Reputation in the Academic World

Nardine Osman   Carles Sierra

Artificial Intelligence Research Institute (IIIA-CSIC), Barcelona, Spain
{nardine, sierra}@iiia.csic.es

Abstract

With open access gaining momentum, **open reviews** becomes a more persistent issue. Institutional and multidisciplinary open access repositories play a crucial role in knowledge transfer by enabling immediate accessibility to all kinds of research output. However, they still lack the quantitative assessment of the hosted research items that will facilitate the process of selecting the most relevant and distinguished content. This paper addresses this issue by proposing a computational model based on peer reviews for assessing the reputation of researchers and their research work. The model is developed as an overlay service to existing institutional or other repositories. We argue that by relying on peer opinions, we address some of the pitfalls of current approaches for calculating the reputation of authors and papers. We also introduce a much needed feature for review management, and that is calculating the reputation of reviews and reviewers.

1 Motivation

There has been a strong move towards **open access** repositories in the last decade or so. Many funding agencies — such as the UK Research Councils, Canadian funding agencies, American funding agencies, the European Commission, as well as many universities — are promoting open access by requiring the results of their funded projects to be published in open access repositories. It is a way to ensure that the research they fund has the greatest possible research impact. Academics are also very much interested in open access repositories, as this helps them maximise their research impact. In fact, studies have confirmed that open access articles are more likely to be used and cited than those sitting behind subscription barriers [Eys06]. As a result, a growing number of open access repositories are becoming extremely popular in different fields, such as PLoS ONE for Biology, arXiv for Physics, and so on.

With open access gaining momentum, **open reviews** becomes a more persistent issue. Institutional and multidisciplinary open access repositories play a crucial role in knowledge transfer by enabling immediate accessibility to all kinds of research output. However, they still lack the quantitative assessment of the hosted research items that will facilitate the process of selecting the most relevant and distinguished content. Common currently available metrics, such as number of visits and downloads, do not reflect the quality of a research product, which can only be assessed directly by peers offering their expert opinion together with quantitative ratings based on specific criteria. The articles published in the Frontiers book [KD12] highlight the need for open reviews.

To address this issue we develop an open peer review module, the Academic Reputation Model (ARM), as an overlay service to existing institutional or other repositories. Digital research works hosted in repositories using

In: J. Zhang, R. Cohen, and M. Sensoy (eds.): Proceedings of the 18th International Workshop on Trust in Agent Societies, Singapore, 09-MAY-2016, published at http://ceur-ws.org

our module can be evaluated by an unlimited number of peers that offer not only a qualitative assessment in the form of text, but also quantitative measures to build the works reputation. Crucially, our open peer review module also includes a reviewer reputation system based on the assessment of reviews themselves, both by the community of users and by other peer reviewers. This allows for a sophisticated scaling of the importance of each review on the overall assessment of a research work, based on the reputation of the reviewer.

As a result of calculating the reputation of authors, reviewers, papers, and reviews, by relying on peer opinions, we argue that the model addresses some of the pitfalls of current approaches for calculating the reputation of authors and papers. It also introduces a much needed feature for review management, and that is calculating the reputation of reviews and reviewers. This is discussed further in the concluding remarks.

In what follows, we present the ARM reputation model and how it quantifies the reputation of papers, authors, reviewers, and reviews (Section 2), followed by some evaluation where we use simulations to evaluate the correctness of the proposed model (Section 3), before closing with some concluding remarks (Section 4).

2 ARM: The Academic Reputation Model

2.1 Data and Notation

In order to compute reputation values for papers, authors, reviewers, and reviews we require a *Reputation Data Set*, which in practice should be extracted from existing paper repositories.

