DIMACS Technical Report 2004-12
June 2004

How to Increase the Acceptance Ratios of Top Conferences?

by

Graham Cormode[1,2] Artur Czumaj[3] S. Muthukrishnan[4,5]

[1]DIMACS Postdoctoral Fellow
[2]graham@dimacs.rutgers.edu Center for Discrete Mathematics and Computer Science (DIMACS) Rutgers University, Piscataway NJ. Supported by NSF ITR 0220280 and NSF EIA 02-05116.
[3]czumaj@oak.njit.edu Computer Science Department, New Jersey Institute of Technology, Newark, NJ USA
[4]DIMACS Permanent Member
[5]muthu@cs.rutgers.edu Division of Computer and Information Systems, Rutgers University Supported by NSF CCR 0087022, NSF ITR 0220280 and NSF EIA 02-05116.

ABSTRACT

In the beginning was the pub. This work was triggered by a pub conversation where the authors observed that many resumes list acceptance ratios of conferences where their papers appear, boasting the low acceptance ratio. The lower the ratio, the better your paper looks. The list might look equally impressive if one listed the rejection ratio of conferences where one's paper was submitted and rejected. We decided to lampoon rather than lament the effort the PC typically put in: wouldn't the world be better if we could encourage only high quality submissions and so run top conferences with very high acceptance ratios? This paper captures our thoughts, and it is best consumed in a pub (and in color).

How to *Increase* the Acceptance Ratios of Top Conferences?

Graham Cormode[1], Artur Czumaj[2], S. Muthukrishnan[3]

Abstract

In the beginning was the pub. This work was triggered by a pub conversation where the authors observed that many resumés list acceptance ratios of conferences where their papers appear, boasting the low acceptance ratio. The lower the ratio, the better your paper looks. The list might look equally impressive if one listed the rejection ratio of conferences where one's paper was submitted and rejected. We decided to lampoon rather than lament the effort the PC typically put in: wouldn't the world be better if we could encourage only high quality submissions and so run top conferences with very high acceptance ratios? This paper captures our thoughts, and it is best consumed in a pub (and in color).

1. Introduction

We are good citizens in the computer science community. We are active participants in conferences, as authors, program committee (PC) members (PCMs) and organizers. We will describe some problems we have seen with computer science conferences, and propose some solutions, from the perspective of the PCMs. We really want to improve everyone's lives, and not waste everyone's time. We are also idealistic, so we did not always pay too much attention to the fine details of implementing the solutions we have come up with or their social cost and implications.

The Problem

We have noticed that the numbers of submissions to conferences have gone up over recent years[4]. This puts an increasing burden on the PCs of these conferences. PC members have to read far too many papers; yet an overwhelming fraction of them are rejected. Sometimes one feels that all this effort is pointless. But still, sub-standard papers have to be read and referee reports written. This leads to problems. We feel jealous, since it seems that other PC members get better papers. We start to worry that this is because we are expert in a particularly poor area. Maybe our own research is less than stellar if the rest of committee thinks we are best equipped to referee the chaff. Ultimately we begin to look forward to being on PCs with fear[5] and loathing. A big disappointment when reviewing a poor paper is that it will not die. Once a paper is written it will be revised and resubmitted over and over (causing pain to all

[1] Center for Discrete Mathematics and Computer Science, Rutgers University, Piscataway, NJ USA graham@dimacs.rutgers.edu
[2] Computer Science Department, New Jersey Institute of Technology, Newark, NJ USA czumaj@oak.njit.edu
[3] Division of Computer and Information Sciences, Rutgers University, Piscataway, NJ USA muthu@cs.rutgers.edu

[4] We are by no means the only people to have noticed this problem. It is highlighted in an editorial in SIGACT News (Vol. 35, No. 1, March 2004, page 2). There will be a panel devoted to the subject in SIGMOD 2004: http://www.sciences.univ-nantes.fr/irin/SIGMODPODS04/panelcamera.pdf .

[5] And this is a bad thing, since we all know that fear leads to anger, anger leads to hate, hate lead to the suffering, and this leads us to the *dark side*, e.g. giving up academia to work in Seattle.

involved) until it finds its correct level somewhere in the conference food-chain. We need to find ways to reduce the work of the PC, and perhaps also the authors. Several of our solutions below either directly or indirectly raise the acceptance rates of conferences, hence the title of our work.

