Evolving Program Construction at the System Dynamics Conference

Bob Eberlein, isee systems
Billy Schoenberg, isee systems
Sara Metcalf, University at Buffalo

Abstract

For 2024 we are using a new process for organizing submitted work into sessions for presentation at the conference. The process involves more automation than has been used in the past, but at the same time is a product of the evolution of the overall conference over the past 30 years. In this paper we give some background on how we have arrived where we are, then detail the mechanics of assessment and assignment that are in use for 2024. The key innovations involve assignment of work to Thread Chairs inside and outside of the submitted Thread, then creating sessions based on similarity of the submitted works. The second part of this is more complicated, and two similar but distinct algorithms are described to accomplish it. For 2024 we used the simple greedy algorithm described with additional flexibility to combine the algorithm with individually combined papers.

Background

In the simplest terms, putting on a conference with submitted work involves collecting the submissions, determining which to include and which to exclude, and then scheduling those that are included for presentation. The first conferences of the System Dynamics Society were organized by small groups of people who simply used this guidance in whatever manner seemed to work. The submissions themselves were hard-copy, sent by mail. In a time before Excel, the organization was done by shuffling index cards or cut-up pieces of paper containing titles into what would eventually become a schedule.

This was all done by a single person or a small group of people working side by side. The System Dynamics conference required full papers, as opposed to an abstract, making the evaluation process more burdensome. As technology progressed, we moved from paper to electronic submissions, initially by email. The less burdensome process allowed for more submissions, and as the number of submissions grew, so did the difficulty of assessing and organizing by one or a handful of Program Chairs.

After the Society moved to an office at the State University of New York at Albany, a web-based submission system was implemented, and a peer review process was put in place beside it. This work was primarily done by Vedat Diker as a PhD Student at Albany working for the Society. The peer review process eased the assessment process, and the much of the organization was managed by the Society's Home Office. This was, of course, after Excel had come into common usage and thus the burden on the Program Chair or Chairs was manageable, but the distributed decision making and organization was leading to noticeable inconsistency between sessions, with many sessions including work that did not seem to fit together well. Part of this was due to the highly variable review process, and part to the difficulty of organizing so much work coherently.

In 2004, or perhaps a little later, we created the role of Thread Chair to provide somewhat more oversight. To quote Stafford Beer – "for every brick a bricklayer." This distributed the burden of assessment and organization to a number of different, and trusted, individuals. Like anything, the list of Threads and the work of Thread Chairs evolved over the years. Focus Areas for each Thread were introduced in 2022 so that submitting authors would select a Thread as well as a more specific Focus Area within that Thread. With each Thread having at least 2 Thread Chairs, and some Threads receiving many more submissions than others, it has been an ongoing process of adjustment to keep things working smoothly. Coordination of work even within a Thread was also sometimes challenging when Thread Chairs were in different time zones.

In 2023 we asked the question of ourselves whether there was a way to organize conference programs that would ease the coordination burden and streamline the process; that answer was clearly yes. As to whether the way we envisioned would actually be feasible to implement, the answer was somewhat less clear. We broke the process down into 2 parts – assessment (disposition), and session organization. We will discuss each in turn.

Assessment (Disposition)

There are three categories of submissions that are allocated among five presentation categories, with authors further subdividing the Interactive category as shown in Figure 1. Assessing the three submission types side-by-side is challenging, and this is especially true for Research Papers and Practitioner Applications, which can be put into any of the Plenary, Parallel, Interactive, WIP, or Feedback sessions. Thus, when presented to Thread Chairs for assessment, disposition of these different submission types can lead to different outcomes depending on the backgrounds of the Thread Chairs and the fit of a given submission with other submissions.

In short, with the traditional way of forming programs, the assessment and session organization steps are not always sequential, whereas the approach we are about to outline is. In the discussion and extensions section, we will elaborate on ways that might make the process described more iterative, and the final program for this year's conference did involve iteration. For now, however, we will treat assessment as a distinct step.