Definition 2.1 (Data)**.** A *Reputation data Set* is a tuple $\langle P, R, E, D, a, o, v\rangle$, where

- $P = \{p_i\}_{i\in\mathcal{P}}$ is a set of papers (e.g. DOIs).
- $R = \{r_j\}_{j\in\mathcal{R}}$ is a set of researcher names or identifiers (e.g. the ORCHID identifier).
- $E = \{e_i\}_{i\in\mathcal{E}} \cup \{\bot\}$ is a totally ordered evaluation space, where $e_i \in \mathbb{N} \setminus \{0\}$ and $e_i < e_j$ iff $i < j$ and $\bot$ stands for the absence of evaluation. We suggest the range [0,100], although any other range may be used, and the choice of range will not affect the performance.
- $D = \{d_k\}_{k\in\mathcal{K}}$ is a set of evaluation dimensions, such as *originality*, *technical soundness*, etc.
- $a : P \to 2^R$ is a function that gives the authors of a paper.
- $o : R \times P \times D \times \mathit{Time} \to E$, where $o(r, p, d, t) \in E$ is a function that gives the opinion of a reviewer, as a value in E, on a dimension d of a paper p at a given instant of time t.
- $v : R \times R \times P \times \mathit{Time} \to E$, where $v(r, r', p, t) = e$ is a function that gives the judgement of researcher r over the opinion of researcher r', on paper p as a value $e \in E$.[1] Therefore, a judgement is a reviewer's opinion about another reviewer's opinion. Note that while opinions about a paper are made with respect to a given dimension in D, judgements are not related to dimensions. We assume a judgement is only made with respect to one dimension, which describes how good the review is *in general.*

We will not include the dimension (or the criteria being evaluated, such as originality, soundness, etc.) in the equations to simplify the notation. There are no interactions among dimensions so the set of equations apply to each of the dimensions under evaluation.

Also, we will also omit the reference to time in all the equations. Time is essential as all measures are dynamic and thus they evolve along time. We will make the simplifying assumption that all opinions and judgements are maintained in time, that is, they are not modified. Including time would not change the essence of the equations, it will simply make the computation complexity heavier.

Finally, if a data set allowed for papers, reviews, and/or judgements to have different versions, then our model simply considers the latest version only.

[1] In tools like ConfMaster (www.confmaster.net) this information could be gathered by simply adding a private question to each paper review, answered with elements in E, one value in E for the judgement on each fellow reviewer's review.

2.2 Reputation of a Paper

We say the reputation of a paper is a weighted aggregation of its reviews, where the weight is the reputation of the reviewer. (Section 2.4).

$$R_P(p) = \begin{cases} \dfrac{\sum\limits_{\forall r \in rev(p)} R_R(r) \cdot o(r,p)}{\sum\limits_{\forall r \in rev(p)} R_R(r)} & \text{if } |rev(p)| \geq k \\ \perp & \text{otherwise} \end{cases} \tag{1}$$

where $rev(p) = \{r \in R \mid o(r,p) \neq \perp\}$ denotes the reviewers of a given paper.

Note that when a paper receives less that k reviews, its reputation is defined as unknown, or $\perp$. We currently leave k as a parameter, though we suggest that $k > 1$, so that the reputation of a paper is not dependent on a single review. We also recommend small numbers for k, such as 2 or 3, because we believe it is usually difficult to obtain reviews. As such, new papers can quickly start building a reputation.

2.3 Reputation of an Author

We consider that a researcher's author reputation is an aggregation of the reputation of her papers. The aggregation is based on the concept that *the impact of a paper's reputation on its authors' reputation is inversely proportional to the total number of its authors.* In other words, if one researcher is the sole author of a paper, then this author is the only person responsible for this paper, and any (positive or negative) feedback about this paper is propagated as is to its sole author. However, if the researcher has co-authored the paper with several other researchers, then the impact (whether positive or negative) that this paper has on the researcher decreases with the increasing number of co-authors. We argue that collaborating with different researchers usually increases the quality of a research work since the combined expertise of more than one researcher is always better than the expertise of a single researcher. Nevertheless, the gain in a researcher's reputation decreases as the number of co-authors increase. Hence, our model might cause researchers to be more careful when selecting their collaborators, since they should aim at increasing the quality of the papers they produce in such a way that the gain for each author is still larger than the gain it could have received if it was to work on the same research problem on her own. As such, adding authors who do not contribute to the quality of the paper will also discouraged.

