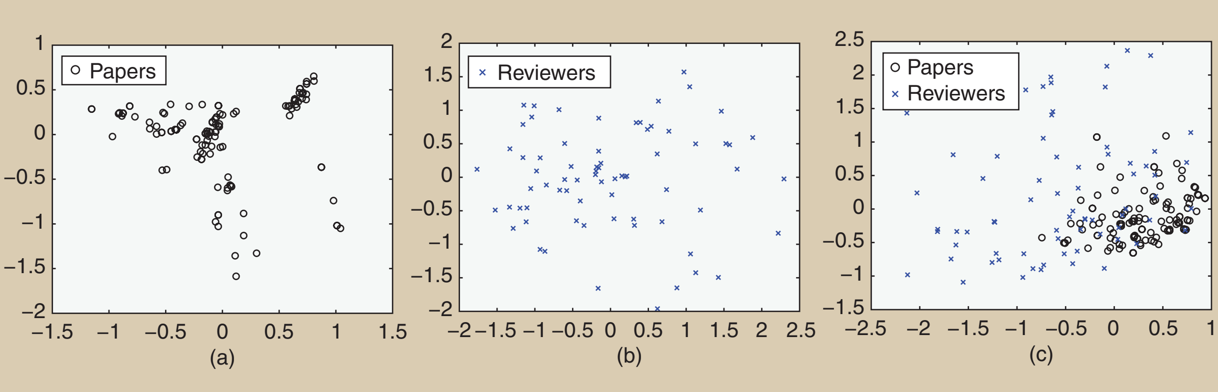
1) Each paper or proposal usually requires multiple types of expertise for proper review. For example, a paper on cross-layer resource allocation in wireless networking requires expertise in physical layer wireless communication, optimization, and networking. Using an aggregate preference or similarity score per reviewer can (and does) result in assignments where no reviewer covers a certain aspect of the paper (e.g., networking). This is, of course, highly undesirable, as already noted in some earlier work on automated review assignment [1], [3], [4]. A typical situation is depicted in Figure 1, which clearly shows the deficiency of total similarity/affinity score-based assignments. [While it is conceptually possible that one might be able to judiciously design a paper-reviewer score matrix that prohibits such bad assignments when used in conjunction with the totally unimodular programming approach in [1] and [2], this seems like a daunting task. Entry (p, r) of such a matrix should not only depend on the feature vectors of paper p and reviewer r; it should be a function of the feature vectors of potentially all papers and all reviewers.]

2) Reviewers tend to down-weight past experience in favor of their current interests when clicking on topical areas to summarize their expertise and generally bid to review papers or proposals that are close to their current interests, "in fashion," or from well-known researchers, without regard to the collective reviewing needs of the conference or panel. The TPC chair or program manager often has to tap a reviewer's past expertise to

VISUALIZING PAPERS AND REVIEWERS

The number of key words/features used to describe papers and reviewers will typically be in the order of dozens, making it hard to visualize the distribution of papers and reviewers in a feature space. One approach is to compute the first two or three principal components and project those points onto the principal subspace for visualization. Another tool that is commonly used for visualization is multidimensional scaling (MDS). Given a matrix of pairwise distances between m objects, MDS computes a map of m points in two-dimensional (2-D) [or three-dimensional (3-D)] space that approximately preserves the given distances.

Figure S1 shows 2-D MDS maps of points corresponding to papers and reviewers from the Signal Processing for Communications and Networking Technical Committee (SPCOM TC) track of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP) 2009, for which Nicholas D. Sidiropoulos served as TPC chair. The dimension of the vector space is $N = 44$—that is, there are 44 key words, and each paper or reviewer is represented by a (sparse) 44×1 vector. Figure S1 shows a map of (a) papers, (b) reviewers, and (c) a joint map of papers and reviewers. Notice how papers are clustered in (a), but this is not evident from the joint map (c). The reviewers are almost uniformly scattered, which speaks for the difficulty of optimal assignment: real data do not nicely fall in clusters. This situation is typical in our experience.



[FIGS1] (a) An MDS visualization of papers, (b) reviewers, and (c) joint papers and reviewers.

ensure a fair and unbiased assignment to the extent possible. These factors are very difficult to capture by reviewing preferences or aggregate similarity scores.

The first step toward a more pragmatic approach is a multidimensional description of each reviewer and each paper or proposal, in a common feature space that captures the essential dimensions of expertise for the specific scientific domain. In other words, we advocate viewing reviewers and papers/proposals as points in a higher-dimensional vector space. The canonical coordinates in this vector space are key words or phrases used to represent papers and reviewers (e.g., *optimization* or *sensor networks*), or other types of attributes (e.g., *timeliness*). This concept is illustrated in Figures 2 and 3, and it is central to our approach (see also "Visualizing Papers and Reviewers"). Note that feature vectors are widely used in the machine-learning literature; see, e.g., [5] and [6].

The list of keys for the papers (dimensions of the feature vector) can be produced as follows:

- The list can be prepared by the TPC chair before submission, in which case authors can mark the features relevant to their paper at the time of submission. This would correspond to a refined Editors' Information Classification Scheme.
- They can be compiled by taking the union of standard plus free-text key words provided by the authors at submission time, followed by stemming to consolidate synonyms.
- They can be parsed from the list of submitted paper titles. This parsing can be done manually by the TPC chair (for up to a few hundred papers—a seasoned chair can process about three papers per minute), or it can be automated using text retrieval [7] and consolidation tools [8]. Natural language processing will likely be helpful in this context, but this remains to be seen in practice. At any rate, spending a couple of hours producing a list of keys and marking papers is far less than what is needed for producing a well-rounded technical program from the list of accepted papers, let alone producing a scientifically sound review assignment.
- Most conferences and workshops recur annually or periodically; therefore, a prepared list of key words for the previous edition can serve as an excellent starting point for the next one, with the addition of a few key words for emerging topics and possible deletion or consolidation of those that are obsolete.

Drawing upon this multidimensional description of papers and reviewers, this article aims to present signal processing tools for paper-to-reviewer assignment and paper-to-session assignment. We examine these two problems in the remainder of this article.

A PRIOR ART

PAPER-TO-REVIEWER ASSIGNMENT

In addition to the key works [1], [9] and related follow-up work, such as [2], there are several more references on mass review assignment, e.g., [10] and the references therein. Those that are related to our viewpoint are reviewed in this article. Our vector space viewpoint of review assignment is implicit in [3], which considered representing each reviewer and each proposal with a list of key words or terms in a common term space and proposed evaluating reviewing assignments and making additional reviewer recommendations by measuring how the assigned reviewers collectively cover a proposal's key words; see also later work in [4]. The work of Hettich and Pazzani [3] is in fact a lucid and very insightful account of lessons learned in designing and implementing an early review aide system at the National Science Foundation (NSF) several years ago. What is missing from [3] (and [4]) is formulating review assignment as a joint optimization problem subject to reviewing capacity constraints, addressing complexity issues, and coming up with suitable algorithms to solve it. Instead, a simple greedy hill-climbing approach to making individual recommendations one reviewer at a time is discussed in [3]. Paper-to-session/proposal-to-panel assignment is not discussed at all in [3] and [4].

Today, several NSF program managers use a tool developed in [11] for review assignment. The approach in [11] is based on panelist reviewing preferences and uses a generalized assignment formulation with a branch-and-bound solution technique that is complex for large problems; however, it is tailored for the NSF panel review and complexity is not a major issue for modest panel sizes. On the other hand, it does not account for the need to cover all bases in reviewing a particular proposal or the bias that is typical in reviewing preferences. Additional work related to review assignment can be found in [12]; see also [13] for a recent overview of assignment problems.

PAPER-TO-SESSION ASSIGNMENT

Fitting the accepted papers into sessions is a clustering problem under equality constraints on the number of points per cluster—because each session has a fixed capacity. In this article, we focus on clustering using a centroid model, in which each cluster is represented by a single mean vector, and we have a given number of data points per cluster. In our context, each cluster corresponds to a session, and its centroid reflects the key words that are dominant in that session, thereby serving as a crude session title (which can be polished later by the TPC chair). The traditional signal processing and computer science literature treats clustering mostly using the well-known k-means algorithm [14], which cannot be directly applied in our context due to the presence of the session capacity constraints. Modifications of k-means to account for must-link/cannot-link constraints are discussed in [15], distance-type constraints on the cluster centers are discussed in [16], and lower-bound constraints on the number of points per cluster are discussed in [17]. As an alternative to alternating optimization-based k-means, approximation algorithms based on convex (semidefinite) optimization [18] are also known; see, e.g., [19] and the references therein.

Our formulation of paper-to-session assignment can be called a *capacitated k-means* problem. Whereas the general literature on clustering is immense [20], [21], we did not find any prior work on capacitated k-means, likely because there is no motivation to specify cluster sizes a priori in most applications of unsupervised clustering—where we typically know little about the clusters we are trying to find. Imposing a lower bound on cluster size may seem reasonable to avoid degeneracy, but an upper bound does not make sense in most other applications.

In practice, paper-to-reviewer assignment naturally precedes paper-to-session assignment. Paper-to-reviewer assignment is more challenging than paper-to-session assignment because there are typically many more papers submitted than accepted and many more reviewers than sessions in the final program. Furthermore, paper-to-reviewer assignment quality is more important from a scientific and ethical point of view. Yet the paper-to-session assignment problem is important and hard in its own right (we will show that it is NP-hard, in fact). There is also something special about the paper-to-session assignment problem: it is near and dear to our signal processing hearts. We will show how to modify k-means to account for strict cluster capacity constraints and produce a very practical and efficient low-complexity algorithm. We will also develop a more sophisticated one-shot approximation that can be used in smaller paper-to-session assignment problem instances. For these reasons, and despite the conceptual order of the two problems, we will start from the paper-to-session assignment problem. Before proceeding to the mathematical formulations, we first briefly review the mathematical tools that will be used.

MATHEMATICAL PRELIMINARIES

Assignment problems are optimization problems of a combinatorial nature; some have a special structure that enables efficient solution, while others are provably hard, even though they may not look all that different at first sight. The good news is that some of these problems can be well approximated (albeit not optimally solved) using convex optimization tools.

One way to deal with an optimization problem that is hard to solve is to efficiently obtain an approximate solution through convex relaxation. This comprises two steps (if the cost function of the original problem is not convex, then an additional transformation is required). In the first step, one replaces the feasible region of the original problem with a convex superset (hence the term *relaxation*); then the resulting problem is solved using convex optimization algorithms. In the second step, one converts the solution of the relaxed problem into a good admissible solution for the original problem through suitable postprocessing. The postprocessing step involves projection of the solution of the relaxed problem (and possibly related candidates generated via randomization) onto the feasible set of the original problem. Obviously, the optimal value of the relaxed problem provides a bound on the optimal value of the original problem; one goal is to find the tightest such bound (make the relaxation as tight as possible), as this impacts the quality of the final solution. We now illustrate how the idea of convex relaxation applies to both paper-to-session and review assignment.