2. Known solutions to high submission rates

We consider four of the strategies that are used to alleviate the load on the program committee and raise acceptance rates. In addition, we include two *case studies* of similar situations to see if there are lessons that can be learned from other areas.

A. Increase the size of the PC

As theoretical computer scientists who once took a class in systems, we know that the systems solution to the problem of too much data is to throw more resources at it. Many major theory conferences have between 10 and 20 people on the program committee. This is to be contrasted with a database conference like SIGMOD 2004, which listed 85 members of its main PC; ICDE 2004 with 140; or the networking conference INFOCOM, which in 2004 had a PC that was 178 strong. With such large committees, it is not really a question of reducing the work, but rather spreading the pain. Rather than the systems approach ("get more resources"), we look for a theory-style approach ("design a more efficient protocol").

Further, if we expand the PC to this extent, then we must allow PC members to submit papers or risk having no submissions: if the whole community is on the PC, who is left to write the papers? In TCS, we tend to look down on this practice. Forbidding PC members from submitting papers suggests the following strategy: in order to get the best papers submitted, one must form the PC from all authors who will likely write poor papers. This assumes that such authors will be able to recognize a good paper when they see one. Following this line of reasoning too far may lead some of us to question why we were asked to be on the PC of certain conferences: should we list PC membership under "dishonors'' on our resumé?

B. Delegate

Consider the following party game: we pass a brightly wrapped parcel from person to person as some

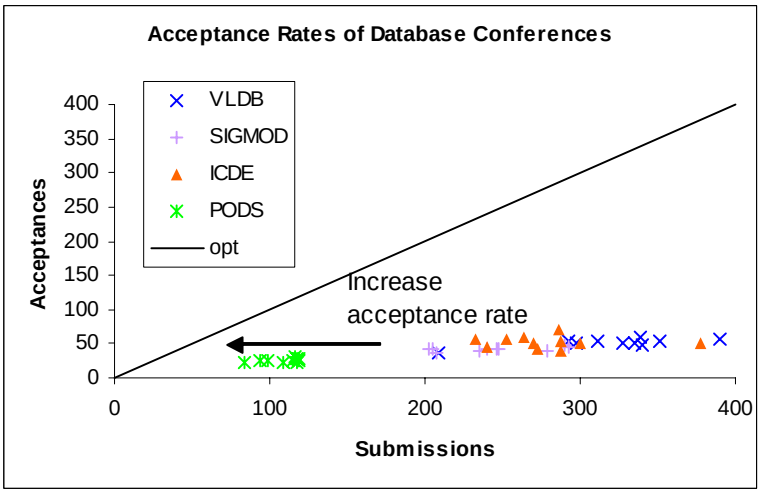


Figure 1: We propose to increase acceptance rates, partly by decreasing submission numbers. This figure is not referred to elsewhere in the paper.

music plays. When the music stops, whoever is holding the parcel unwraps it, revealing a pack of conference submissions which they must review. This, to a broad approximation, is how the sub-refereeing process works. Or, in some cases, a tree data structure is used: an academic passes each paper to one of his ex-students, who in turn passes it down until it lands on the desk of a graduate student, who must review it. For a PCM, the task of finding reviewers for each paper they have been allocated is sometimes so daunting that it looks quicker to do all the reviewing alone. This can make a mountain of work out of a molehill, and in the case of the PCMs, makes a manager out of a mole. Most of us want to be researchers, not managers, so why make us manage reviewers?

Case Study 1: University Admissions

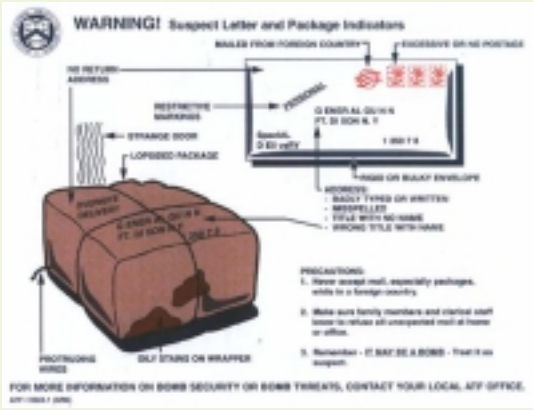
When considering applications, for future collaborators (graduate students who can write code and run experiments for us) or future competitors (energetic new faculty who will soak up all available grant money and leave us in their shadow), we face many submissions to sift through. From these we aim to choose the brightest and the best, or at least the solid and the so-so after the brightest and the best have accepted offers from elsewhere and turned us down. In order to reduce the winter woe of applications to read, committees apply various techniques.