$$R_A(r) = \begin{cases} \dfrac{\sum\limits_{\forall p \in pap(r)} \gamma(p)^{\gamma} \times R_P(p) + (1 - \gamma(p)^{\gamma}) \times 50}{|pap(r)|} & \text{if } pap(r) \neq \emptyset \\ \perp & \text{otherwise} \end{cases} \tag{2}$$

where $pap(r) = \{p \in P \mid r \in a(p) \land R_P(p) \neq \perp\}$ denotes the papers authored by a given researcher r, $\perp$ describes ignorance, $\gamma(p) = \dfrac{1}{|a(p)|}$ is the coefficient that takes into consideration the number of authors of a paper (recall that $a(p)$ denotes the authors of a paper p), and γ is a tuning factor that controls the rate of decrease of the $\gamma(p)$ coefficient. Also note the multiplication by 50, which describes ignorance, as 50 is the median of the chosen range $[0, 100]$. If another range was chosen, the median of that range would be used here. The choice of range and its median does not affect the performance of the model (i.e. the results of the simulation of Section 3 would remain the same).

2.4 Reputation of a Reviewer

Similar to the reputation of authors (Section 2.3), we consider that if a reviewer produces 'good' reviews, then the reviewer is considered to be a 'reputed' reviewer. Furthermore, we consider that the reputation of a reviewer is essentially an aggregation of the opinions over her reviews.[2]

We assume that the opinions on how good a review is can be obtained, in a first instance, by other reviewers that *also reviewed the same paper.* However, as this is a new feature to be introduced in open access repositories and conference and journal paper management systems, we believe collecting such information might take some

[2] We assume a review can only be written by one reviewer, and as such, the number of co-authors of a review is not relevant as it was when calculating the reputation of authors.

time. An alternative that we consider here is that in the meantime we can use the 'similarity' between reviews as a measure of the reviewers opinions about reviews. In other words, the heuristic could be phrased as 'if my review is similar to yours then I may assume your judgement of my review would be good.'

We note $v^*(r_i, r_j, p) \in E$ for the 'extended judgement' of r_i over r_j's opinion on paper p, and define it as an aggregation of opinions and similarities as follows:

$$v^*(r_i, r_j, p) = \begin{cases} v(r_i, r_j, p) & \text{if } v(r_i, r_j, p) \neq \bot \\ Sim(\bar{o}(r_i, p), \bar{o}(r_j, p)) & \text{If } \bar{o}(r_i, p) \neq \bot \text{ and } \bar{o}(r_j, p) \neq \bot \\ \bot & \text{Otherwise} \end{cases} \tag{3}$$

where Sim stands for an appropriate similarity measure. We say the similarity between two opinions is the difference between the two: $Sim(\bar{o}(r_i, p), \bar{o}(r_j, p)) = 100 - |\bar{o}(r_i, p) - \bar{o}(r_j, p)|$.

Given this, we consider that the overall opinion of a researcher on the capacity of another researcher to make good reviews is calculated as follows. Consider the set of judgements of r_i over reviews made by r_j as: $V^*(r_i, r_j) = \{v^*(r_i, r_j, p) \mid v(r_i, r_j, p) \neq \bot \text{ and } p \in P\}$. This set might be empty. Then, we define the judgement of a reviewer over another one as a simple average:

$$R_R(r_i, r_j) = \begin{cases} \dfrac{\sum\limits_{\forall v \in V^*(r_i, r_j)} v}{|V^*(r_i, r_j)|} & \text{if } V^*(r_i, r_j) \neq \emptyset \\ \bot & \text{otherwise} \end{cases} \tag{4}$$