RELAXATION OF PAPER-TO-SESSION ASSIGNMENT

The main algorithm is given in the section "Proposed Algorithm for Paper-to-Session Assignment" and is based on alternating optimization; see [16] and the references therein. This is an iterative procedure for optimizing a cost function by alternating conditional updates of different subsets of variables given the rest of the variables. However, we also show in the section "Gauging the Optimality Gap: Semidefinite Relaxation" that paper-to-session assignment can be equivalently rewritten as a quadratically constrained quadratic program (QCQP). This has the form

$$\begin{aligned} &\underset{x}{\text{minimize}}\ \mathbf{x}^{\mathrm{T}}\mathbf{Q}\mathbf{x} \\ &\text{subject to: } \mathbf{x}^{\mathrm{T}}\mathbf{C}_i\mathbf{x} \le b_i,\ \ i = 1, \ldots, n+1, \end{aligned} \tag{1}$$

with $\mathbf{Q}$ and $\mathbf{C}_i \in \mathbb{R}^{n\times n}$ symmetric matrices and b_i scalar quantities. Casting paper-to-session assignment as a QCQP is interesting since there are many tools available in the literature for quadratic optimization and they are well understood. The best convex relaxation bounds for (1) are based on semidefinite relaxation (SDR) [22]: one starts by 1) rewriting the quadratic cost in (1) as $\mathrm{Tr}(\mathbf{x}^{\mathrm{T}}\mathbf{Q}\mathbf{x}) = \mathrm{Tr}(\mathbf{Q}\mathbf{x}\mathbf{x}^{\mathrm{T}})$ (and similarly rewriting every quadratic constraint), and then 2) lifting the problem in a higher dimensional space using the change of variables $\mathbf{X} = \mathbf{x}\mathbf{x}^{\mathrm{T}}$. This lifting isolates the nonconvexities of the original QCQP into a single rank-1 constraint. The rank-1 constraint is subsequently relaxed into a convex, positive semidefinite cone constraint [18], or even simply dropped, thereby producing a convex (relaxed) problem. This is the main idea of SDR—the details of the transformation along with the corresponding postprocessing step, which produces the final approximate solution, are described in the section "Gauging the Optimality Gap: Semidefinite Relaxation."

PAPER-TO-REVIEWER ASSIGNMENT

As we explain in detail in the section "The Review and Assignment Problem," the associated optimization problem has the following form:

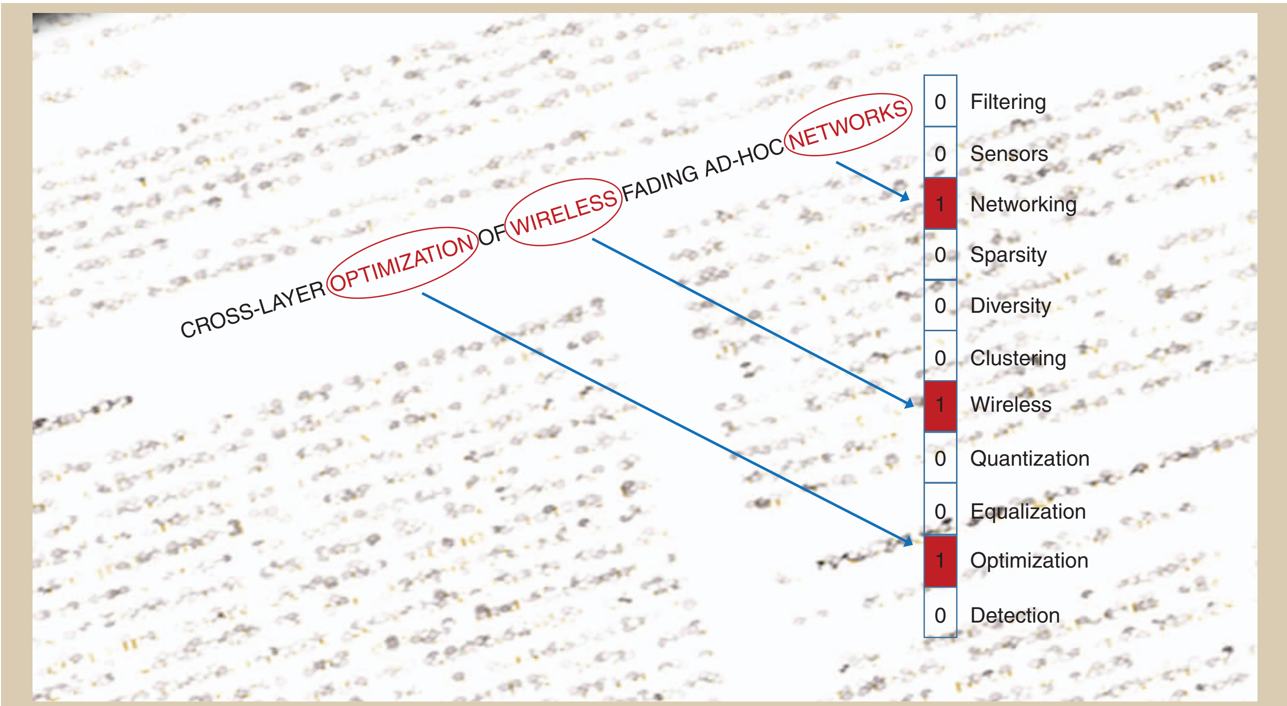
$$\begin{aligned} &\underset{\mathbf{x}}{\text{minimize}}\ f(\mathbf{x}) \\ &\text{subject to: } \mathbf{A}\mathbf{x} \preceq \mathbf{b},\ \mathbf{x} \in \{0,1\}^n, \end{aligned} \tag{2}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is a convex piecewise linear function in the variables $\mathbf{x} \in \mathbb{R}^n$, and $\preceq$ indicates componentwise inequality. The set defined by the inequality $\mathbf{A}\mathbf{x} \preceq \mathbf{b}$ is convex and is called a *polyhedron* [18]. Note that, even though the cost in (2) is convex, the design variables are Boolean, either zero or one. Boolean constraints are nonconvex constraints; in fact, it is often convenient to write them explicitly as quadratic equalities since $\mathbf{x}_i \in \{0,1\} \Longleftrightarrow \mathbf{x}_i(1-\mathbf{x}_i) = 0$.

In the first step, we produce the tightest convex relaxation (to be concise, the phrase "tightest convex relaxation" should be interpreted as "tightest relaxation in the class of Langragian relaxations"; see [18]) of (2): it can be shown that this is tantamount to replacing the Boolean constraints on the $\mathbf{x}_i$s with the interval ones $0 \le \mathbf{x}_i \le 1$ [18, Ch. 5]. We refer to this relaxation as *linear programming relaxation* because the resulting problem can be cast as a linear program (LP). Since the relaxed solution is not guaranteed to be Boolean, in the second step (the postprocessing), we make use of the structure of $\mathbf{A}$ and the nature of $\mathbf{b}$ to efficiently compute the Euclidean projection of the relaxed solution onto the feasible set of (2). This is the main idea—we defer the details to the section "The Review Assignment Problem."

PAPER-TO-SESSION ASSIGNMENT: A CLOSER LOOK

Recall from Figure 2 that we use feature vectors to represent papers. If N is the number of features (key words), then feature vectors are nonnegative vectors in $\mathbb{R}^N$. Let $\mathbf{p}_i$ be the $N \times 1$ feature vector for paper $i \in \mathcal{I} \triangleq \{1, \ldots, I\}$, where I is the total number of



[FIG2] Representing a paper as a point (feature vector) in the key word space. In this illustration, the feature vector is Boolean, with 1 if the paper possesses the specific key word and 0 otherwise.

(accepted) papers. Define the $N \times I$ matrix $\mathbf{P} := [\mathbf{p}_1, \cdots, \mathbf{p}_I]$. The capacity of session $j \in \mathcal{J} \triangleq \{1, \cdots, J\}$ is denoted c_j; $\sum_{j=1}^{J} c_j = I$, i.e., the total number of accepted papers.

The design variables are the $N \times J$ matrix of session centers $\mathbf{S} := [\mathbf{s}_1, \cdots, \mathbf{s}_J]$, where $\mathbf{s}_j$ is the center (profile, or title) of session j; and the $J \times I$ paper-to-session assignment matrix $\mathbf{X}$. The elements $\mathbf{X}_{ji}$ of $\mathbf{X}$ must satisfy the following constraints:

$$\mathbf{X}_{ji} \in \{0, 1\}, \forall i \in \mathcal{I}, j \in \mathcal{J}, \tag{3a}$$

$$\sum_{j=1}^{J} \mathbf{X}_{ji} = 1, \forall i \in \mathcal{I}, \tag{3b}$$

$$\sum_{i=1}^{I} \mathbf{X}_{ji} = c_j, \forall j \in \mathcal{J}. \tag{3c}$$

Here, $\mathbf{X}_{ji} = 1$ means that paper i is assigned to session j. The constraint $\sum_{j=1}^{J} \mathbf{X}_{ji} = 1$ ensures that paper i will be assigned to the one and only one session, whereas $\sum_{i=1}^{I} \mathbf{X}_{ji} = c_j$ enforces the capacity constraint for session j.

For brevity, let $\mathcal{A}$ denote the set of matrices $\mathbf{X} \in \mathbb{R}^{J \times I}$ that satisfy (3a)–(3c). With these definitions, the paper-to-session (or technical program optimization) problem can be posed as follows: assign papers to sessions (pick $\mathbf{X}$) and find the appropriate "session titles" (pick $\mathbf{S}$) to

$$\underset{\mathbf{S}, \mathbf{X}}{\text{minimize}} \; \| \mathbf{P} - \mathbf{S}\mathbf{X} \|_F^2, \tag{4a}$$

$$\text{subject to: } \mathbf{X} \in \mathcal{A}. \tag{4b}$$

See "Distance Considerations" for a discussion on the choice of distance measure.

A property worth pointing out explicitly is that any matrix feasible for (4) is row-orthogonal. To see this, define the vector of session capacities $\mathbf{c} = [c_1, c_2, \cdots, c_J]^T$ and the $J \times J$ matrix $\mathbf{\Lambda} = \text{Diag}(\mathbf{c})$, with the entries of the vector $\mathbf{c}$ on its main diagonal and zero elsewhere. Then, we have that

$$\mathbf{X} \in \mathcal{A} \Longrightarrow \mathbf{X}\mathbf{X}^{\mathrm{T}} = \mathbf{\Lambda}. \tag{5}$$

This observation will be useful on multiple occasions later on, in the problem transformations.

REMARK 1

Note that, in principle, one can place inequality constraints on the session capacities instead of the equality constraints (3c). Inequality constraints on the capacities make sense perhaps for poster sessions, but not for oral sessions, where a fixed number of papers should be presented. Although using inequalities instead of equalities is possible, the overall treatment of the problem (in particular, the material in the section "Gauging the Optimality Gap: Semidefinite Relaxation") becomes more involved. We choose to work with equality constraints to simplify exposition; after all, the TPC chair can explore minor reallocations of poster session capacities by running the proposed algorithms a few times if so desired. Also note that collisions (an author having to present simultaneously in two parallel sessions) are usually handled at the end by permuting the order of the presentation of papers in oral sessions or manual reallocation to a different session if a poster presentation is involved. Such scheduling conflicts are usually rare and also depend on the metadata, such as who is the presenting coauthor, and session time-scheduling, which in turn depends on the session content, the number of parallel tracks, room

DISTANCE CONSIDERATIONS

Returning to (4),

$$\min_{\mathbf{S}, \mathbf{X} \in \mathcal{A}} \| \mathbf{P} - \mathbf{S}\mathbf{X} \|_F^2 \Longleftrightarrow \min_{\mathbf{X} \in \mathcal{A}} \sum_{j=1}^{J} \min_{\mathbf{s}_j} \sum_{i \in I_j(\mathbf{X})} \| \mathbf{p}_i - \mathbf{s}_j \|_2^2,$$

the use of the Euclidean distance can be motivated as follows. Assume that the $\mathbf{p}_i$'s are drawn from J classes, with each class represented by a class mean, $\mathbf{s}_j$. A paper drawn from class j follows a multivariate Gaussian distribution $\mathcal{N}(\mathbf{s}_j, \sigma^2 \mathbf{I})$. Different papers are independently distributed, and we know the number of papers in each class (the session capacities). Then, maximum likelihood joint paper classification and class mean estimation reduces to the above formulation, as can be easily seen by taking the log-likelihood and invoking independence.