i. Pre-filtering by GPA, Publication Count or other scores

A candidate with inferior academics may get rejected without any further consideration of their merits or future potential. It seems a little unfair to apply this rule directly to conference submissions — “You must be this smart to submit here” — but finding technicalities on which to reject papers might work. One could throw out papers which are badly presented, use an ugly font, or go one line over the ‘strict' page limit. Unfortunately, people usually want a reason for their rejected papers and some may find complaints such as these... unacceptable.

ii. Letters of recommendation

Nowadays any application is bolstered with letters that are never less than effusive with praise for the candidate's intellect, research ability, communication skills, punctuality, obedience and hygiene. This is a model we could apply to conference submissions: rather than soliciting referee reports, each paper could be submitted along with two or more letters of recommendation from non-authors. But the success of this is far from guaranteed. Exercising our control over who our referees are merely ensures that PCs will be inundated with hyperbolic praise for papers. Computer Science is a small enough community that few would dare to write a damning review of a colleague’s work when asked to give an honest appraisal.



Some recommendation letters may be suspect

Where such a system could succeed is in highlighting the strongest contributions. An unsolicited recommendation for a new result from an authority in the field could guarantee a fast decision to accept. Certainly, a trustworthy certificate of correctness could reduce the burden of refereeing and checking. But papers that generate such a buzz are few, and these are sufficiently well advertised by their authors that little further is required by the PC than a decision of whether the advertised result is strong enough to accept. It is already easy to skim off the cream of the crop, and filter out the dregs. The problem we face is to find a way to deal with the mountain of mediocrity in the middle.

Case Study 2: Spam Filtering

We are all too familiar with finding our inbox stuffed dull of unwanted material with vastly exaggerated claims for how some revolutionary new discovery gives amazing results, and how it will make us more of a man or a woman. And as well as papers to referee, we also have to deal with spam. The analogy between junk email and junk papers is enticing, and makes us wonder whether we can apply some of the methods to detect and reject spam to conference submissions.

α. Keyword Filtering

The crudest form of spam detection marks as suspect any message with bad words as 'viagra', '$$$', or 'AOL'. What keywords might be useful to detect junk papers? Perhaps phrases like "a simple proof of P=NP", "$\frac{\log \log \log n}{\log \log \log \log n}$" or "PRAM algorithm", should set our alarm bells ringing. But such an approach is too crude for email, as spammers soon disguised their keywords ("V*1ag.r.a"); for conferences this technique would only catch the most trivially bad papers, and has a high false positive rate.

β. White-listing

When spammers sent from real addresses, it was easy to block their mail. Now they use ad-hoc faked addresses. The white-listing approach involves the recipient keeping a list of people from whom they automatically accept mail. The rest is marked as suspect. This approach does not translate well applied to conferences. Of course, we treat each submission identically, and never accept a paper from a colleague without applying the same level of scrutiny and verification we apply that from a rival that criticizes our previous work. While established figures might approve of accepting on the basis of name alone, this will stifle innovation and reduce creativity, as can be seen by comparing people's work before and after they are granted tenure.

"If your name's not down, you're not coming in"

γ. Signature Matching

In the Signature Matching scheme, a central database holds signatures of messages previously marked as spam. New messages are checked against this and rejected if there is a match. This is a tempting way for us to detect papers that have been rejected from one conference and resubmitted to another with only superficial changes. But computer scientists are wily, and will find ways to defeat the matching algorithm. Just as we have a repertoire of tricks, hacks and macros to squish an 18 page paper into 12 pages of LNCS style without altering the content, so we will find ways to automatically beat the system without any effort on our part. One trick used by spammers is to insert random words and phrases that make no sense whatsoever to the body of the text. In computer science parlance, this is known as "adding a motivations section".

We could go on, but metaphors with spam don't really work. We can see at a glance if a mail message is junk, and hit delete. In a conference setting, there are few papers that are so bad, and the search and destroy operation is fairly painless. The real problem comes in detecting or discouraging contributions that appear serious, and only after closer inspection are found to be mostly content-free.