Finally, the reputation of a reviewer r, $R_R(r)$, is an aggregation of judgements that her colleagues make about her capability to produce good reviews. We weight this with the reputation of the colleagues as a reviewer:

$$R_R(r) = \begin{cases} \dfrac{\sum\limits_{\forall r_i \in R^*} R_R(r_i) \cdot R_R(r_i, r)}{\sum\limits_{\forall r_i \in R^*} R_R(r_i)} & R^* \neq \emptyset \\ 50 & \text{otherwise} \end{cases} \tag{5}$$

where $R^* = \{r_i \in R \mid V^*(r_i, r) \neq \emptyset\}$. When no judgements have been made over r, we take the value 50 to represent ignorance (as 50 is the median of the chosen range $[0, 100]$ — again, we note that any the choice of range and its median does not affect the performance of the model; that is, the results of the simulation of Section 3 would remain the same).

Note that the reputation of a reviewer depends on the reputation of other reviewers. In other words, every time the reputation of one reviewer will change, it will trigger changing the reputation of other reviewers, which might lead to an infinite loop of modifying the reputation of reviewers. We address this by using an algorithm similar to the EigenTrust algorithm, as illustrated by Algorithm 4 of the Appendix. In fact, this algorithm may be considered as a variation of the EigenTrust algorithm, which will require some testing to confirm how fast it converges.

2.5 Reputation of a Review

The reputation of a review is similar to the one for papers but using judgements instead of opinions. We say the reputation of a review is a weighted aggregation of its judgements, where the weight is the reputation of the reviewer (Section 2.4).

$$R_O(r', p) = \begin{cases} \dfrac{\sum\limits_{\forall r \in jud(r', p)} R_R(r) \cdot v^*(r, r', p)}{\sum\limits_{\forall r \in jud(r', p)} R_R(r)} & \text{if } |jud(r', p)| \geq k \\ R_R(r') & \text{otherwise} \end{cases} \tag{6}$$

where $jud(r', p) = \{r \in R \mid v^*(r, r', p) \neq \bot\}$ denotes the set of judges of a particular review written by r' on a given paper p.

Note that when a review receives less that k judgements, its reputation will not depend on the judgements, but it will inherit the reputation of the author of the review (her reputation as a reviewer).

We currently leave k as a parameter, though we suggest that $k > 1$, so that the reputation of a review is not dependent on a single judge. Again, we recommend small numbers for k, such as 2 or 3, because we believe it will be difficult to obtain large numbers of judgements.

2.6 A Note on Dependencies

Figure 1 shows the dependencies between the different measures (reputation measures, opinions, and judgements). The decision of When to re-calculate those measures is then based on those dependencies. We provide a summary of this below. Note that measures in white are not calculated, but provided by the users. As such, we only discuss those in grey (grey rectangles represent reputation measures, whereas the grey oval represents the extended judgements).

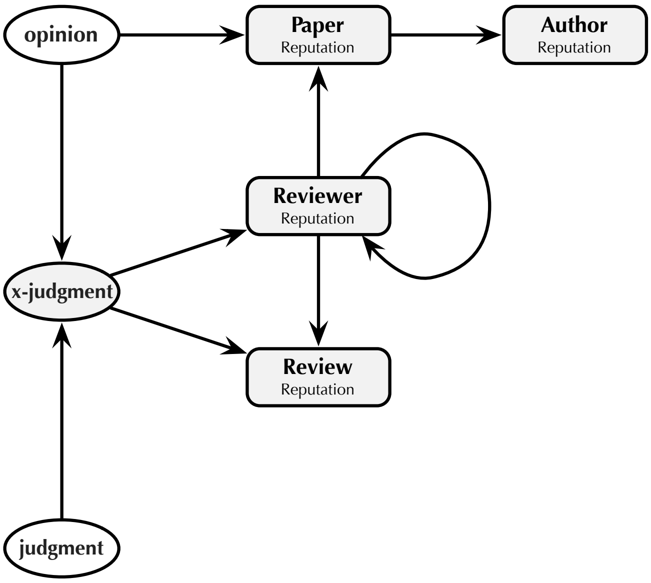


Figure 1: Dependencies

- **Author's Reputation.** The reputation of the author depends on the reputation of its papers (Equation 2). As such, every time the reputation of one of his papers changes, or every time a new paper is created, the reputation of the author must be recalculated (Algorithm 2 of the Appendix).