The Gaussian assumption/Euclidean distance can be motivated in many ways; a testament to its ubiquity is that classical k-means uses Euclidean distance. But there are many alternatives that might be worth investigating. If clusters appear to be oriented, then a Mahalanobis distance (quadratic form involving the inverse cluster covariance matrix) is more appropriate, but the cluster covariance(s) should be estimated as well. If the $\mathbf{p}_i$'s can be modeled as probability mass functions, then the Kullback–Leibler divergence can be well-motivated; see also [40] for a tutorial overview of clustering with Bregman divergences.

In our numerical experiments, we have limited ourselves to using binary feature vectors, mainly because this is enough to capture the essence of the problems considered. Richer alphabets are needed to capture the degree of expertise required in each latent dimension—some papers may only need common expertise in a particular area, while others may demand much deeper understanding. If we stay with binary features, however, then a more natural metric is the Hamming distance $d_H(\mathbf{p}_i, \mathbf{s}_j) = \sum_{n=1}^{N} 1(\mathbf{p}_i(n) \neq \mathbf{s}_j(n))$. This corresponds to saying that the probability of drawing $\mathbf{p}_i$ from class j is $q^{d_H(\mathbf{p}_i, \mathbf{s}_j)}(1-q)^{N - d_H(\mathbf{p}_i, \mathbf{s}_j)}$, for some $q < 0.5$, so the more likely vectors are those with few bit flips. If we also force the estimated $\mathbf{s}_j$s to be 0-1 binary, the Hamming distance reduces to ℓ_1-distance, i.e., the sum of absolute values. Then, the conditional update of each $\mathbf{s}_j$ is the elementwise median of the vectors in the cluster. Although not shown here, in many of these variations, the conditional update of $\mathbf{X}$ given $\mathbf{S}$ is also tractable, i.e., it reduces to a totally unimodular LP.

The appropriateness of any assumption and engineering design is ultimately judged by how well it performs in practice. Euclidean distance works well enough in our context, as illustrated in our experiments with real conference data.

capacities, etc. While it is possible to incorporate some of these aspects in the problem formulation, we prefer to keep the exposition simple and address the core problem instead.

COMPLEXITY OF OPTIMAL PAPER-TO-SESSION ASSIGNMENT

If we drop the session capacity constraints (3c) from (2), a classic k-means problem emerges. k-means is NP-hard; in loose terms, this means that we cannot expect to solve an arbitrary instance of k-means in time polynomial in the number of papers I. In the signal processing community, k-means is also known as *vector quantization* (*VQ*), usually dealt with using the celebrated (generalized) Lloyd–Max (GLM) [23], [24] or Linde–Buzo–Gray (LBG) algorithm [25], which is an alternating optimization procedure. The reason we usually resort to LBG is precisely because the problem is hard, and the LBG iteration offers an attractive simplicity-performance-complexity tradeoff. Proof that k-means is NP-hard was only recently provided [26], [27].

Here, we are actually dealing with a restriction of the VQ/k-means problem due to the session capacity constraints, which will always be active. We show next that, unfortunately, this restriction is also an NP-hard problem. Given a feasible $\mathbf{X}$, let $I_j(\mathbf{X})$ denote the indices of papers falling in session j. Then,

$$\min_{\mathbf{S},\mathbf{X}} \| \mathbf{P} - \mathbf{S}\mathbf{X} \|_F^2 \iff \min_{\mathbf{X}} \left\{ \min_{\mathbf{S}} \| \mathbf{P} - \mathbf{S}\mathbf{X} \|_F^2 \right\} \iff$$

$$\min_{\mathbf{X}} \sum_{j=1}^{J} \min_{\mathbf{s}_j} \sum_{i \in I_j(\mathbf{X})} \| \mathbf{p}_i - \mathbf{s}_j \|_2^2 .$$

The solution of the inner minimization for $\mathbf{s}_j$ is clearly the mean of those vectors falling in session j. Setting $\mathbf{s}_j$ equal to this mean, i.e., setting $\mathbf{s}_j$ equal to

$$\mathbf{s}_j^\star \triangleq \frac{1}{|I_j(\mathbf{X})|} \sum_{\ell \in I_j(\mathbf{X})} \mathbf{p}_\ell ,$$

it can be easily shown by expanding the squares that

$$\sum_{i \in I_j(\mathbf{X})} \| \mathbf{p}_i - \mathbf{s}_j^\star \|_2^2 = \frac{1}{2\,|I_j(\mathbf{X})|} \sum_{i \in I_j(\mathbf{X})} \sum_{k \in I_j(\mathbf{X})} \| \mathbf{p}_i - \mathbf{p}_k \|_2^2 ,$$

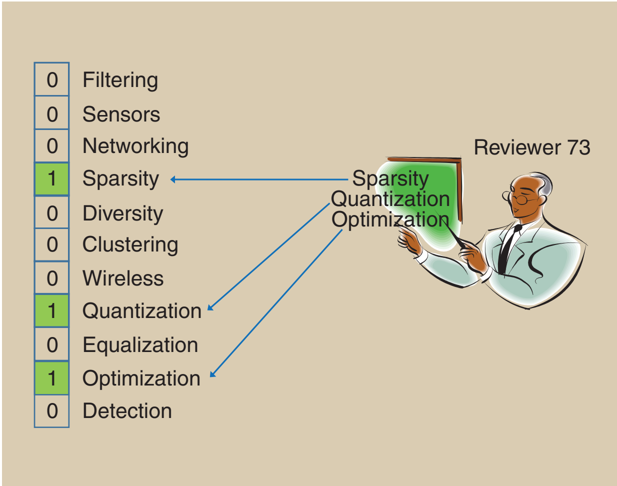
where $|\cdot|$ denotes cardinality. If all session capacities are equal, we may thus use the following criterion instead:

$$\min_{\mathbf{X}} \sum_{j=1}^{J} \sum_{i \in I_j(\mathbf{X})} \sum_{k \in I_j(\mathbf{X})} \| \mathbf{p}_i - \mathbf{p}_k \|_2^2 ,$$

which is to be optimized over $\mathbf{X} \in \mathcal{A}$. This is now what is known as the *minimum k-clustering sum* problem (in our context J plays the role of k), which is in the list of NP-hard problems [28]; see also [29]—The poor TPC chair souls were right all along.

Claim 1

Technical program optimization (paper-to-session assignment, capacitated k-means) is NP-hard.



[FIG3] Representing a reviewer as a point (feature vector) in the same key word space. The feature vector in this particular illustration is Boolean.

The implication is that we cannot expect to solve an arbitrary instance of (2) in complexity polynomial in the number of papers I. It has been shown in [29] (see also [28]) that the minimum k-clustering sum problem can be approximated within a factor of 2—but the algorithm that provides this approximation guarantee has exponential complexity in J. Since J is not small in our context, we will instead explore familiar signal processing tools to obtain conceptually simple and performance-wise satisfactory solutions.

PROPOSED ALGORITHM FOR PAPER-TO-SESSION ASSIGNMENT

The GLM/LBG algorithm is typically used for VQ design. GLM/LBG alternates between optimizing the codebook $\mathbf{S}$ for a given assignment $\mathbf{X}$ and optimizing the assignment $\mathbf{X}$ for a given codebook $\mathbf{S}$. GLM/LBG exploits necessary optimality conditions, implying that $\mathbf{s}_j$ should be the mean of those $\mathbf{p}_i$s assigned to session j, and $\mathbf{p}_i$ should be assigned to the closest $\mathbf{s}_j$; these yield simple conditional updates. The GLM/LBG iteration converges in terms of fit, but the quality of the final solution depends heavily on the initialization.

GLM/LBG cannot be directly applied in our present context because of the presence of the session capacity constraints. In the following, we propose one possible iteration that explicitly takes these constraints into account.

Given a feasible assignment $\mathbf{X}$, the update for $\mathbf{S}$ is simple and, in fact, identical to the corresponding update in GLM/LBG. The step that requires closer scrutiny is the update of $\mathbf{X}$ given $\mathbf{S}$

$$\underset{\mathbf{X}}{\text{minimize}} \; \| \mathbf{P} - \mathbf{S}\mathbf{X} \|_F^2 \tag{6a}$$

$$\text{subject to: } \mathbf{X} \in \mathcal{A}. \tag{6b}$$

Fortunately, it turns out that an optimal point for (6) can be computed easily, without having to search over all feasible assignments $\mathbf{X}$. To explain how this is possible, note first that the objective function in (6a) can be expressed as $\| \mathbf{P} - \mathbf{S}\mathbf{X} \|_F^2 = \| \mathbf{P} \|_F^2 - 2\mathrm{Tr}(\mathbf{P}^T\mathbf{S}\mathbf{X}) + \| \mathbf{S}\mathbf{X} \|_F^2$, and observe that the quadratic term $\| \mathbf{S}\mathbf{X} \|_F^2$

remains constant for any feasible assignment $\mathbf{X}$. This is because of the property in (5), since $\|\mathbf{SX}\|_F^2 = \mathrm{Tr}(\mathbf{X}^T\mathbf{S}^T\mathbf{SX}) = \mathrm{Tr}(\mathbf{XX}^T\mathbf{S}^T\mathbf{S}) = \mathrm{Tr}(\mathbf{\Lambda S}^T\mathbf{S})$. Thus, the conditional update of $\mathbf{X}$ given $\mathbf{S}$ can be done by solving the Boolean LP

$$\underset{\mathbf{X}}{\text{maximize}} \ \mathrm{Tr}(\mathbf{P}^T\mathbf{SX}) \tag{7a}$$

$$\text{subject to: } \mathbf{X} \in \mathcal{A}. \tag{7b}$$

Problem (7) is the so-called semiassignment problem, and there are many efficient algorithms for its solution. For example, the shortest augmenting path algorithm from [30] is applicable, which computes the solution of (7) at complexity $O(JI^2)$.

Although the shortest augmenting path algorithm from [30] is arguably one of the best choices (among the applicable algorithms) for carrying out the $\mathbf{X}$-update, we here also discuss how this can be done using linear programming. We believe that this discussion offers more insights and demonstrates an interesting connection between convex and combinatorial optimization. Observe first that the system of equations in (3b)–(3c) is linear and, therefore, can be written in the form $\mathbf{Gx} = \mathbf{d}$, where $\mathbf{x} \triangleq \mathrm{vec}(\mathbf{X})$. [The operation $\mathrm{vec}(\mathbf{X})$ stacks the columns of the matrix $\mathbf{X}$ into a vector.] Now, the coefficient matrix $\mathbf{G}$ is totally unimodular, i.e., every square submatrix has a determinant of value $0, \pm 1$; and $\mathbf{d}$ is a vector of integers. As a result [31], the polyhedron

$$0 \le \mathbf{X}_{ji} \le 1, \forall i \in \mathcal{I}, j \in \mathcal{J}$$
$$\sum_{j=1}^{J} \mathbf{X}_{ji} = 1, \forall i \in \mathcal{I}$$
$$\sum_{i=1}^{I} \mathbf{X}_{ji} = c_j, \forall j \in \mathcal{J}$$

is the convex hull of all assignments $\mathbf{X} \in \mathcal{A}$. This result implies that the linear programming relaxation

$$\underset{\mathbf{X}}{\text{maximize}} \ \mathrm{Tr}(\mathbf{P}^T\mathbf{SX}) \tag{9a}$$

$$\text{subject to: } 0 \le \mathbf{X}_{ji} \le 1, \forall i \in \mathcal{I}, j \in \mathcal{J} \tag{9b}$$

$$\sum_{j=1}^{J} \mathbf{X}_{ji} = 1, \forall i \in \mathcal{I} \tag{9c}$$

$$\sum_{i=1}^{I} \mathbf{X}_{ji} = c_j, \ \forall j \in \mathcal{J} \tag{9d}$$

is always exact [i.e., problems (7) and (9) are equivalent]. The situation is graphically illustrated in Figure 4, which shows the geometry of (9) in relation to the geometry of (7).