C. Accept (almost) all submissions

In other fields, where conferences have less formal relevance (i.e., researchers do not list conference papers on their resumes), the conferences need to be less rigorous about policing their content. For example, in mathematics, it is standard to submit a short abstract[6]. Then, provided these abstracts are deemed relevant to the conference, the author is invited to give a presentation at the conference. Acceptance to a conference carries little or no cachet, and consequently little effort is expended in checking and refereeing. The acceptance ratio of these meetings is then the number of authors who accept the invitation to submit, rather than the other way round, and is typically quite high. This approach dodges our problem: instead, it pushes more of a burden onto Journals, which will be faced with higher numbers of submissions that need to be carefully scrutinized. Although we focus here on conferences, making this "somebody else's problem" is no solution.

D. By invitation only

In some areas, the majority of talks at a conference are invited talks. This relies on the program committee knowing enough about recent research to know who has interesting new results, or just who has a reputation for giving interesting talks. Sometimes this is achieved in Computer Science by calling the event a workshop, which has the side effect of giving all workshops a bad name. A similar effect can be achieved by keeping the Call For Papers a secret, so that only friends of the Program Committee get to hear about the conference. Some conferences achieve this state of affairs by accident. This can be observed when an emailed last minute deadline extension effectively serves as the initial call for papers. This is handy for anyone with a stack of half-written (or half-baked) papers, but not for the rest of us. Such a closed system goes rather contrary to the spirit of openness and the spread of knowledge that we strive for in Computer Science. We want to reduce the burden on PCs, but not at the expense of lowering standards by only accepting the work of our cronies.

3. New Conference Procedures

From the preceding discussion it is clear that extant methods do not work. One needs radically new suggestions to solving our problem.
Our work is predicated on the following. The problem is too many papers are submitted, and then recycled after being rejected. It is not that the case that papers are all good and we have a tough time choosing which deserving papers must be dropped. If this were true, then instead we would be worrying about having to increase the number of accepted papers and the consequent organizational problems. Rather, it is the fact that we face many papers that are not appropriate for the conference, but that it takes us too much time and effort to discover this.

We propose a few approaches to reduce sub-standard submissions and reduce the load on PCs:

- Reduce the number of papers submitted
- Reduce the work for the PC by filtering
- Multi-resolution papers
- Economic approaches
- Accept all papers
- Information-based approaches

[6] As opposed to the computer science "Extended abstract", which is frequently several pages longer than the full paper strictly needs to be.

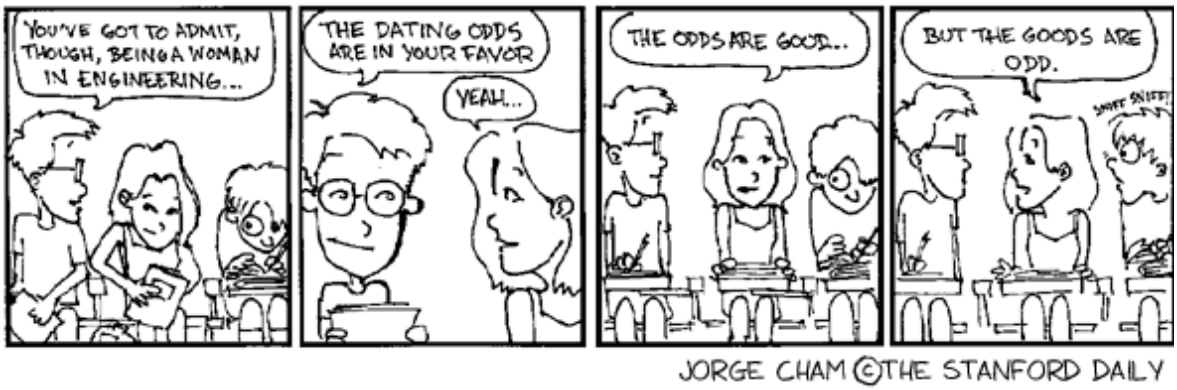


Figure 2: We propose to turn conferences into beauty contests

For each of these, we outline our recommended approaches, and list some of the research problems that emerge from them.

A. Reducing the Number of Submitted Papers

We can consider various ways to restructure our conferences that will guarantee a low number of submissions and a high acceptance rate. For example, we could only accept the last 100 papers to arrive, or the first 100 papers to arrive[7]. In practice, neither of these is practical: in the former case, it will just ensure that any automated submissions server is brought down under a barrage of submissions as the process comes down to a question of who can click 'submit' the closest to the deadline (or write a program to automatically submit the paper as close to the deadline as possible). In the former case, then we would find the same effect at the official "start" date for submissions, and so we succeed only in moving the deadline earlier. Or, we could find some other way to fix the numbers, perhaps by randomly deleting all but 100 of the submissions. Some conference management systems have this feature built in already, but it is currently considered to be a bug, not a feature.