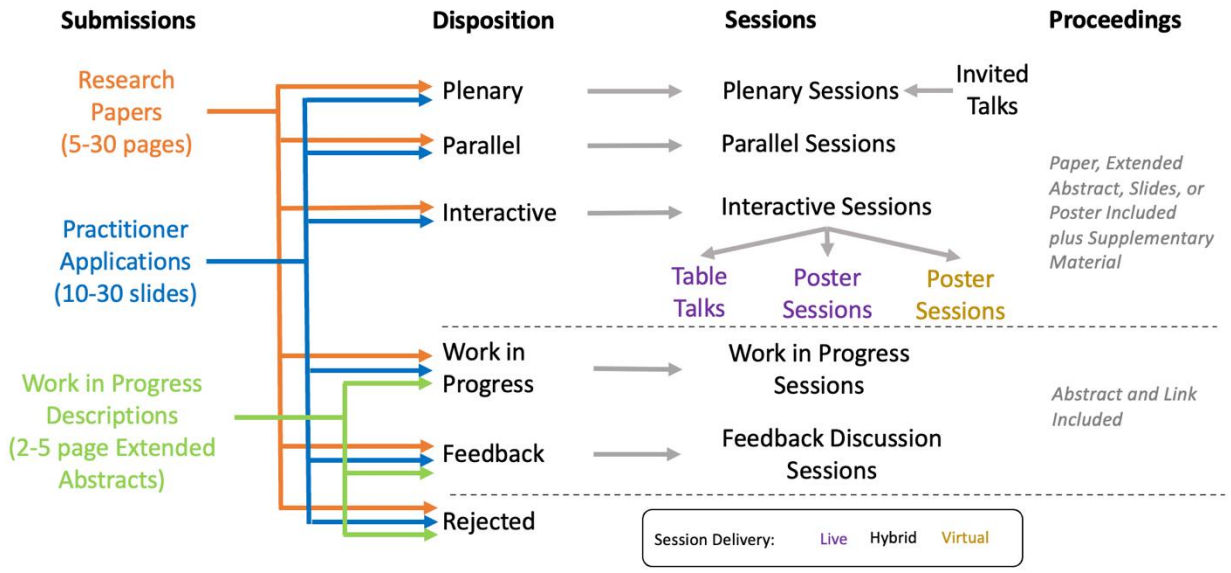

Figure 1. From Submission to disposition and session creation.

The categorization of submissions is done by scoring each submission, then sorting the scored submissions, then determining a cutoff for each of the disposition categories.

The existing peer review process used for the Conference already supplies quality assessments along different dimensions that are readily converted into numeric scores. While these scores were observed to be strongly correlated with ultimate paper disposition for the 2023 conference, they are not sufficient. The peer review process is inclusive by design, which means there will be wide variability within the reviewer pool. Thread Chairs, on the other hand, are chosen because of their demonstrated knowledge of System Dynamics, as well as their expertise in the areas of specific Threads. To simplify the assessment process for the Thread Chairs they are asked to simply score each submission they review on a scale of 1 to 10.

To ensure more conformity across Threads, and to manage the workload of the Thread Chairs, each submission is assigned to one Thread Chair for the Thread to which it was submitted, and another chosen randomly from all Thread Chairs willing to review in that Thread. (Each Thread Chair was asked to indicate at least 3 additional Threads for which

they would review.) The Thread Chair and peer assessments then provide the information needed to rank papers.

The ranking gave more weight to Thread Chair assessments, but the exact weight is not predetermined. In experimenting with this type of ranking based on the 2023 submissions, we used the initial disposition chosen by the thread chairs as a stand in for the Thread Chair scores and it was given an 80% weight relative to the peer reviews. For 2024 we ultimately ended up using a fixed factor scoring approach as discussed in the section on Experience. With every submission having a composite score, they can then be sorted from highest to lowest.

Because work submitted as Work in Progress can only go to Work in Progress or Feedback sessions, it is necessary to rank this work separately from Research Papers and Practitioner Applications. In working with the 2023 submissions, Research Paper and Practitioner Applications were kept together. For the 2024 submissions the categories were all kept together with an adjustment factor included for each category.

The sorted submission lists allow the Program Chairs to assign cutoff scores for assignment to the different categories. These thresholds are based both on the value the submissions bring when presented in the chosen format and the physical presentation constraints dictated by the conference venue. Though venue constraints have always been present, determining grouping for submissions as a whole ensures that different Threads do not get more strongly restricted because of venue constraints.