Since (9) is an LP, it follows that either an interior point method or the simplex method can be used for solving (7). When using an interior point method, one should be mindful of cases where there are multiple Boolean solutions with the same (optimal) objective value because the interior point algorithm may converge to the center of a polyhedral facet (instead of a vertex), yielding a noninteger solution. We actually need a basic solution of the LP [32], and advanced interior point LP solvers include means of identifying such a solution, e.g., [33]. These subtleties are avoided altogether if one uses the simplex method or, better yet, the shortest augmenting path algorithm [30], which has favorable low-order polynomial complexity even in the worst case. If only a general interior point LP solver is available, then a random perturbation heuristic can be applied, see [2].

The overall algorithm for (4) is now clear: one starts from a suitable initialization and iterates between updating $\mathbf{S}$ and updating $\mathbf{X}$. For initialization, one can use regular VQ/k-means to come up with an initial $\mathbf{S}$ without regard to capacity constraints. The sessions can be ordered according to population, and excess papers can be moved to the next session in line to produce an initial feasible assignment. Updating can start from $\mathbf{X}$ or from $\mathbf{S}$, and continue as long as the cost is reduced. Finally, initialization does matter (and VQ/k-means is itself sensitive with respect to initialization), so the overall algorithm should be initialized from different starting points 10–30 times to get close to the best possible results. The solution with the smallest cost is then chosen as the final one. At this point, the reader might rightfully wonder how well this algorithm works in practice, compared to expert human assignment. To get a sense of the kind of results that can be expected, see "How Well Does This Work? The ICASSP 2009/SPCOM TC Case Study."

GAUGING THE OPTIMALITY GAP: SEMIDEFINITE RELAXATION

Even though the capacitated k-means clustering problem in (4) is NP-hard, it is possible to efficiently obtain a nontrivial lower bound on its optimal value. Notice that a tight lower bound also serves as a nice exploratory tool, e.g., it can be used to evaluate the performance of the GLM/LBG-based approximation algorithm. In obtaining this lower bound, we first demonstrate that the capacitated k-means clustering problem in (4) can be cast as a QCQP. This is an important link because the literature on quadratic optimization is rich and the tools that have been developed in the field of quadratic optimization are well understood.

In particular, we show that the capacitated k-means clustering problem in (4) can be cast in a form that closely resembles the (in)famous quadratic assignment problem (QAP) [34], [35]. Unlike the classical QAP, however, ours is a semiassignment problem, due to the particular structure of our set of admissible assignment matrices $\mathcal{A}$. Nonetheless, many relaxation strategies that have been developed for the QAP can be applied in our context as well. The best convex relaxations known for QAP are based on SDR. We also apply an SDR method [22], [36]–[39] to our problem. It is worth noting that a different SDR approach to (unconstrained) k-means clustering was pursued in [19].

The main reason why the capacitated k-means clustering problem (4) can be cast as a QCQP is that the optimal $\mathbf{S}^\star$ can be analytically derived as a function of $\mathbf{X}$; that is, the cost function can be concentrated with respect to $\mathbf{S}$ for a given $\mathbf{X}$. There are no constraints on $\mathbf{S}$; therefore, the minimizer is given by $\mathbf{S}^\star = \mathbf{PX}^\dagger$, where $\mathbf{X}^\dagger$ denotes the Moore–Penrose pseudoinverse of $\mathbf{X}$. It

HOW WELL DOES THIS WORK? THE ICASSP 2009/SPCOM TC CASE STUDY

The list of accepted papers from the SPCOM TC track of ICASSP 2009 is used for validation. There were 132 papers accepted, which were to be split among a total of 14 sessions: six lectures and eight poster sessions, containing six and 12 papers each, respectively. The algorithmic results will be compared to the final technical program that was manually produced by Nicholas D. Sidiropoulos, who chaired SPCOM TC at the time.

The list of key words (features) was manually produced by the authors, parsing the list of paper titles. Each title was examined, existing key words were added to the paper as appropriate, and new key words were created and added to list of key words as needed. The final list contains a total of 44 key words:

optimization, cross-layer, networking, resource, QCSI, game, precoding, DSL, distributed, sensor, sparse, MIMO, detection, performance, blind, cognitive, cooperative, capacity, network, coding, security, multiuser, beamforming, downlink, relay, uplink, CDMA, OFDM, synchronization, turbo, quantization, equalization, interference, estimation, training, tracking, localization, consensus, diversity, PAR, STBC, FH, scheduling, communications.

The feature vector of each paper is 44×1, with ones in the positions corresponding to features it possesses, and zeros elsewhere. The median number of (nonzero) features per paper was three.

The computer-generated conference program (using the algorithm in the section "Proposed Algorithm for Paper-to-Session Assignment") for ICASSP 2009/SPCOM TC is listed as Appendix A (available as supplementary material accompanying this article in IEEE *Xplore*). Session pseudotitles were produced by session centroid thresholding. If a key word is included in more than 30% of the papers in a session (the corresponding centroid element is greater than 0.3), then the key word is included in the session pseudotitle. Note that the order of key words in the pseudotitles is arbitrary (one could list them in order of importance, determined by the magnitude of centroid elements). The listed computer-generated program attains a (sum-of-squares) cost of 148.1 (after 30 initializations). The actual technical program that was manually produced by Sidiropoulos attains a cost of 187.25, primarily because, after two days of manual optimization and with a looming deadline ahead, he gave up and used an "umbrella" poster session for papers that did not fit elsewhere but otherwise had little in common. This is avoided in the solution listed in the supplementary material (Appendix A) available in IEEE *Xplore*, and in several other suboptimal solutions, which typically have a few discrepancies but avoid umbrella sessions. Note also that the running time of the algorithm in the section "Proposed Algorithm for Paper-to-Session Assignment" was less than 1.5 minutes (on a Dell E64000 laptop) for this data set, for 30 runs from different initial points.

follows that the conference program optimization problem in (4) can be written equivalently as

$$\underset{\mathbf{X}}{\text{minimize}} \quad \|\mathbf{P} - \mathbf{P}\mathbf{X}^{\dagger}\mathbf{X}\|_F^2 \tag{10a}$$

$$\text{subject to: } \mathbf{X} \in \mathcal{A}. \tag{10b}$$

Since any $\mathbf{X} \in \mathcal{A}$ is full row rank, the pseudoinverse has the form $\mathbf{X}^{\dagger} = \mathbf{X}^{\mathrm{T}}(\mathbf{X}\mathbf{X}^{\mathrm{T}})^{-1}$. Using the property (5), this reduces to the simpler form $\mathbf{X}^{\dagger} = \mathbf{X}^{\mathrm{T}}\boldsymbol{\Lambda}^{-1}$. It follows that (10) is equivalent to the problem

$$\underset{\mathbf{X}}{\text{minimize}} \quad \|\mathbf{P} - \mathbf{P}\mathbf{X}^{\mathrm{T}}\boldsymbol{\Lambda}^{-1}\mathbf{X}\|_F^2 \tag{11a}$$

$$\text{subject to: } \mathbf{X} \in \mathcal{A}. \tag{11b}$$

Expanding the squares in the objective of (11), we get $\|\mathbf{P} - \mathbf{P}\mathbf{X}^{\mathrm{T}}\boldsymbol{\Lambda}^{-1}\mathbf{X}\|_F^2 =$

$$\begin{aligned} &= \|\mathbf{P}\|_F^2 - 2\mathrm{Tr}(\mathbf{P}^{\mathrm{T}}\mathbf{P}\mathbf{X}^{\mathrm{T}}\boldsymbol{\Lambda}^{-1}\mathbf{X}) + \|\mathbf{P}\mathbf{X}^{T}\boldsymbol{\Lambda}^{-1}\mathbf{X}\|_F^2 \\ &= \|\mathbf{P}\|_F^2 - 2\mathrm{Tr}(\mathbf{P}^{\mathrm{T}}\mathbf{P}\mathbf{X}^{\mathrm{T}}\boldsymbol{\Lambda}^{-1}\mathbf{X}) + \|\mathbf{P}\mathbf{X}^{\mathrm{T}}\boldsymbol{\Lambda}^{-1/2}\|_F^2 \\ &= \|\mathbf{P}\|_F^2 - \|\mathbf{P}\mathbf{X}^{\mathrm{T}}\boldsymbol{\Lambda}^{-1/2}\|_F^2, \end{aligned}$$

where we have used $\|\mathbf{Y}\|_F^2 = \mathrm{Tr}(\mathbf{Y}\mathbf{Y}^T)$, so

$$\|\mathbf{P}\mathbf{X}^{\mathrm{T}}\boldsymbol{\Lambda}^{-1}\mathbf{X}\|_F^2 = \mathrm{Tr}(\mathbf{P}\mathbf{X}^{\mathrm{T}}\boldsymbol{\Lambda}^{-1}\mathbf{X}\mathbf{X}^{T}\boldsymbol{\Lambda}^{-1}\mathbf{X}\mathbf{P}^{T}),$$

and, since in this particular case, $\mathbf{X}^{\mathrm{T}}\boldsymbol{\Lambda}^{-1} = \mathbf{X}^{\dagger}$,

$$\begin{aligned} \|\mathbf{P}\mathbf{X}^{\mathrm{T}}\boldsymbol{\Lambda}^{-1}\mathbf{X}\|_F^2 &= \mathrm{Tr}(\mathbf{P}\mathbf{X}^{\mathrm{T}}\boldsymbol{\Lambda}^{-1}\mathbf{X}\mathbf{P}^{T}) \\ &= \mathrm{Tr}(\mathbf{P}\mathbf{X}^{\mathrm{T}}\boldsymbol{\Lambda}^{-1/2}\boldsymbol{\Lambda}^{-1/2}\mathbf{X}\mathbf{P}^{T}) = \|\mathbf{P}\mathbf{X}^{\mathrm{T}}\boldsymbol{\Lambda}^{-1/2}\|_F^2. \end{aligned}$$

Hence, the problem in (11) can be expressed equivalently as

$$\underset{\mathbf{X}}{\text{minimize}} \quad \|\mathbf{P}\|_F^2 - \|\mathbf{P}\mathbf{X}^{\mathrm{T}}\boldsymbol{\Lambda}^{-1/2}\|_F^2 \tag{12a}$$

$$\text{subject to: } \mathbf{X} \in \mathcal{A}. \tag{12b}$$

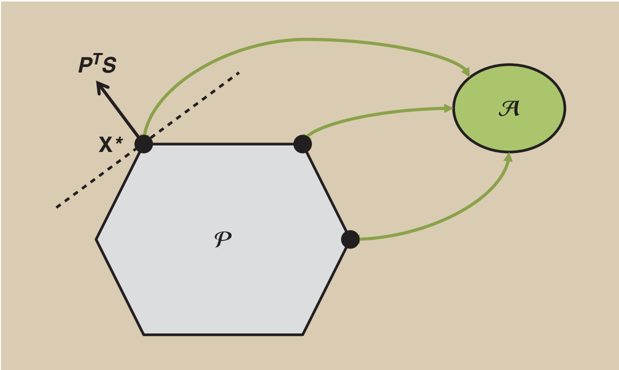
This is now a quadratic minimization problem subject to Boolean constraints, which is intractable [formally, the NP-hardness of (12) follows from its equivalence to (4)]. The form (12) closely resembles the QAP: The difference is that, in (12), $\mathbf{X}$ is constrained to lie in $\mathcal{A}$ instead of the set of permutation matrices, as in the classical QAP.