- **Paper's Reputation.** The reputation of the paper depends on the opinions it receives, and the reputation of the reviewers giving those opinions (Equation 1). As such, every time a paper receives a new opinion, or every time the reputation of one of the reviewers changes, then the reputation of the paper must be recalculated (Algorithm 1 of the Appendix).

- **Review's Reputation.** The reputation of a review depends on the extended judgements it receives, and the reputation of the reviewers giving those judgements (Equation 6). As such, every time a review receives a new extended judgements, or every time the reputation of one of the reviewers changes, then the reputation of the review must be recalculated (Algorithm 5 of the Appendix).

- **Reviewer's Reputation.** The reputation of a reviewer depends on the extended judgements of other reviewers and their reputation (Equation 5). As such, the reputation of the reviewer should be modified every time there is a new extended judgement or the reputation of on of the reviewers changes. As the reputation of a reviewer depends on the reputation of reviewers, then we suggest to calculate the reputation of all reviewers repeatedly (in a manner similar to EigenTrust) in order to converge (Algorithm 4 of the Appendix). If this will be computationally expensive, then this can be computed once a day, as opposed to triggered by extended judgements and the change in reviewers' reputation.

- **x-judgement.** The extended judgement is calculated either based on judgements (if available) or the similarity between opinions (when judgements are not available) (Equation 3). As such, the extended judgement should be recalculated every time a new (direct) judgement is made, or every time a new opinion is added on a paper which already has opinions by other reviewers (Algorithm 3 of the Appendix).

3 Evaluation through Simulation

3.1 Simulation

To evaluate the effectiveness of the proposed model, we have simulated a community of researchers, using NetLogo [TW04]. We clarify that the focus of this work is not implementing a simulation that models the real world, but a simulation that allows us to verify our model. As such, many assumptions that we make for this simulation, and will appear shortly, might not be precisely (or always) true in the real world (such as having the true quality of a paper inherit the quality of the best author).

In our simulation, a breed in NetLogo (or a node in the research community's graph) represents either a researcher, a paper, a review, or a judgement. The relations between breeds are: (1) *authors of*, that specifies which researchers are authors of a given paper, (2) *reviewers of*, that specifies which researchers are reviewers of a given paper, (3) *reviews of*, that specifies which reviews give opinions on a given paper, (4) *judgements of*, that specifies which judgements give opinions on a given review; and (5) *judges of*, that specifies which researchers have judged which other researcher.

Also, each researcher has four parameters that describe: (1) her reputation as an author, (2) her reputation as a reviewer, (3) her *true* research quality; and (4) her *true* reviewing quality. The first two are calculated by our ARM model, and they evolve over time. However, the last two describe the researcher's true quality with respect to writing papers as well as reviewing papers or other reviews, respectively. In other words, our simulation assumes true qualities exist, and that they are constant. In real life, there are no such measures. Furthermore, how good one is at writing papers or writing reviews or making judgements naturally evolves with time. Nevertheless, we chose to keep the simulation simple by sticking to constant true qualities, as the purpose of the simulation is simply to evaluate the correctness of our ARM model.

Similar to researchers, we say each paper has two parameters that describe it: (1) its reputation, which is calculated by our ARM model, and it evolves over time; and (2) its *true* quality. Again, we assume that a paper's true quality exists. How it is calculated is presented shortly.

Reviews also have two parameters: (1) the opinion provided by the review, which in real life is set by the researcher performing the review, while in our simulation it is calculated by the simulator, as illustrated shortly; and (2) the reputation of the review, which is calculated by our ARM model and it evolves over time.

Judgements, on the other hand, only have one parameter: the opinion provided by the judgement, which in real life is set by the researcher judging a review, while in our simulation it is calculated by the simulator, as illustrated shortly.