Sometimes individual authors are responsible for a large fraction of the work of a PC. Some authors regularly submit several—as many as five or ten—papers to a single conference. We propose limiting the number of papers that any author can submit to a conference—perhaps to as few as one. Such restrictions apply in certain grant writing scenarios, such as the NSF ITR, where it is assumed that working on one grant will consume all of one's available time. This approach could give rise to interesting combinatorial decision problems for multi-author papers, and would also require some authors to commit to papers. This would punish the most prolific authors… or at least slow them down a bit so that we can more reasonably compete with them. Since they can no longer deliver half a dozen papers in a single conference, the prolific would have to travel more, thus keeping them tired and jet-lagged and hopefully reducing their output and hence their overall burden on the community in terms of reading and reviewing their interminable work. To succeed, the limit may have to be set quite low. For example, in the recent SODA 2004 conference, there were only two authors with 3 or more papers out of the 135 accepted submissions. But setting the limit at one submission would have affected 55 authors who had two or more accepted papers.

[7] The former suggestion has been proven *experimentally* to work well in practice … as we have heard on many business meetings, in many conferences 90% of accepted papers are those submitted in the last moment before the submission deadline … sometimes even after the deadline.

Problem 1: *We must evaluate the effect of limiting each author to one submission on paper submission rates. Would it really reduce load on PCs? This needs detailed data analysis at various conferences across multiple years.*

Problem 2: *Given a set of co-authored papers and a limit of at most submission one per author, how would authors maximize the number of submissions? Can this be done without the authors discovering who else is submitting to the conference?*

B. Reducing PC Work by Filtering

It is often said that getting computer science papers into conferences is in part a *beauty contest*. To an outsider visiting a CS conference, this remark would be quite surprising. What is meant is that the most attractive (or popular) papers, on hot topics or containing many buzzwords stand the best chance of being accepted. We propose to formalize this system.

All submissions are listed and paraded. Each PCM ranks each paper according to her/his interest in it, *based only on a cursory inspection*. Perhaps each PC member will have a limited budget of total points to allocate. The papers with the highest popularity would be accepted to the conference. To reduce bias and vote rigging, we would take the median score rather than the average.

In fact, we believe that this ranking would give a very good *prediction* of the output of a typical refereeing process without the tiresome bother of actually reading the papers. This process would also accept papers that are most interesting for the PC members, and so lead to conferences with the most interesting papers. The downside is that this process certainly is not aimed at selecting the best papers, and some excellent papers on unpopular topics would never make conference publications. This also could lead to promoting known-names only. The next step is to automate this process: use machine-learning and data-mining methods to rank papers based on previously seen decisions. It may then be desirable to keep the rules that are found a secret, since otherwise this gives a recipe of keywords and phrases to use to increase chance of acceptance.

Problem 3: *We must implement and test computerized filtering based on, e.g., keywords in the abstract and author names. How does this contrast to the rankings (scores) produced by the PC? What are the secrets to getting a paper accepted?*

C. Multi-resolution Papers

For our next model, we take inspiration from procedure used for some grant proposals. Now authors must submit **two** versions of the paper: a (say) 12-page **long paper** containing both general presentation of the results and more detailed or complete analyses, and a 2-page "long introduction" **short paper**. The work of the PC is performed in two steps. First, the PC would read only the short paper and made the first selection on this basis alone. In this way, a large portion of all submissions would be rejected with a significantly smaller effort from the PC. In this phase, the PC members would have to read only motivation, comparison to prior work, and the statement of the results[8]. With enough caring brutality, half or more of the submissions could be rejected in this phase, significantly reducing the work of the PC. This way also allows a dynamic resource allocation approach: if in the first phase the PC decides that a given paper should be accepted unless its

[8] Some PC members admit privately that they already apply such a system informally, by only reading the introduction of most submissions.

analysis is incorrect, then we need only 1-2 experts in the area to verify the soundness of the result without further troubling the PC.

To make this proposal more user friendly for contributors those who don't want to waste their time on preparing the short paper, if only a 12-page long paper has been submitted the PC could run a script that extract a two page prefix of the paper – these two pages would then be by default treated as the short (version of the) paper. It should be also very clearly made in the CFP that the PC would judge the papers in such a two-phase process. We also believe that ensuring that the discipline of writing readable two page self-contained summaries will improve the quality of CS papers.