To illustrate how one can apply SDR to the problem above, we write it first in a more clear form using simple algebraic manipulations. Problem (12) can be written equivalently as

$$\underset{\mathbf{X}}{\text{minimize}} \quad \|\mathbf{P}\|_F^2 - \|(\mathbf{P} \otimes \boldsymbol{\Lambda}^{-1/2})\,\mathrm{vec}(\mathbf{X})\|_2^2 \tag{13a}$$

$$\text{subject to: } \mathbf{X} \in \mathcal{A}, \tag{13b}$$

where $\otimes$ denotes the Kronecker product operation, and vec the operator that stacks the columns of a matrix into one vector. Recall that the linear system of equations (3b) and (3c) can be written in the form $\mathbf{G}\mathbf{x} = \mathbf{d}$, where $\mathbf{x} = \mathrm{vec}(\mathbf{X})$ and define the matrices



[FIG4] The feasible set of (9), which is a polyhedron, is shaded and denoted as $\mathcal{P}$. The objective $\mathrm{Tr}(\mathbf{P}^T\mathbf{S}\mathbf{X})$ is linear, and the point $\mathbf{X}^\star$ is optimal; it is the point in $\mathcal{P}$ as far as possible in the direction $\mathbf{P}^T\mathbf{S}$. As illustrated in the figure, the polyhedron $\mathcal{P}$ is such that its vertices are all points in the set $\mathcal{A}$, and thus Boolean (see the corresponding definition of $\mathcal{A}$ in the text).

$$\mathrm{Q} \triangleq \mathrm{P} \otimes \boldsymbol{\Lambda}^{-1/2}, \text{ and } \mathrm{L} \triangleq \begin{bmatrix} \mathrm{G}^T\mathrm{G} & -\mathrm{G}^T\mathrm{d} \\ -\mathrm{d}^T\mathrm{G} & \mathrm{d}^T\mathrm{d} \end{bmatrix}.$$

With these definitions, (13) can be written equivalently as

$$\underset{\mathrm{x}}{\text{minimize}} \quad \|\mathrm{P}\|_F^2 - \mathrm{x}^T\mathrm{Q}^T\mathrm{Q}\mathrm{x} \tag{14a}$$
$$\text{subject to:} \quad \mathrm{diag}(\mathrm{x}\mathrm{x}^T) = \mathrm{x}, \tag{14b}$$
$$\|\mathrm{G}\mathrm{x} - \mathrm{d}\|_2^2 = 0. \tag{14c}$$

This is now a standard form QCQP, the quadratic constraints in (14b) ensuring that all variables x_i are Boolean. Let us illustrate how one can apply SDR to the above problem step by step.

TECHNICAL DETAILS OF SDR

Using the fact that $\mathrm{x}^T\mathrm{Q}^T\mathrm{Q}\mathrm{x} = \mathrm{Tr}(\mathrm{x}^T\mathrm{Q}^T\mathrm{Q}\mathrm{x}) = \mathrm{Tr}(\mathrm{Q}^T\mathrm{Q}\mathrm{x}\mathrm{x}^T)$ and the change of variables

$$\mathrm{W} = \begin{bmatrix} \mathrm{x}\mathrm{x}^T & \mathrm{x} \\ \mathrm{x}^T & 1 \end{bmatrix} = \begin{bmatrix} \mathrm{W}_{1,1} & \mathrm{W}_{1,2} \\ \mathrm{W}_{1,2}^T & \mathrm{W}_{2,2} \end{bmatrix}, \tag{15}$$

problem (14) is reformulated in a higher dimensional space as follows:

$$\underset{\mathrm{W}}{\text{minimize}} \quad \|\mathrm{P}\|_F^2 - \mathrm{Tr}(\mathrm{W}_{1,1}\mathrm{Q}^T\mathrm{Q}) \tag{16a}$$
$$\text{subject to:} \quad \mathrm{diag}(\mathrm{W}_{1,1}) = \mathrm{W}_{1,2},\ \mathrm{W}_{2,2} = 1, \tag{16b}$$
$$\mathrm{Tr}(\mathrm{L}\mathrm{W}) = 0, \tag{16c}$$
$$\mathrm{W} \succeq 0,\ \mathrm{rank}(\mathrm{W}) = 1. \tag{16d}$$

Here, $\mathrm{W}_{1,1}$ denotes the $JI \times JI$ upper-left block, $\mathrm{W}_{1,2}$ the $JI \times 1$ upper-right block, and $\mathrm{W}_{2,2}$ the 1×1 lower-right block of the $(JI+1) \times (JI+1)$ matrix W. Problem (16) is equivalent to (14), since any rank-1 matrix satisfying (16b) can be factored according to the definition in (15), and, hence, the solution of (14) can be easily constructed from the solution of (16) and vice versa. The only difficult part of (16) is the nonconvex rank-1 constraint on W. Dropping this constraint yields an SDR of (16)

$$\underset{\mathrm{W}}{\text{minimize}} \quad \|\mathrm{P}\|_F^2 - \mathrm{Tr}(\mathrm{W}_{1,1}\mathrm{Q}^T\mathrm{Q}) \tag{17a}$$
$$\text{subject to:} \quad \mathrm{diag}(\mathrm{W}_{1,1}) = \mathrm{W}_{1,2},\ \mathrm{W}_{2,2} = 1, \tag{17b}$$
$$\mathrm{Tr}(\mathrm{L}\mathrm{W}) = 0, \tag{17c}$$
$$\mathrm{W} \succeq 0. \tag{17d}$$

In contrast with (16), problem (17) is convex (in fact, a semidefinite program), and it can be readily solved in polynomial time using efficient interior point methods [18]. If the solution $\mathrm{W}^\star$ of this semidefinite program turns out to have rank 1, then it is a solution for (16) as well. However, because of the relaxation, $\mathrm{W}^\star$ will not always be a rank-1 matrix; hence, the optimal value of (17) generally provides a lower bound on the optimal value of (16) [note that (4) and (16) have the same optimal value].

Given $\mathrm{W}^\star$, an approximate solution for the technical program optimization problem in (4) can be produced using a procedure known as *Gaussian randomization* [37]. This procedure consists of three main steps: 1) draw a random vector $\mathrm{v} = [v_1, \cdots, v_{JI+1}]^T$ from $\mathcal{N}(0, \mathrm{W}^\star)$, 2) form the new vector ξ consisting of the first JI entries of v divided by v_{JI+1}, and 3) find the vector that is closest to ξ and is feasible for (14), i.e., the vector x that minimizes $\|\xi - \mathrm{x}\|_2^2$ subject to (14b)–(14c).

This three-step procedure can be repeated a number of times, and the vector that gives the smallest objective value in (14) can be eventually chosen as an approximate solution. The intuition behind randomization is that it will generate candidate solutions that are close to the eigenvector of $\mathrm{W}^\star$ that corresponds to the largest eigenvalue, but will also take the other eigenvalues into account when these are large enough. Randomization has been widely used in the quadratic optimization literature, and its merits are well documented; see [37, Section IV] for an excellent discussion on this issue.

The rounding problem in step 3) seems hard, but it is not. To explain this, note that for any x feasible for (14), we have that $\|\xi - \mathrm{x}\|_2^2 = \|\xi\|_2^2 - \mathrm{x}^T\xi + I$, and therefore, rounding corresponds to

$$\underset{\mathrm{x}}{\text{maximize}} \quad \mathrm{x}^T\xi \text{ subject to: (14b)–(14c).} \tag{18}$$

Notice that the constraint in (14c) is equivalent to the convex constraint $\mathrm{G}\mathrm{x} = \mathrm{d}$, and, since G is a totally unimodular matrix, problem (18) can be solved efficiently in polynomial time, using, e.g., the shortest augmenting path algorithm from [30]. The same discussion as that for problem (7) applies for (18) as well.

COMPLEXITY CONSIDERATIONS

It is important to recognize that the alternating optimization algorithm in the section "Proposed Algorithm for Paper-to-Session Assignment" is much cheaper and faster than the SDR approach in the section "Gauging the Optimality Gap: Semidefinite Relaxation." This is similar to classical k-means, and it is the reason why alternating optimization is so popular in applications of k-means clustering. For alternating optimization, the conditional update of S is very simple; the most expensive part in every iteration is the conditional update of X. The shortest augmenting path algorithm from [30] can carry out the X-update in time $O(JI^2)$. Linear programming (either with an interior point or with a simplex method) can be effectively used for the X-update as well. In relation to the alternating optimization algorithm of the section "Proposed Algorithm for Paper-to-Session Assignment," the computational disadvantage of SDR in the section "Gauging the Optimality Gap: Semidefinite Relaxation" stems from the fact that it lifts the problem in a higher dimensional space [in (15)–(16)], and this lifting squares the number of variables. This implies much higher complexity. Two important advantages of the SDR approach, on the other hand, are that it yields an approximation in one shot (read: with a predictable number of interior-point iterations for the relaxed convex problem), and it also yields a bound on how far any solution is from an optimum one. The latter is something that cannot be gauged from alternating optimization.

VARIATIONS OF THE BASIC FORMULATION

There are several variations of the basic formulation that one can readily envision. We now briefly mention a few interesting alternatives.

WEIGHTING

In some cases, the TPC chair may wish to highlight emerging or important areas in the technical program. This can be accomplished via feature weighting, i.e., optimizing a weighted least squares cost of the form

$$\|\mathbf{D}(\mathbf{P}-\mathbf{S}\mathbf{X})\|_F^2,$$

where $\mathbf{D}$ is a full-rank diagonal matrix holding the feature weights. Such weighting can be absorbed in $\mathbf{P}$ and $\mathbf{S}$, and, since the latter is unconstrained, it does not change the essence of the proposed solutions. It is clear that the proposed GLM/LBG algorithm can be readily modified to handle this extension. Following steps similar to (10)–(12), it is a simple exercise to verify that the SDR approach can be extended as well.

ALIGNMENT WITH ORGANIZATIONAL STRUCTURE

Organizations such as the NSF often prefer to form panels that reflect their organizational structure. For example, for a large cross-disciplinary solicitation that falls under the auspices of multiple divisions (sometimes even across directorates), from a logistics point of view, it makes a lot of sense to produce panels that are reasonably well aligned with the constituent programs. This can be accomplished by anchoring panel centroids in $\mathbf{S}$ not to deviate too far from the constituent organizational unit profiles, stored in $\mathbf{S}_o$, i.e., by augmenting the cost function in (4) with a penalty term as

$$\|\mathbf{P}-\mathbf{S}\mathbf{X}\|_F^2+\rho\|\mathbf{S}_o-\mathbf{S}\|_F^2.$$

By varying the penalty parameter $\rho > 0$, one can trade off between alignment and homogeneity. Notice that this augmentation does not fundamentally change the nature of our solutions. In fact, the optimal session centroid matrix $\mathbf{S}^*$ is still given in simple closed form as $\mathbf{S}^* = (\mathbf{P}\mathbf{X}^{\mathrm{T}}+\rho\mathbf{S}_o)(\boldsymbol{\Lambda}+\rho\mathbf{I})^{-1}$. As a result, both the alternating optimization algorithm and the proposed SDR approach can be easily modified to account for this penalty term.