Simulation starts at time zero with no researchers in the community, and hence, no papers, no reviews, and no judgements. Then, with every tick of the simulation, a new paper is created, which may sometimes require the creation of new researchers (either as authors or reviewers). With the new paper, reviews and judgements are also created. How these elements are created is defined next by the simulator's parameters and methods, that drive and control this behaviour. We note that a tick of the simulation does not represent a fixed unit in calendar time, but the creation of one single paper.

The ultimate aim of the evaluation is to investigate how close are the calculated reputation values to the *true* values: the reputation of a researcher as an author, the reputation of a researcher as a reviewer, and the reputation of a paper.

The parameters and methods that drive and control the evolution of the community of researchers and the evolution of their research work are presented below.

1. *Number of authors.* Every time a new paper is created, the simulator assigns authors for this paper. How many authors are assigned is defined by the number of authors parameter ($\#_{co-authors}$), which is defined as a Poisson distribution. For every new paper, a random number is generated from this Poisson distribution. Who to assign is chosen randomly from the set of researchers, although sometimes, a new researcher is created and assigned to this paper (see the 'researchers birth rate' below). This ensures the number of researchers in the community grows with the number of papers.

2. *Number of reviewers.* Every time a new paper is created, the simulator also assigns reviewers for this paper. How many reviewers are assigned is defined by the number of reviewers parameter ($\#_{reviewers}$), which is defined as a Poisson distribution. For every new paper, a random number is generated from this Poisson distribution. As above, who to assign is chosen randomly from the set of researchers, although sometimes, a new researcher is created and assigned to this paper.

3. *Researchers birth rate.* As illustrated above, every paper requires authors and reviewers to be assigned to it. When assigning authors and reviewers, the simulation will decide whether to assign an already existing researcher (if any) or create a new researcher. This decision is controlled by the researchers birth rate parameter ($birth\_rate$), which specifies the probability of creating a new researcher.

4. *Researcher's true research quality.* The author's true quality is sampled from a beta distribution specified by the parameters α_A and β_A. We choose the beta distribution because it is a very versatile distribution which can be used to model several different shapes of probability distributions by playing with only two parameters, α and β.

5. *Researcher's true review quality.* The reviewer's true quality is sampled from a beta distribution specified by the parameters α_R and β_R. Again, the beta distribution is a very versatile distribution which can be used to model several different shapes of probability distributions by playing with only two parameters, as illustrated shortly by our experiments.

6. *Paper's true quality.* We assume that a paper's true quality is the true quality of its best author, that is, the author with the highest true research quality). We believe this assumption has some ground in real life. For instance, some behaviour (such as looking for future collaborators, selecting who to give a funding to, etc.) assumes researchers to be of a certain quality, and their research work to follow that quality respectively.

7. *Opinion of a Review.* The opinion presented by a review is specified as the paper's true quality plus some noise, where the noise depends on the reviewer's true quality. This noise is chosen randomly from the range $[-(100 - review\ quality)/2, +(100 - review\ quality)/2]$. In other words, the maximum noise that can be added for the worst reviewer (whose review quality is 0) is ±50, and the least noise that can be added for the best reviewer (whose review quality is 100) is 0.

8. *Opinion of a Judgement.* The value (or opinion) of a judgement on a review is calculated as the similarity between the review's value (opinion) and the judge's review value (opinion), where the similarity is defined by the metric distance as: $100 - |review - judge's\ review|$. Note that, for simplification, direct judgements have not been simulated, we only rely on indirect judgements.

3.2 Results

3.2.1 Experiment 1: The impact of the community's quality of reviewers

Given the above, we ran the simulator for 100 ticks (generating 100 papers). We ran the experiment over 6 different cases. In each, we had the following parameters fixed:

$\#_{co-authors} = 2$
$\#_{reviewers} = 3$
$birth\_rate = 3$
$\alpha_A = \beta_A = 1$
$k = 3$ (of Equations 1 and 6)
$\gamma = 1$ (of Equation 2)

The only parameters that changed where those defining the beta distribution of the reviewers' qualities. This experiment illustrated the impact of the community's quality of reviewers on the correctness of the ARM model.