With the cutoff points in place, each submission can then be assigned a disposition in accordance with Figure 1.

Session Organization

Our approach to organizing sessions is not novel. Gündoğan and Kaya (2023) outline a similar set of steps. What is unique to our conference are the types of sessions and the main topic area categorization (in our case Threads), used for conference submissions.

Sessions involving oral presentations work best when the material being presented by the different speakers is closely related. This means grouping work by topic area, method of approach, or some other unifying characteristic. The self-selection of Thread and Focus Area by people submitting work provides a first pass at this grouping, but it is not sufficient to make truly coherent sessions. Traditionally Thread and Program Chairs have to dig a little deeper to organize things, and inevitably some sessions are more coherent than others.

In automating the session creation process, the first step is thus to determine how similar one paper is to another. To do this, we use the text classification scheme developed by Joulin et al (2016) that is available through https://fasttext.cc. This code allows two sets of text to be input and returns the similarity, also called the cosine, as a number between -1 and 1 where 1 is identical, 0 unrelated, and -1 opposite. The code can be applied to arbitrary text strings. We experimented with a number of different sets of text:

- Full article text. This yielded results that seemed dominated by writing style rather than content. It would also pose challenges to implement automatically.
- Abstracts. The abstracts are more structured, or at least shorter, and high consistency measures here did seem to correlate with content as we judged it. That said, the discrimination was not high. In fact, most abstracts for the conference seem to be quite close to one another.
- Titles. The titles are shorter still, and when well-constructed, do contain the essential words to describe a work. Still, there was significant variability in pairwise comparisons.
- Thread and Focus Area. These are terse and consistent, and the comparisons clearly reflected this. However, as noted above, Thread and Focus Area by themselves are not enough.

In the end we ended up using Thread and Focus Area with a high weight (80% in our test of 2023 submissions) and combining Title and Abstract with a lower weight. Since authors and Thread Chairs entered keywords for 2024 submissions, we had keywords in addition to thread and focus areas and used a 60/20/20 weighting for the three as discussed in the section on experience. Notably, we only used the title, and not the Abstract at all as the Abstracts were all considered quite close to one another.

Once the pairwise comparison of every submission with every other submission has been made, it is then possible to group the most similar submissions. This is done after the submissions have been assigned a category as detailed in the assessment discussion above. In order to do this, it is necessary to define the concept of session coherence, which is done by combining the pairwise comparisons. Simply put, session coherence is defined as the product of all possible pairwise comparisons. So, for example, with 4 works A, B, C, D and pairwise comparison denoted by AB and so on, the session coherence would be:

AB * AC * AD * BC * BD * CD **(1)**

By definition, a work has similarity of 1 with itself, and similarity is symmetric (AB = BA) so equation 1 gives all the useful combinations. This metric clearly decreases as the number

of works increases, but this is not an issue as long as all sessions are being assigned the same number of presentations as they are for our conference.

Simple Greedy

Our first algorithm is a simple greedy algorithm that starts by finding the two works that have the highest similarity score out of all those that need to be assigned. These two works are used to form the kernel for a session, and the next paper found is the one that results in the highest coherence score for the session. We continue adding papers based on this highest coherence criterion until the desired number of papers has been added. The process is then repeated with the remaining papers.

Using this approach, we would expect sessions to start out with high coherence, and then have decreasing coherence until the last session or two which would be collections of seemingly unrelated material. While that general trend does hold true, when run using the 2023 papers, both the speed of degradation, and the number of very low coherence sessions, pleasantly surprised us.