DIVIDE-AND-CONQUER AND TREE-STRUCTURED VQ

For the special case where we are interested in splitting the papers into just $J=2$ sessions, the conditional update of $\mathbf{X}=[\mathbf{x}_1\mathbf{x}_2]^{\mathrm{T}} \in \mathbb{R}^{2\times I}$ given $\mathbf{S}=[\mathbf{s}_1\mathbf{s}_2]\in\mathbb{R}^{N\times 2}$ takes a very simple form. This simplification can be used to construct a divide-and-conquer algorithm for paper-to-session assignment, reminiscent of hierarchical clustering approaches and tree-structured VQ [20], [21]. Consider the conditional paper-to-session assignment problem for $J=2$ sessions only. Using the equivalence shown in (6) and (7), the optimization problem is

$$\underset{\mathbf{x}_1,\mathbf{x}_2}{\text{maximize}}\ \operatorname{Tr}(\mathbf{P}^{\mathrm{T}}[\mathbf{s}_1\mathbf{s}_2][\mathbf{x}_1\mathbf{x}_2]^T), \tag{19a}$$

$$\text{subject to: } \mathbf{x}_1(i)\in\{0,1\},\ \mathbf{x}_2(i)\in\{0,1\},\ \forall i\in\mathcal{I}, \tag{19b}$$

$$\mathbf{x}_1(i)+\mathbf{x}_2(i)=1,\ \forall i\in\mathcal{I}, \tag{19c}$$

$$\sum_{i=1}^{I}\mathbf{x}_1(i)=c_1,\ \sum_{i=1}^{I}\mathbf{x}_2(i)=c_2=I-c_1. \tag{19d}$$

Using the constraints (19b)–(19d) and the fact that $\operatorname{Tr}(\mathbf{AB})=\operatorname{Tr}(\mathbf{BA})$, one can eliminate variable $\mathbf{x}_2$ from (19), yielding the simpler problem

$$\begin{aligned}&\underset{\mathbf{x}_1}{\text{maximize}}\ \mathbf{x}_1^{\mathrm{T}}\mathbf{P}^{\mathrm{T}}(\mathbf{s}_1-\mathbf{s}_2)\\&\text{subject to: } \mathbf{x}_1(i)\in\{0,1\},\ \forall i\in\mathcal{I},\\&\qquad\sum_{i=1}^{I}\mathbf{x}_1(i)=c_1,\end{aligned}$$

from which it is clear that the optimal solution is to allocate the c_1 units to the c_1 largest elements of $\mathbf{P}^{\mathrm{T}}(\mathbf{s}_1-\mathbf{s}_2)$. These can be found using a sorting operation, at complexity $O(I\log I)$, or by direct parsing at $O(Ic_1)$.

Now, using the above result for $J=2$, we can construct a potentially appealing divide-and-conquer solution for the paper-to-session assignment problem for $J>2$ as follows: We start with regular VQ/k-means to produce an initial centroid matrix $\mathbf{S}$, the columns of which are then ordered according to paper population. In the divide step, we first process the columns of $\mathbf{S}$ (e.g., using plain 2-means) to produce two new (super)centroids, then use the sorting-based algorithm to assign papers to these two centroids in a way that respects the session capacity constraints. We then recursively refine and conquer the subproblems in a similar manner. Once we produce the final assignment, we update $\mathbf{S}$ and repeat the procedure. This algorithm is fast and can be quite effective, mainly depending on the quality of the initialization point.

THE REVIEW ASSIGNMENT PROBLEM

The review assignment stage is even more difficult than putting together the final technical program simply because it involves (a lot) more papers and every paper must be reviewed by more than one reviewer. Suppose that I papers are to be assigned for review to (at most) J reviewers. Reviewer j has a fixed vector profile $\mathbf{s}_j$ representing the reviewer's expertise and reviewing interests, and a prenegotiated reviewing capacity r_j. Every paper should be reviewed by, say, three reviewers. Our goal here is to minimize the paper-to-reviewer mismatches, i.e., a paper should be assigned for review to three reviewers whose individual vector profiles cover as much as possible the paper profile $\mathbf{p}_i$. At the same time, and of equal importance, is that the reviewer profiles should collectively cover the paper profile $\mathbf{p}_i$ as much as possible.

One can thus pose the review assignment problem as follows:

$$\underset{\mathbf{X}\in\{0,1\}^{J\times 3I}}{\text{minimize}} \quad (1-\lambda)\sum_{i=1}^{3I}\mathbf{1}^T\left(\mathbf{P}_{*\lceil \frac{i}{3}\rceil}-\mathbf{S}\mathbf{X}_{*i}\right)_+$$

$$+\lambda\sum_{i=1}^{I}\mathbf{1}^T[\mathbf{P}_{*i}-\mathbf{S}(\mathbf{X}_{*3i-2}+\mathbf{X}_{*3i-1}+\mathbf{X}_{*3i})]_+, \qquad (21a)$$

$$\text{subject to:} \quad \mathbf{X}_{\text{ij}}\in\{0,1\},\ \forall i,j, \qquad (21b)$$

$$\sum_{j=1}^{J}\mathbf{X}_{\text{ji}}=1,\ \forall i\in\{1,\dots,3I\}, \qquad (21c)$$

$$\sum_{i=1}^{3I}\mathbf{X}_{\text{ji}}\le r_j,\ \forall j\in\{1,\dots,J\}, \qquad (21d)$$

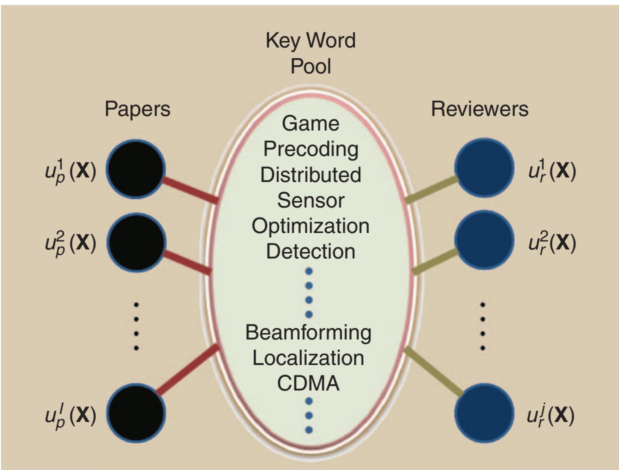
$$\sum_{i=3k+1}^{3k+3}\mathbf{X}_{\text{ji}}\le 1,\ \forall j\in\{1,\dots,J\},$$

$$\forall k\in\{0,\cdots,I-1\}, \qquad (21e)$$

$$\mathbf{X}_{\text{ij}}=0 \quad \forall (i,j)\in\mathcal{COI}. \qquad (21f)$$

Here, $\mathbf{P}$ is the matrix of paper profiles, $\mathbf{S}=[\mathbf{s}_1,\cdots,\mathbf{s}_J]\in\mathbb{R}^{N\times J}$ is the matrix of the reviewer profiles, and $\mathbf{X}_{*i}$ denotes the ith column of $\mathbf{X}$. The symbol $(\cdot)_+$ denotes projection to the nonnegative orthant, $\lceil\cdot\rceil$ denotes the ceiling function, and $\mathbf{1}$ denotes the $N\times 1$ vector of all ones.

Let us now explain the mathematical formulation of the review assignment problem in detail. Observe that the cost function in (21a) comprises two sums. The first aims to minimize the paper key words not covered by the associated reviewers individually, while the second (the collective span term) accounts for the paper key words that are not covered by the sum of profiles of the associated reviewers. The two cost factors are weighted using a suitable regularization parameter $0<\lambda<1$.



[FIG5] An illustration of the "dual" nature of the problem in terms of utility functions. Each paper has as utility its key word coverage (collectively, from all assigned reviewers), and each reviewer has as a utility the aggregate amount of key words matched from his/her assigned papers.

The inequality constraints in (21e) protect each paper from being assigned to the same reviewer twice, while the constraints (21c) and (21d) ensure that each paper $\mathbf{P}_{*i}$ will be assigned for review to three reviewers, while respecting the reviewer capacity constraints. In particular, columns $3i-2, 3i-1, 3i$ in $\mathbf{X}\in\{0,1\}^{J\times 3I}$ comprise Boolean variables, which select three different reviewers for paper i [see (21c) and (21e)]. Moreover, the ceiling operation $\lceil i/3\rceil$ repeats three times the ith column of $\mathbf{P}$ (paper i) to calculate its mismatch with each of the three individual assigned reviewers [see (21a)].

Finally, note that reviewers should not have a conflict of interest (COI) with the papers they are reviewing (e.g., they cannot be from the same department as any of the paper's authors). In case there is a COI between a reviewer and specific papers, additional COI constraints must be included in the optimization. These are taken into account by the constraint in (21f), which enforces the pertinent assignment variables to be equal to zero.

The review assignment problem as posed in (21) is combinatorial, but it has a convex objective function, and also the constraints in (21c)–(21f) are convex constraints. Interestingly, replacing the Boolean constraints in (21b) by the convex inequality constraints $0\le\mathbf{X}_{\text{ji}}\le 1$ leads to a relaxation problem whose feasible set is a polyhedron with Boolean vertices only (we shall call this the *review assignment polyhedron*). This can be seen by noting that the coefficient matrix of the set of linear inequalities (21c)–(21f) is totally unimodular (see, e.g., [31]). Even so, problem (21) is difficult to solve due to the collective span term in the objective, which is a nonlinear function of $\mathbf{X}$. One can construct, however, an approximate solution through convex relaxation and rounding.

Before we explain this approach in detail, let us first discuss several interesting points that can be gauged from the problem formulation in (21). To simplify exposition and better highlight these points, we temporarily confine attention to the case of Boolean matrices $\mathbf{P}$ and $\mathbf{S}$. We emphasize, however, that the convex relaxation approach that we propose for (21) holds for general matrices $\mathbf{P}$ and $\mathbf{S}$.