The results of the simulation are presented by Figure 2. For each case, the distribution of the reviewers' true quality is illustrated to the right of the results. The results, in numbers, are also presented by Table 1. We notice that the least error is presented when the reviewers are all of relatively good quality, with the majority being great reviewers (Figure 2e). The errors start increasing as bad reviewers are added to the community (Figure 2c). They increase even further in both cases, when the quality of reviewers follows a uniform distribution (Figure 2a), as well as when the reviewers are equiprobably good or bad, with no average reviewers (Figure 2b). As soon as the majority of reviewers are of poor quality (Figure 2d), the errors increase even further, with the worst case being when good reviewers are absent from the community (Figure 2f). These results are not surprising. A paper's true quality is not something that can be measured, or even agreed upon. As such, the trust model depends on the opinions of other researchers. As a result, the better the reviewing quality of researchers, the more accurate the trust model will be, and vice versa.

The numbers of Table 1 illustrate how the error in the papers' reputation increases with the error in the reviewers' reputation, though at a smaller rate. One curious thing about these results is the constant error in the reputation of authors. The next experiment investigates this issue.

Last, but not least, we note that the error is usually stable. This is because every time a paper is created, all the reviews it receives and the judgements those reviews receive are created at the same simulation time-step. In other words, it is not the case that papers accumulate more reviews and judgements over time, for the error to decrease over time.

	Error in Reviewers' Reputation	**Error in Papers' Reputation**	**Error in Authors' Reputation**
$\alpha_R = 5$ and $\beta_R = 1$	$\sim$ 11 %	$\sim$ 2 %	$\sim$ 22 %
$\alpha_R = 2$ and $\beta_R = 1$	$\sim$ 23 %	$\sim$ 5 %	$\sim$ 23 %
$\alpha_R = 1$ and $\beta_R = 1$	$\sim$ 30 %	$\sim$ 7 %	$\sim$ 23 %
$\alpha_R = 0.1$ and $\beta_R = 0.1$	$\sim$ 34 %	$\sim$ 5 %	$\sim$ 22 %
$\alpha_R = 1$ and $\beta_R = 2$	$\sim$ 44 %	$\sim$ 8 %	$\sim$ 23 %
$\alpha_R = 1$ and $\beta_R = 2$	$\sim$ 60 %	$\sim$ 9 %	$\sim$ 20 %

Table 1: The results of experiment 1, in numbers

3.2.2 Experiment 2: The impact of co-authorship

In the second experiment, we investigate the impact of co-authorship on authors' reputation. We choose the two extreme cases from experiment 1, when there are only relatively good authors in the community ($\alpha = 5$ and $\beta_R = 1$), and when there are only relatively bad authors in the community ($\alpha = 5$ and $\beta_R = 1$). For each of these cases, we then change the number of co-authors, investigating three cases: $\#_{co-authors} = \{0, 1, 2\}$. All other parameters remain set to those presented in experiment 1 above.

The results of this experiment are presented by Figure 3. The numbers are presented in Table 2. The results show that the error in the reviewers and papers reputation almost does not change for different numbers of co-authors. However, the error in the reputation of authors does. When there are no co-authors ($\#_{co-authors} = 0$), the error in authors' reputation is almost equal to the error in papers' reputation (Figures 3a and 3b). As soon as 1 co-author is added ($\#_{co-authors} = 0$), the error in authors' reputation increases (Figures 3c and 3d). When 2 co-authors are added ($\#_{co-authors} = 2$), the error in authors' reputation reaches the maximum, around 20–22% (Figures 3e and 3f). In fact, unreported results show that the error in authors' reputation is almost the same in all cases for $\#_{co-authors} \geq 2$.