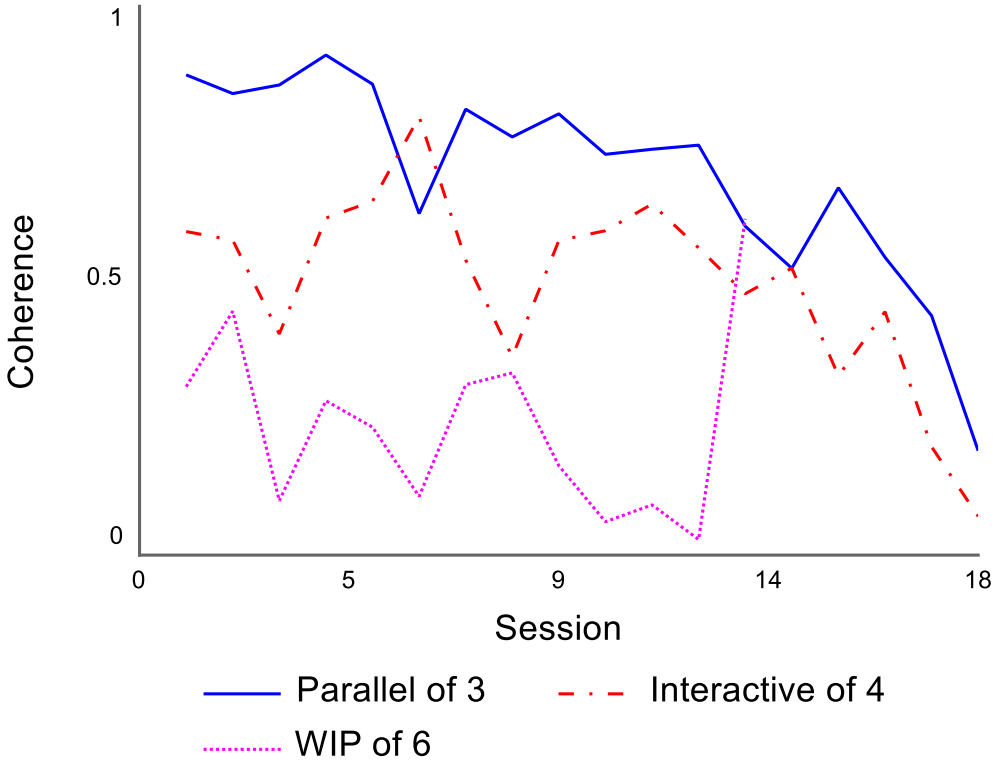

Figure 2. Coherence of sessions for 3 different session types using simple greedy

As can be seen in Figure 2, the coherence does generally decline for later sessions, though it is not uniform. It also stays relatively high until the last 4 or 5 sessions which, though it may seem like quite a few, is not out of line with programs that were created in the past. The more presentations in a session, the lower the coherence, as would be expected from formula 1. The final uptick in the WIP coherence is a result of fewer presentations in the final session (total works was not a multiple of 6). The interactive groupings of 4 per session was used for testing. In the final program creation, there were Posters grouped by thread with a flexible number of works, Table Talks with 3 works per session and Virtual Posters with 4 per session.

Least Worst

The least-worst approach starts out the same as the simple greedy algorithm, but after finding the best pairwise match, it then looks for the submission that has the lowest coherence with that pair. That submission is then matched to its closest work, and then the submission with the lowest match to existing sessions (taken as a product of the coherence with each session). This continues until there are two submissions for every session, then we pass through each session adding the best matching submissions of those that remain.[1]

[1] There is an obvious variation on this in which one submission is added to each session in succession until the sessions are filled. We have not experimented with this variation.