REMARK 2: PAPER AND REVIEWER UTILITY FUNCTIONS

One may think of the review assignment problem in terms of utility functions. To see this, it is convenient to introduce some mathematical notation first. Suppose that both $\mathbf{P}$ and $\mathbf{S}$ are Boolean. Moreover, suppose that assignment $\mathbf{X}$ assigns paper $\mathbf{p}_i$ to the reviewer set $\mathcal{R}_i(\mathbf{X})$ (with $|\mathcal{R}_i(\mathbf{X})|=3$) and the same assignment $\mathbf{X}$ assigns to reviewer j the paper set $\mathcal{N}_j$ (with $|\mathcal{N}_j(\mathbf{X})|\le r_j$). Let $u_p^i(\mathbf{X})=\mathbf{1}^T\mathbf{p}_i-\mathbf{1}^T\left(\mathbf{p}_i-\sum_{k\in\mathcal{R}_i(\mathbf{X})}\mathbf{s}_k\right)_+$ be the utility function of paper i (in case of Boolean $\mathbf{P}$ and $\mathbf{S}$ this is paper i's collective key word coverage resulting from assignment $\mathbf{X}$), and let $u_r^j(\mathbf{X})=\sum_{k\in\mathcal{N}_j(\mathbf{X})}[\mathbf{1}^T\mathbf{p}_k-\mathbf{1}^T(\mathbf{p}_k-\mathbf{s}_j)_+]$ be the utility function of reviewer j (in case of Boolean $\mathbf{P}$ and $\mathbf{S}$, this is the total number of key word matches between the reviewer and all papers assigned to the reviewer). Maximizing reviewer satisfaction and paper utility can be conflicting objectives, as illustrated in Figure 5 and exemplified in Figure 6. The tradeoff between the two is captured in the problem formulation (21) because the objective function in (21a) can be written in terms of the $\{u_p^i(\mathbf{X})\}_{i=1}^I$ and $\{u_r^j(\mathbf{X})\}_{j=1}^J$, by regrouping terms accordingly. ■

REMARK 3

Observe that for Boolean matrices $\mathbf{P}$ and $\mathbf{S}$ the first sum term in (21a) can be replaced by a function linear in $\mathbf{X}$ since, for any feasible assignment $\mathbf{X}$ and Boolean matrices $\mathbf{P}$ and $\mathbf{S}$, it holds that $\sum_{i=1}^{3I} \mathbf{1}^T(\mathbf{P}_{*\lceil(i/3)\rceil} - \mathbf{S}\mathbf{X}_{*i})_+ = \sum_{i=1}^{3I} \mathbf{1}^T\mathbf{P}_{*\lceil(i/3)\rceil} - \sum_{i=1}^{3I} \mathbf{P}^T_{*\lceil(i/3)\rceil}\ \mathbf{S}\mathbf{X}_{*i}$. In other words, this sum attempts to maximize the total affinity between papers and reviewers, which is reminiscent of the approach followed in [2]. ■

Let us now turn the discussion to general $\mathbf{P}$ and $\mathbf{S}$, and describe explicitly the convex relaxation approach for (21). Let $\mathbf{X}^*$ denote the solution to the relaxation program where the Boolean constraints $\mathbf{X}_{ji} \in \{0,1\}$ are replaced by the interval ones, $0 \le \mathbf{X}_{ji} \le 1$. This relaxation yields a convex problem, which can be reformulated as an LP and solved efficiently. To see this, introduce for every individual summand in (21a) an associated slack variable t_i, and note that $\max(x,0) \le t_i \iff x \le t_i$ for $t_i \ge 0$. The constraint $\max(x,0) \le t_i$ will always be satisfied with equality at the optimum, which yields the LP reformulation.

Unfortunately, however, $\mathbf{X}^*$ is not guaranteed to be Boolean (the LP emerging after introducing the slack variables is not guaranteed to be totally unimodular); therefore, we need a way of converting the solution of the relaxed program into a good admissible solution for (21). This can be done by finding an assignment $\mathbf{X}$, which is as close as possible (in a Euclidean sense) to $\mathbf{X}^*$, i.e., by finding an $\mathbf{X}$ that minimizes $\|\mathbf{X} - \mathbf{X}^*\|_F^2$ subject to (21b)–(21f). This rounding problem seems hard, but it is not. To explain this, note that for any assignment $\mathbf{X}$ feasible for (21), we have that $\|\mathbf{X} - \mathbf{X}^*\|_F^2 = \|\mathbf{X}^*\|_F^2 - \mathrm{Tr}(\mathbf{X}^T\mathbf{X}^*) + 3I$, and, therefore, rounding corresponds to

$$\text{maximize } \mathrm{Tr}(\mathbf{X}^T\mathbf{X}^*) \text{ subject to (21b)–(21f).}$$

The above problem is equivalent to its linear programming relaxation (and is therefore easy to solve), since the polyhedron arising from the relaxation has only Boolean vertices [which are precisely the feasible set (21b)–(21f)]. To appreciate how well the proposed review assignment method works, see "How Well Does Automated Review Assignment Work? A SPAWC 2010 Case Study" and "Quantitative Assessment of Review Assignment Quality."

SOME VARIATIONS OF THE BASIC FORMULATION

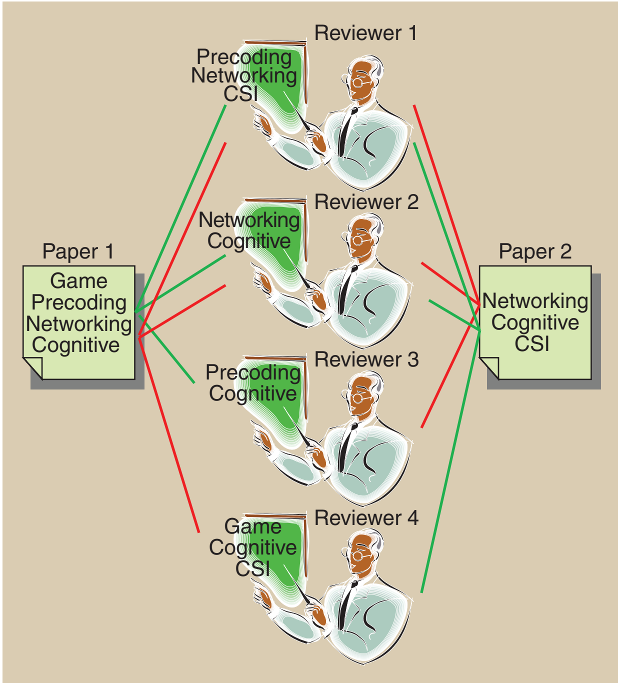
ALTERNATIVE COST FUNCTIONS

For simplicity, we use the sum of inconsistencies in the cost of our formulation in (21). An interesting alternative would be to employ the sum of squares of inconsistencies, essentially putting more emphasis (and penalizing more) the bad assignments. Note that using the sum of squares of inconsistencies would still lead to a convex cost function.

CONTROLLING THE WORST MATCHING

It is possible to design the assignment while explicitly imposing an upper bound T^* on the cost of the worst paper-reviewer matching

$$\mathbf{1}^T[\mathbf{P}_{*i} - \mathbf{S}(\mathbf{X}_{*3i-2} + \mathbf{X}_{*3i-1} + \mathbf{X}_{*3i})]_+ \le T^*, \ \forall i,$$



[FIG6] A green assignment: paper #1 utility = 3 + paper #2 utility = 3 ⇒ total paper utility = 6; reviewer utilities 2+2+2 (for paper #1) + 2+2+2 (for paper #2) ⇒ total reviewer utility = 12. Red assignment: paper #1 utility = 4 + paper #2 utility = 3 ⇒ total paper utility = 7; reviewer utilities 2+2+2 (for paper #1) + 2+2+1 (for paper #2) ⇒ total reviewer utility = 11. So, the green assignment is better in terms of reviewer utility, but the red one is better in terms of paper utility. This explains why the two objectives can be conflicting.

in addition to (21b)–(21e). This imposes a stricter requirement but changes the nature of the feasible set, as for general (even Boolean) $\mathbf{P}$ and $\mathbf{S}$, the new polyhedron is not guaranteed to have only Boolean vertices.

A more flexible approach to this issue is to consider varying λ in the cost function of (21) to trade off reviewer satisfaction for paper key word coverage. One can easily check the quality of a particular assignment after the optimization, by producing statistics, most notably how many key words of each paper have been collectively covered by its respective reviewers. If the result is not satisfactory, one can resolve the problem by changing λ so as to strike a more appropriate tradeoff. In fact, one can associate a different parameter $\lambda_i > 0$ to each paper i, if that is desired.

CONCLUSIONS

WHAT WE LEARNED

By viewing papers as vectors in a suitable feature space, the loosely defined tasks of paper-to-session and paper-to-reviewer assignment have been formulated as optimization problems that are strikingly familiar in many ways. The core problem underlying paper-to-session assignment is capacitated k-means, i.e., clustering under capacity constraints, and is NP-hard. For paper-to-reviewer assignment, it was shown that ensuring scientifically sound reviews (each aspect of each paper covered by at least one assigned reviewer) and

HOW WELL DOES AUTOMATED REVIEW ASSIGNMENT WORK? A SPAWC 2010 CASE STUDY

The submitted paper list and reviewing pool of SPAWC 2010 was used for validation. There were 203 submitted papers, and the reviewing pool comprised 64 reviewers (20+2+42 reviewers of capacity 8|15|16 papers, respectively). The list of key words (features) was manually produced by the authors by updating the previous list for ICASSP 2009; the final SPAWC key word list contained a total of 50 key words:

beamforming, blind, capacity, CDMA, classification, coding, cognitive, consensus, cooperative, cross-layer, detection, distributed, diversity, downlink, UWB, DSL, equalization, estimation, feedback, FH, game, joint source-channel, localization, MIMO, multiuser, network coding, networking, OFDM, optimization, par, performance, QCSI, quantization, random matrix, relay, resource, RFID, scheduling, security, sensor, sparse, speech-image, STBC, synchronization, time-varying, tracking, training, turbo, underwater, uplink.

The feature vector of each paper and each reviewer is 50×1, with ones in the positions corresponding to features it possesses, and zeros elsewhere. Feature vectors for the reviewers were created by Nicholas D. Sidiropoulos (acting as TPC chair), using his knowledge of their expertise. Feature vectors for the papers were partially entered by the respective authors, using a separate key word-clicking system that was set up for this purpose; however, not all authors obliged, so features for papers were also entered by Sidiropoulos after looking at paper titles. Parameter λ in the algorithm was set to $\lambda = 0.5$. It is worth mentioning that the ratio between the objective value of the linear programming relaxation and that of the rounded final solution was 98.7% [hinting that the final assignment was (at least) close to the optimal one]. The running time of the algorithm (relaxation + rounding) was less than two minutes for this data set, on a Dell E64000 laptop. The computer-generated review assignment is listed as Appendix B in the supplementary document accompanying this article in IEEE *Xplore*. Perusing this assignment, one can observe that four out of 64 reviewers were not assigned any paper at all in the final solution. In the cases where we have spare total reviewing capacity, we may consider adding a penalty term to avoid fully loading some reviewers and idling others.

maximizing reviewer satisfaction can be (and often are) conflicting objectives that must be traded against each other. The resulting paper-to-reviewer assignment problem is generally hard (albeit reducing it to a known NP-hard problem is not straightforward). Still, it was shown that it is possible to generate good suboptimal solutions using familiar signal processing tools. While there is certainly a lot more work to be done (e.g., automatic key word retrieval and paper mark-up, exploration of alternative problem formulations), our results indicate that computer-generated technical programs outperform expert manual work at a fraction of the time and with very limited input by the chair.

WHY IT IS IMPORTANT?

If you are a TPC chair, spend some time to come up with the right set of key words that capture what is happening in your area, invite enough good reviewers (a margin of 20% more reviewing capacity is always helpful, so do secure a few more reviewers; if you do not need all that reviewing power, reduce everyone's quota—they will be thankful). We have tested our algorithms with actual conference data, producing review and program assignments that TPC chairs have found very useful. We will make our algorithms freely available to the research community at the time of publication of this article.

As a final note, one can envision many other interesting applications of clustering under capacity constraints:

- assigning students to classrooms or study groups according to educational background, level of accomplishment in math/science/language, interests, etc.
- production-line packaging according to product quality features (e.g., tolerances)
- design of stock performance indices based on market sector, segment, capitalization, exposure to commodity price fluctuations, etc.