What about the problem of rejecting good papers in the first phase? We argue that, if the result cannot be succinctly summarized in two pages for a general CS audience, then the paper, in its current shape, is not ready for publication. In general, good papers read well, and so we expect few good papers would be pulped in this way. The rejections of at the first stage can be given a positive spin: this is a faster option than waiting for complete referee reports, and avoids keeping the fate of the paper up in the air for months on end. How does this affect acceptance rates? It increases them! Suppose a conference accepts 25% of submissions. Then, under this two-stage system, 50% are accepted by stage one, and after detailed study, 50% of those are accepted at stage two. Thus the conference now has 50% + 50% = 100% acceptance rate.

Problem 4: *The two-stage process is just one possibility. Design and analyze the optimal structure and acceptance rates of a reviewing process so as to minimize PC effort and maximize quality of accepted papers. Given n submissions, how many stages does the optimal solution need?*

D. Economic

The global capitalist system tried to reduce the workload of computer science conferences, by starting the Internet Economy. This had the effect of redirecting the attentions of those researchers who thought that their perfect hash functions could be turned into perfect cash functions. Unfortunately, this effect was not felt uniformly by all areas, and it did not last. Before we set about beginning a new Internet bubble we should also admit that it would probably cut down the good submissions, but not the bad. So, instead we will come up with some more principled methods for using economic ideas to reduce poor submissions.

The obvious approach is to design schemes of payments and penalties, to punish authors who write poor papers that waste the PC's time. This is open to criticisms that authors from poor institutions would be affected, while those from rich institutions or with big grants could more easily pay any fees levied. Instead, we look to less-direct applications of Economics.

We advocate outsourcing paper refereeing to India. It has a trained population that is well versed in recent results in computer science. This marks a natural progression in the market forces. This will not hinder employment in other countries. In fact, this will lead to academics in US and EU taking up higher value jobs. It will also be an opportunity to learn another language as researchers learn the nuances of "Indian English", putting commas arbitrarily, deleting articles carelessly, and long-winding sentences that abruptly…We will have conference call centers (pun intended) in India and change the social fabric there.

Problem 5: *Another approach is to tap* information markets: *a system where value is attached to each paper, so that the best papers attain the highest value. Effectively, this means that the PC starts betting on which papers or topics are accepted. How could such a system be designed and operated effectively?*

What would the pay-offs be? What about futures (predicting that X will write a paper in the next 6 months on cache-oblivious algorithms) and derivatives? If we correctly predict all the accepted papers, can we make enough money to retire to Barbados?

E. Accept all papers (again)

Having already rejected the idea of accepting all papers in the previous section, we return to the idea and propose it again as if it was an entirely novel idea. Consider a scenario in which the PC will accept every paper submitted to the conference, but there will be more than one category of accepted papers. The first category is as usual – we want to have the best submitted papers appearing in the "real proceedings" of the conference and being presented at the conference. The lowest category contains all papers that normally would be rejected. We want to accept them to proceedings only; no talk, no poster, no fame, only "publication". Realistically, such publication would be electronically only. In this way once a "rejected" paper is accepted it cannot be resubmitted to any other conference because it has been officially published. (It's possible that the rejected papers will become cult reading amongst grad students and other researchers with too much time on their hands, for the *schadenfreude* of reading the worst papers).

This approach certainly leads to our goal – we are *increasing the acceptance ratio to 100%*. At the same time, we are also significantly reducing the work of PC. The nightmare of being on the PC stems from the need to referee the same paper multiple times, as it is rejected and resubmitted to multiple conferences. We conjecture (an exciting research problem) that many conferences have around 50% (or more!) of submissions being papers rejected from other conferences. And so, we are achieving our objective in a perfect way – we are increasing the acceptance ratio to 100% and at the same time we are reducing the work of the PCs by a half!

Rich Martin suggests a more sophisticated version of this approach. The PC still accepts every paper submitted to the conference, and its work will be to decide which form the publication of the paper will be: a "full" paper with a talk, as a poster paper, only in the proceedings (no talk or poster), or published on the website only. This gives a more fine-grained "quality" rating for the paper. The intention is that this will discourage substandard submissions made to "fish for comments" from reviewers, since everything submitted will at least end up published on the web. This approach reduces the time to publication for papers that get rejected a few times before finally being accepted, and so keeps the whole community up to date with the latest research. Lastly, this avoids problems with PCs rejecting good papers, since the paper still gets published, and the community gets to judge whether the PC placed it in the right "bin". Perhaps authors will be allowed to "retract" their paper if they feel that the PC has grossly underestimated their contribution. This approach leads to many interesting research problems.