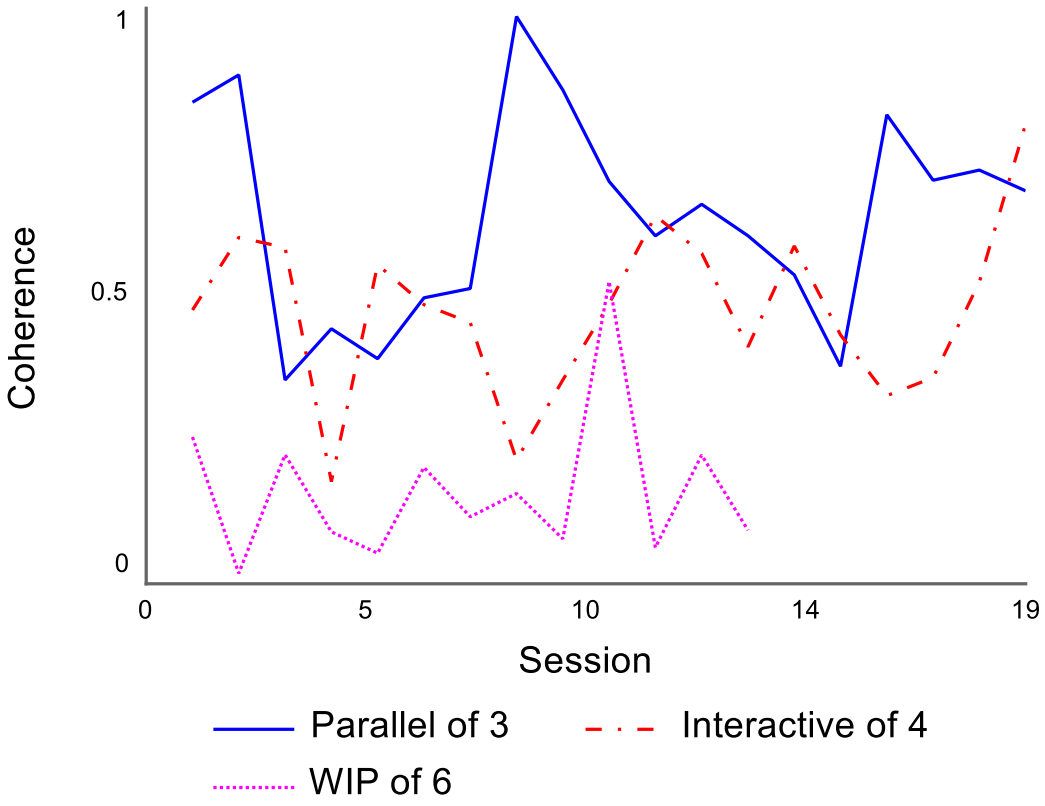

Figure 3. Coherence of sessions for 3 different session types using least-worst

One would expect the least-worst approach to generate more uniform coherence, and this is largely true as shown in Figure 3. Certainly there is no tendency to drop in the later sessions. However, a comparison of the two algorithms for parallel sessions as shown in Figure 4 reveals that the least-worst approach also does quite a bit worse on average.