QUANTITATIVE ASSESSMENT OF REVIEW ASSIGNMENT QUALITY

We now discuss various performance metrics and statistics to appreciate the quality of the computer-generated solution.

DEFINITION: We define the quality index (QI) of a particular reviewer, as the average percentage of key word matches between the reviewer's profile and his/her assigned papers. As an example, suppose that a certain reviewer is assigned two papers for review, the papers having five and six key words, respectively, and let us assume that there are two key word matches from the first paper and three matches from the second. The reviewer's QI is then calculated as the average $((2/5 + 3/6)/2) \times 100\% = 45\%$.

The reviewers' QIs for the SPAWC 2010 case study can be found in the supplementary material (Appendix B), together with the optimized assignment. One can observe that 39/60 utilized reviewers had a QI above 80%, 54/60 reviewers had a QI above 70%, and all 60 utilized reviewers had a QI above 40%. From the collective span point of view, note that 187/203 papers ($\approx 92\%$) were fully covered (collectively) by their respective reviewers; the few papers that were not fully covered are marked with an asterisk in Appendix B.

As a final measure of the quality of the overall assignment, we compute the percentage of the overall key word matches, i.e., the total number of paper key words covered collectively by all assigned reviews. The percentage ratio (covered key words/total key words) was 98.1% for the SPAWC 2010 computerized assignment, indicating the high quality of the solution.

FURTHER INFORMATION

This article has supplementary downloadable material available in IEEE *Xplore*; see http://ieeexplore.ieee.org. The material includes a computer-generated conference program and a computer-generated review assignment using the methods presented in this article. Contact nicos@umn.edu for further questions regarding this work. In addition, a companion Web site is under development, and a link will be posted at http://www.ece.umn.edu/~nikos/.

AUTHORS

Nicholas D. Sidiropoulos (nikos@umn.edu) received his diploma in electrical engineering from the Aristotelian University of Thessaloniki, Greece, and his M.S. and Ph.D. degrees in electrical engineering from the University of Maryland in 1988, 1990, and 1992, respectively. He has been a professor at the University of Minnesota since 2011. His current research focuses primarily on signal and tensor analytics with applications in cognitive radio and big data. He received the National Science Foundation/CAREER Award in 1998 and the IEEE Signal Processing Society (SPS) Best Paper Award in 2001, 2007, and 2011. He was an IEEE SPS Distinguished Lecturer (2008–2009) and chair of the IEEE Signal Processing for Communications and Networking Technical Committee (2007–2008). He was an associate editor of *IEEE Transactions on Signal Processing* (2000–2006), *IEEE Signal Processing Letters* (2000–2002), and on the editorial board of *IEEE Signal Processing Magazine* (2009–2011). He has been an area editor of *IEEE Transactions on Signal Processing* since 2012. He received the 2010 IEEE SPS Meritorious Service Award. He is a Fellow of the IEEE.

Efthymios E. Tsakonas (tsakonas@kth.se) received his diploma degree in electronics and computer engineering from the Technical University of Crete, Greece, 2008. Since September 2009, he has been a member of the Signal Processing Laboratory, Royal Institute of Technology (KTH), Stockholm, Sweden, where he earned his Ph.D. degree in 2014. His research interests are in convex optimization and approximation theory and algorithms, with applications in signal processing, communications, and preference measurement.

REFERENCES

[1] D. Hartvigsen, J. Wei, and R. Czuchlewski, "The conference paper-reviewer assignment problem," *Decision Sci.*, vol. 30, no. 3, pp. 865–876, 1999.

[2] C. Taylor, "On the optimal assignment of conference papers to reviewers," Tech. Rep. MS-CIS-08-30, Dept. Comput. Inform. Sci., Univ. Pennsylvania, Dec. 2008.

[3] S. Hettich and M. Pazzani, "Mining for proposal reviewers: Lessons learned at the National Science Foundation," in *Proc. 12th Association for Computing Machinery's Special Interest Group on Knowledge Discovery and Data Mining (ACM SIGKDD)*, 2006, pp. 862–871.

[4] M. Karimzadehgan, C. Zhai, and G. Belford, "Multi-aspect expertise matching for review assignment," in *Proc. 17th ACM Conf. Information and Knowledge Management (CIKM 2008)*, Napa Valley, CA, pp. 1113–1122.

[5] S. Theodoridis and K. Koutroumbas, *Pattern Recognition,* 4th Ed. New York: Academic Press, 2008.

[6] C. Bishop, *Pattern Recognition and Machine Learning*. New York: Springer, 2006.

[7] R. Baeza-Yates and B. Ribeiro-Neto, *Modern Information Retrieval*. Boston, MA: ACM Press and Addison-Wesley, 1999.

[8] E. Papalexakis, N. D. Sidiropoulos, and M. Garofalakis, "Reviewer profiling using sparse matrix regression," in *Proc. 2010 IEEE International Conf. Data Mining (ICDM) Workshops/Workshop on Optimization Based Methods for Emerging Data Mining Problems (OEDM)*, Sydney, Australia, Dec. 2010, pp. 1214–1219.

[9] S. Dumais and J. Nielsen, "Automating the assignments of submitted manuscripts to reviewers," in *Proc. Association for Computing Machinery Special Interest Group on Information Retrieval*, 1992, pp. 233–244.

[10] F. Wang, B. Chen, and Z. Miao, "A survey on reviewer assignment problem," *Lect. Notes Comput. Sci.,* vol. 5027, pp. 718–727, 2008.

[11] S. Janak, M. Taylor, C. Floudas, M. Burka, and T. Mountziaris. (2006). A novel and effective integer optimization approach for the NSF panel assignment problem: A multi-resource and preference-constrained generalized assignment problem. *Ind. Eng. Chem*. Res. 45, pp. 258–265. [Online]. Available: http://ares.princeton.edu/casl_nsf_mgap/login.php

[12] P. Hahn, B. Kim, M. Guignard, J. Smith, and Y. Zhu, "An algorithm for the generalized quadratic assignment problem," *Comput. Optim. Appl.* (Springer), vol. 40, no. 3, pp. 351–372, July 2008.

[13] R. Burkard, M. Dell'Amico, and S. Martello, *Assignment Problems*. Philadelphia, PA: SIAM, 2009.

[14] A. Jain, M. Murty, and P. Flynn, "Data clustering: A review," *ACM Comput. Surv.*, vol. 31, no. 3, pp. 264–323, Sept. 1999.

[15] K. Wagstaff, C. Cardie, S. Rogers, and S. Schroedl, "Constrained k-means clustering with background knowledge," in *Proc. 18th Int. Conf. Machine Learning*, 2001, pp. 577–584.

[16] I. Davidson and S. Ravi, "Clustering with constraints: Feasibility issues and the k-means algorithm," in *Proc. 2005 SIAM Int. Conf. Data Mining*.

[17] P. Bradley, K. Bennett, and A. Demiriz, "Constrained k-means clustering," Tech. Rep. TR-2000-65, Microsoft Research, 2000.

[18] S. Boyd and L. Vandenberghe. (2004). *Convex Optimization*. [Online]. Cambridge, U.K.: Cambridge Univ. Press. Available: http://www.stanford.edu/~boyd/cvxbook.html

[19] J. Peng and Y. Wei, "Approximating k-means-type clustering via semidefinite programming," *SIAM J. Optim.*, vol. 18, no. 1, pp. 186–205, 2007.

[20] K. Jain and R. Dubes, *Algorithms for Clustering Data*. Englewood Cliffs, NJ: Prentice Hall, 1988.

[21] L. Kaufman and P. Rousseeuw, *Finding Groups in Data, an Introduction to Cluster Analysis*. New York: Wiley, 1990.

[22] H. Wolkowicz, "Relaxations of Q2P," in *Handbook of Semidefinite Programming: Theory, Algorithms, and Applications*, H. Wolkowicz, R. Saigal, and L. Vandenberghe, Eds. Norwell, MA: Kluwer, 2000, Ch. 13.4, pp. 395–419.

[23] S. P. Lloyd, "Least squares quantization in PCM," *IEEE Trans. Inform. Theory*, vol. 28, pp. 129–137, Mar. 1982.

[24] J. Max, "Quantizing for minimum distortion," *IRE Trans. Inform. Theory*, vol. 6, pp. 7–12, Mar. 1960.

[25] Y. Linde, A. Buzo, and R. Gray, "An algorithm for vector quantizer design," *IEEE Trans. Commun.*, vol. 28, no. 1, pp. 84–95, Jan. 1980.

[26] D. Aloise, A. Deshpande, P. Hansen, and P. Popat, "NP-hardness of Euclidean sum-of-squares clustering," *Mach. Learn.*, vol. 75, pp. 245–248, May 2009.

[27] S. Dasgupta and Y. Freund, "Random projection trees for vector quantization," *IEEE Trans. Inform. Theory*, vol. 55, pp. 3229–3242, July 2009.

[28] P. Crescenzi and V. Kann. A compendium of NP optimization problems. [Online]. Available: http://www.nada.kth.se/~viggo.

[29] N. Guttmann-Beck and R. Hassin, "Approximation algorithms for minimum sum p-clustering," *Disc. Appl. Math.* vol. 89, nos. 1–3, pp. 125–142, Dec. 1998.

[30] J. Kennington and Z. Wang, "A shortest augmenting path algorithm for the semi-assignment problem," *Operations Res.*, vol. 40, no. 1, pp. 178–187, Feb. 1992.

[31] G. Birkhoff, "Tres observaciones sobre el algebra lineal," *Universidad Nacional de Tucuman Revista*, vol. 5, pp. 147–151, 1946.

[32] D. Bertsimas and J. Tsitsiklis, *Introduction to Linear Optimization*. Belmont, MA: Athena Scientific, 1997.

[33] MOSEK ApS. The MOSEK optimization toolbox for MATLAB manual. [Online]. Available: http://docs.mosek.com/7.0/toolbox.pdf

[34] Q. Zhao, S. Karisch, F. Rendl, and H. Wolkowicz, "Semidefinite programming relaxations for the quadratic assignment problem," *J. Combinatorial Optim.*, vol. 2, no. 1, pp. 71–109, 1998.

[35] J. Povh and F. Rendl, "Copositive and semidefinite relaxations of the quadratic assignment problem," *Discrete Optim.*, vol. 6, no. 3, pp. 231–241, Aug. 2009.

[36] M. Goemans and D. Williamson, "Improved approximation algorithms for maximum cut and satisfiability problem using semidefinite programming," *J. ACM,* vol. 42, no. 6, pp. 1115–1145, Nov. 1995.

[37] Z. Q. Luo, W. K. Ma, A. M. C. So, Y. Ye, and S. Zhang, "Semidefinite relaxation of quadratic optimization problems," *IEEE Signal Processing Mag.*, vol. 27, no. 3, pp. 20–34, May 2010.

[38] P. Tseng, "Further results on approximating nonconvex quadratic optimization by semidefinite programming relaxation," *SIAM J. Optim.*, vol. 14, no. 1, pp. 268–283, July 2003.

[39] S. Zhang, "Quadratic maximization and semidefinite relaxation," *Math. Program.*, vol. 87, no. 3, pp. 453–465, May 2000.

[40] A. Banerjee, S. Merugu, I. S. Dhillon, and J. Ghosh, "Clustering with Bregman divergences," *J. Mach. Learn. Res.*, vol. 6, pp. 1705–1749, 2005.

[SP]