Problem 6: *What is the best strategy for authors submitting their papers? Is it better to submit the paper to strong conferences, where there is a big chance that the paper will appear in "rejected proceeding", or to a weak conference and not risk our reputation? Another problem is of a social nature — how to convince our colleagues in other fields who are deciding about our tenure, etc., that publications in some proceedings are very prestigious while some electronic publications are not worth the bits they are written on.*

Figure 3: An Amazon-style approach to paper refereeing

F. Information-based

There is a great deal of information generated in the process of reviewing and evaluating papers. One possibility is to use this information as a weapon against authors who submit substandard papers. At the extreme is the possibility of making all reviews, of both accepted and rejected papers, fully public. One could even envision an Amazon-style system where each paper is listed along with reviews (from referees, and from casual readers) and given five-star ratings. We acknowledge that such a course of action is unlikely to prove popular, and given some of the reviews our past papers have received (from reviewers who must have somehow failed to appreciate their greatness), we think this might not be a good idea.

Indeed, even just the information about the titles and authors of rejected papers would seem to be too sensitive to make public, specially in the case of authors for whom, when given the title of one of their unpublished papers, it is possible for one to completely reconstruct their claimed results over the course of an afternoon. We propose that it is reasonable to release the (multi)set of names of all people who submitted a paper to the conference, in no particular order, in addition to the list of accepted papers. This gives a mild disincentive for people to submit papers to conferences if they are fairly sure that they will not be accepted. Think about your chairman deciding whether you will get tenure or not on the base of your "success ratio" - the number of accepted papers over those submitted.

This effect can be amplified if these "participants lists" are made publicly available for many conferences. Then we can pursue a baseball analogy, and begin to compute various statistics on performance. For example, the basic batting average, given by (number of acceptances)/(number of acceptances + number of rejections) gives an individual's personal 'acceptance ratio'. This number could be quoted on their

resume, in addition to the acceptance ratio of the conferences they were accepted to. Such statistics could find official use, being an additional criterion to consider in addition to awards, jobs, service, PCs served on and so on. Do not underestimate the power of statistics in baseball or real life.

The rejected papers are the ones that really create the most work for the PCs. A paper once rejected is often re-submitted, possible many times. This is certainly a natural situation, because of different standards of various conferences. Nowadays we even set up conference deadlines to catch as many as possible papers rejected from stronger conferences! To reduce the work of the PC we could maintain a central confidential database for PCs eyes only. Every reference report or a report from TCS conference would be stored there and could be sent on the request of a PC chairman of any other conference to which a paper has been re-submitted. Thus, the PC could reuse this information and so reduce their work. It also might discourage recycling the same bad paper again and again, as the paper will get similar low score.

Yin Zhang agreed with this direction: "a big part of the problem comes from recycled bad papers. So it'll be very useful if multiple conferences can collaborate on this (while maintaining some form of anonymity etc.) For example, it helps tremendously if we can identify recycled papers and obtain the reviews for the old submissions, or better yet, ask some (if not all) old reviewers to review the paper. This clearly saves a lot of work because it effectively reuses the resource (past reviews / reviewers). It also discourages recycling the same bad paper again and again — as you tend to get the same low score."

Problem 7: *How to manage and index the database of reviews? We must design a privacy-preserving scheme to manage reviews from all TCS/Database/Networking conferences and make it available to PCs.*

4. Experimental Evaluation

In order to evaluate the efficiency of our proposals in increasing the acceptance rates of conferences, and reducing the burden on program committees, we suggest that a detailed set of experiments be carried out. Ideally, our proposals should all be adopted immediately for all conferences, but we acknowledge that this may cause some upset. Instead, we plan to carry out a series of experiments on past conferences to analyze the effect of our changes. Due to the increasing computerization of conference deliberations, from papers submissions, PCMs "bidding" for papers to read, discussions and decisions, then each conference potentially generates a very large amount of data that can be analyzed and interpreted. Our next step will be to obtain such logs and subject them to detailed analysis. Although there are ethical and privacy issues to be dealt with, we are confident that we can obtain logs from several major conferences, and discover what effect certain proposals would have on the outcome. This will help us ask the problems that we have posed so far without risking causing problems by deploying our ideas in the field without some idea of the changes they will effect. This leads us to pose a 'meta-problem':