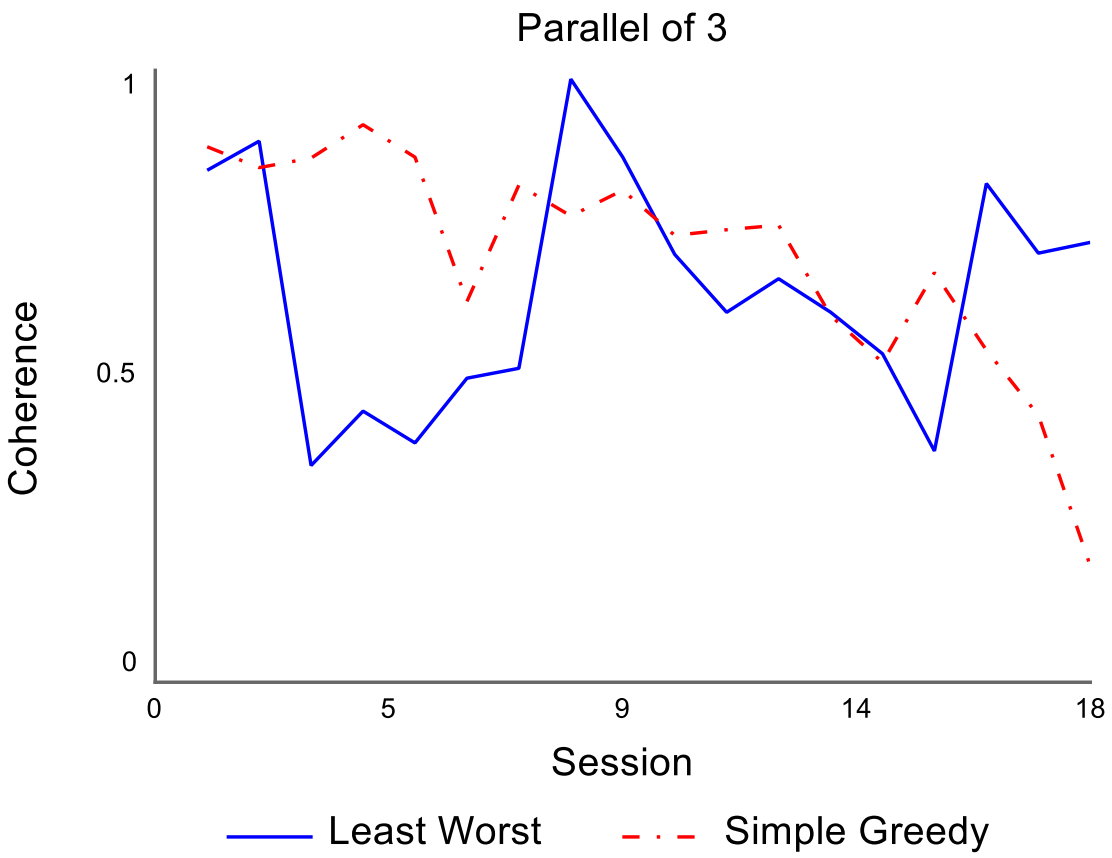

Figure 4. Coherence comparison for Parallel sessions

In our judgement, just by reading through the created schedules, the simple greedy algorithm was slightly superior, and this is consistent with Figure 4. For 2024 the simple greedy algorithm did seem to produce a reasonable set of sessions as noted below.

Experience

For 2024 we had 454 submissions needing review. This was larger than expected and meant that there was more work for both assessment and session creation.

Assessment

There were a total of 41 Thread Chairs who were assigned, on average 22 reviews, though the actual range was from 7 to 30 as shown in Figure 5. Each submission was assigned one Thread Chair from the thread in which it was submitted, and a second Thread Chair from (potentially) a different thread. These secondary assignments were based on the alternative threads for which each Thread Chair had indicated a willingness to review for. The assignment process was automated, using the same algorithm that peer reviewers are assigned with, which attempts to assign an equal number to each Thread Chair. The large

variation in the number of assignments was the result of the need to get a at least one same-thread reviewer and the number of alternate threads each Thread Chair had selected. There was also a lock-in effect that exacerbated the problem as the reviews were assigned on a rolling basis, as well as initial programming errors that over assigned some reviewers and could only be partially corrected.

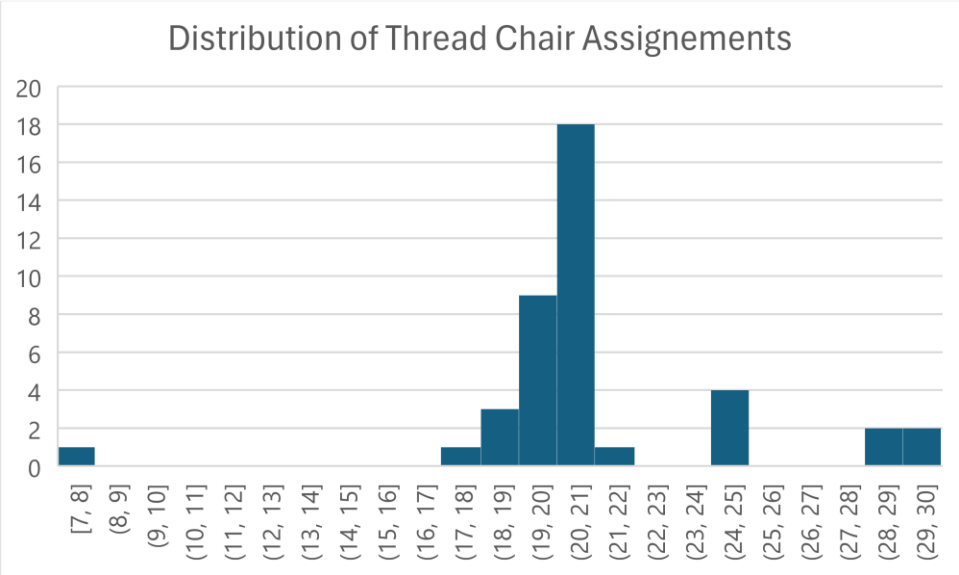

Figure 5. Distribution of actual review assignments

Each Thread Chair assigned a score of 1 to 10 to the submissions they reviewed. All but 30 of the assigned reviews were completed.