Meta-Problem 8: *To design and carry out experiments on large amounts of conference server logs in order to address the previous stated problems, and to see how different approaches interact. For example, the effect of using simple filtering rules to accept or reject papers. E.g., how many papers that initially receive a lukewarm review eventually get accepted?*

This manifesto for data compilation and analysis about the PC process should cause some researchers to salivate. In recent times, rather than actually building new complex systems, the community seems to encourage analyzing data and extracting minutiae from them.

Yin Zhang commented further on the question of experiments (possibly spurred on by the previous slander on systems folks). "One thing missing from the paper is a detailed workload characterization. For example, what are the percentages of recycled vs. original paper? How often does a recycled paper get accepted? Such characterization is essential in order to derive a practical solution."

5. Conclusions

There are many problems with the conference system beyond those focused on paper submission discussed here and we have in mind solutions for *all* of them. For example, the US National Science Foundation would like to encourage diversity in schools of higher education among students and faculty. We suggest that less published a school faculty is the more chance we accept their papers, thereby inducing graduates to prefer jobs in Dakota or Dubai rather than Massachusetts. The same broadening initiative applies to women and minorities. International coalitions should also be rewarded.

Another constituency that is currently discriminated against by conferences is that of cranks: the acceptance ratio for papers that prove Goldbach's conjecture or P=NP is virtually zero. We would suggest setting up a special Conference for Rejected and Abandoned Papers or Journal of Unexpected and Novel Knowledge to nurture and divert this community. This could be combined with the proposal of accepting all papers and then publishing separate Proceedings with those that should be rejected, as discussed in Section 2.C. Think also about a special issue of a journal devoted to the *worst* papers in a conference …

There are other aspects of conference organization that needs fixing. For example, how should the members of the program committee be selected in a fair manner? Since this is service to the community, perhaps it should be done by random selection, like jury service. Note that in the United States, the prosecution and defense (the program chair) can strike off jury members thought to be poor decision makers. If this whole process is televised, much good will come of it. This will be explored in a future production.

By now, we hope to have convinced the readers that the problem we have addressed is very important. We conclude with one final thought from our pub conversation about conference *pub*lications. There are no jokes that begin "A computer scientist goes into a pub and…". We hope that this work will begin to remedy this situation.

Acknowledgements

We have been greatly influenced by the works John ("Jack") Daniels.

References

References available on request.

Picture Credits

Page 1
Bar Conversation http://www.creaweb.it/grafica/illustrazione/giallo/giallo3.htm
http://choson.lab.net/images/ds-bc1.jpg

Page 2
Acceptance rates collected by http://wwwhome.cs.utwente.nl/~apers/rates.html

Page 3
http://www.usps.com/news/2001/press/pr01_1022gsa_print.htm

Page 4
http://news.bbc.co.uk/1/hi/england/3398695.stm

Page 5
http://www.pages.drexel.edu/~kd35/bar.htm

Page 6
http://www.phdcomics.com/comics/archive.php?comicid=8 © Jorge Cham

Page 7
http://www.daleamburnfineart.com/Pages/conversation.html

Page 8
http://www.public.coe.edu/~pthompso/bars3.html

Page 9
http://www.public.coe.edu/~pthompso/bars3.html

Page 10
Source images from
http://www.amazon.com/exec/obidos/tg/detail/-/0898714907/qid=1080833216/sr=8-2/ref=sr_8_xs_ap_i2_xgl14/102-0007473-2076136?v=glance&s=books&n=507846
http://www.amazon.com/exec/obidos/tg/detail/-/0262032937/qid=1080833155/sr=8-1/ref=sr_8_xs_ap_i1_xgl14/102-0007473-2076136?v=glance&s=books&n=507846
http://www.amazon.com/exec/obidos/tg/detail/-/0316919160/qid=1080833242/sr=1-1/ref=sr_1_1/102-0007473-2076136?v=glance&s=books

Icon from http://www.btinternet.com/~donnypubs/

Page 11
http://www.portlandartmuseum.org/artwork/exhibitions/weegee/catalog/full size/sleeping in a bar.jpg

Page 12
http://www.slickrockdesigns.com/artgallery.htm