There were also approximately 1200 peer reviews assigned to about 500 peer reviewers, and this information was also used for assessment. About 1000 of these reviews were completed.

The Thread Chair and peer review scores were then used in the assessment phase of the program development described above. There were two approaches to doing this. One was a weighted average of the scores from Thread Chairs and peer reviewers with most of the weight given the Thread Chairs. The second was introduced by Hazhir Rahmandad, one of the Program Chairs, and is a fixed factor (see for example Uno et al 2016) analysis . this approach takes the various scores and creates a ranking based on the inferred consistency of the scores. In essence this assigns weights to the scores that are high when the scorer is consistent with other scorers, and low when they are not. The logic for doing this computation is shown in Figure 6. The parameters shown in green are optimized in order to minimize the difference between the predicted and actual scores. Underlying quality is then used to rank the papers.

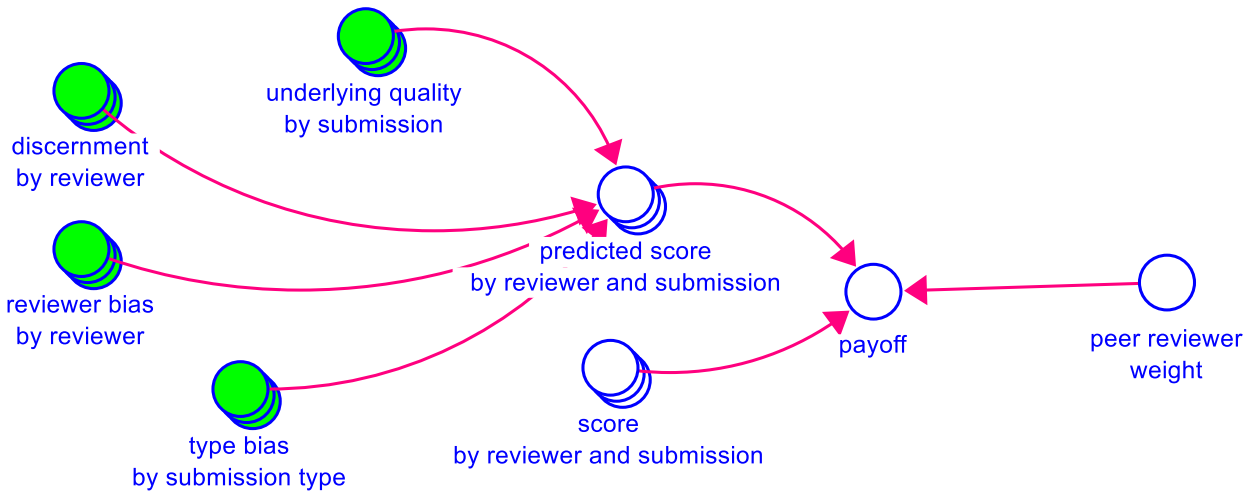

Figure 6. Logic for inferring the underlying submission quality

In Figure 6 the predicted score is just a linear equation:

type_bias + reviewer_bias + discernment * underlying_quality (2)

The payoff then compares that to the score recorded. Predicted scores are computed for all types, submissions and all reviewers (about 60,000 possibilities), but the actual scores used for comparison are only about 2,600. With 2,600 data points and almost 1,000 parameters to estimate it should be clear that the uncertainty around the parameter estimates is fairly large, though this is not something we tried to compute. In the end it is only the underlying quality estimate that is of interest.

The weighted averaging versus the fixed factor approaches (as well as a number of variations on each of them) yielded similar, but distinct results. While most submissions lined up well, there were a handful that moved dramatically in a sorted list based on the two approaches. There was also a notable sensitivity to adding a small number of additional scores (small numbers problem). Ultimately, the Program Chairs used the fixed factor approach to do the initial ranking, then read through the submissions to make the final assessments for placements.

Session Creation

The session creation followed the simple greedy algorithm discussed above and seemed to work quite well. The submission portal was configured so that the algorithm could be run multiple times, with any manual adjustments of sessions retained so that only the unassigned submissions were automatically organized. The weights in doing submission comparison were distinguished by thread, keywords, and title. The abstract ignored in the weighting due to significant overlap (differences seem to be related to language not

content). The most weight (60%) was given to threads, with keywords and title each getting 20%. The consistency of keyword usage was lower than expected and this was likely partly due the ease with which anyone could add keywords. Ultimately, we ended up with about 1500 keywords which was too many given the number of submissions.

It is a little bit difficult to assess how many changes were made to the program after the algorithm ran and why they were made. While the Program Chairs described the initial sessioninzing as quite good, a cursory inspection of the webportal logs suggests that about 100 submissions were moved around. So, while the bulk of work may have found its home with the automated process, there was still substantial manual work done. Session titles were also left to the Program Chairs, as the automatically created titles simply indicated the Threads of the included submissions.

Considerations

Going into the 2024 conference our main concerns were around process and engagement. Though this approach is designed to automate more of the program creation process, it places almost all of the remaining burden on the Program Chairs rather than the Thread Chairs. This, in a sense, means one fewer set of eyes on the program, which may let anomalies sneak through. On the other hand, anomalies have always been part of the program, so on balance it seems unlikely to make things worse.

The other important downside of this approach is the highly routine nature of the Thread Chairs' jobs that results from it. While we believe this does relieve some of the burden, it also removes much of the benefit. As difficult as it is to coordinate among a number of different Thread Chairs in different time zones, such coordination does result in learning.

Conclusions and Lessons

Overall, the more automated process worked quite well, and seems to have made managing such a large number of submissions easier than it otherwise would have been. The process for creating, organizing, and adjusting sessions seemed to work well and smooth the overall creation process. It is possible for future conferences we may want to experiment with adjustments to the algorithm to further reduce the burden on the Program Chairs, but that itself requires engagement with the Program Chairs, which could be considered an additional burden.

The assessment process likely needs more adjustment. In particular, by changing the requirement from recommending placement of submitted work to providing a simple score

the Thread Chairs became more distant to the submitted work. Also, being asked to review work outside of their thread was uncomfortable for many of them as they did not know on what basis they should be making judgements.

To correct this, and also help with the engagement issues discussed under considerations above, we are proposing changes to the assessment process. First, rather than assigning submissions outside of the Thread Chairs primary thread, we suggest that all submissions have two Thread Chairs for the submitted thread score them. In doing so, rather than a simple numeric score, the scores will be annotated to indicate recommended placement.

Following that score exercise, rather than turning over a large, sorted list of submissions to the Program Chairs, we suggest that the Thread Chairs for each thread convene to rank the submissions within their thread and provide recommendations for the placement cutoffs. Since all submissions will be reviewed by two Thread Chairs within the thread, those conversations will not need to be based on a single individual's assessment. This moves us somewhat more toward the idea that every brick has a bricklayer mentioned above.

The Program Chairs will then work with a list of submissions ranked within threads, with recommended cutoff points that will be different between threads. They can make thread specific adjustments to the cutoff, or more likely simply make session type~~s~~ assignments for each submission. Once the assignments have been decided upon, they are simply uploaded into the webportal.

In all, the increased automation seems to have helped in bringing together the 2024 program. With the noted adjustments it can continue to serve us in the years to come.

Bibliography

Gündoğan, E. and M. Kaya, 2023, “Automatically organizing papers in conference sessions using deep learning and network modeling.” Multimedia Tools and Applications. https://doi.org/10.1007/s11042-023-17460-w

Joulin, Armand, E. Grave, P. Bojanowski, and T. Mikolov, 2016, “Bag of Tricks for Efficient Text Classification, arXiv:1607.01759, https://doi.org/10.48550/arXiv.1607.01759

Uno, Kohei, Hironori Satomura, Kohei Adachi, 2016, "Fixed factor analysis with clustered factor score constraint.” *Computational Statistics & Data Analysis,* Volume 94, Pages 265-274, https://doi.org/10.1016/j.csda.2015.08.010.