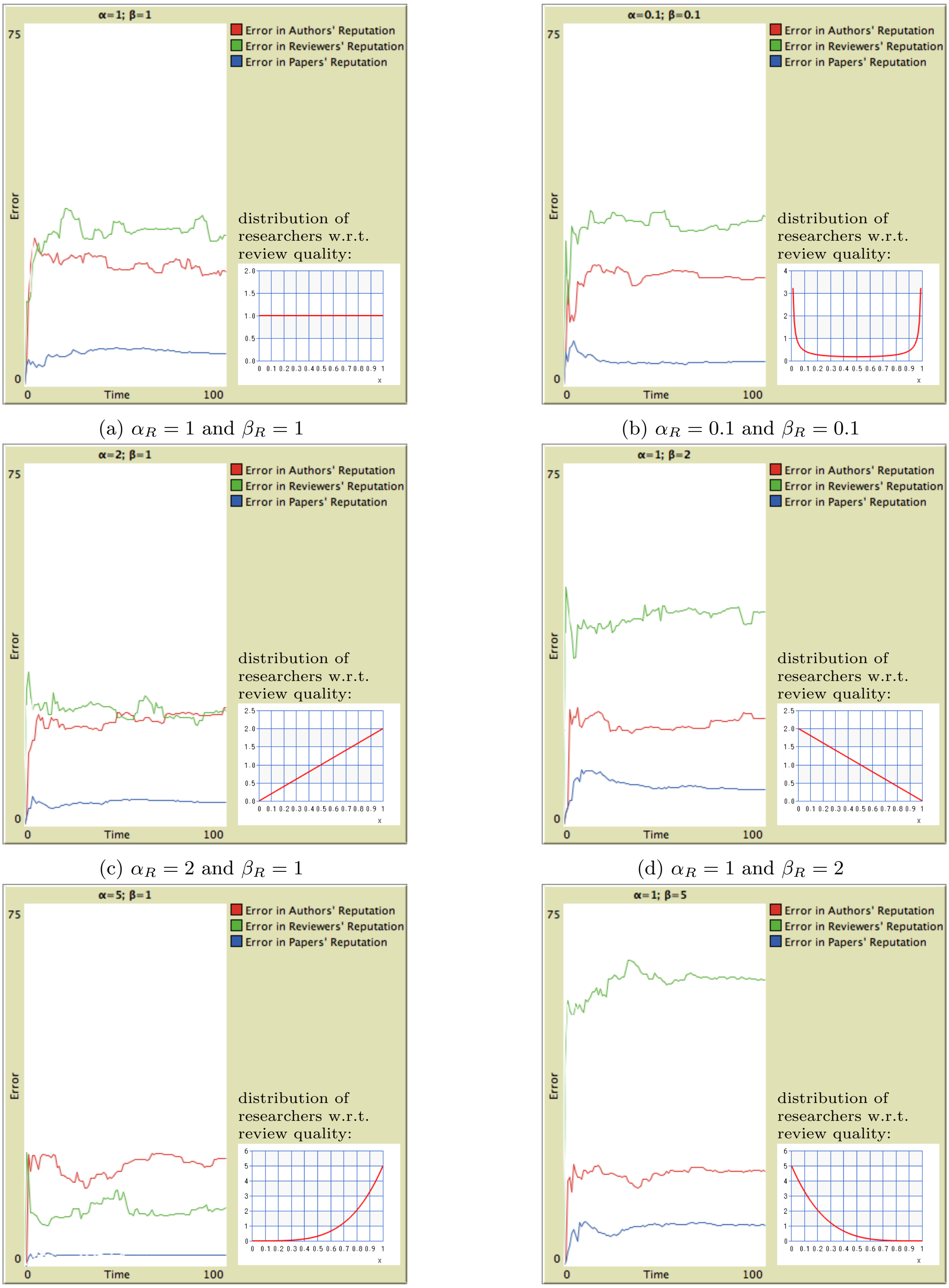
	Error in Reviewers' Reputation		**Error in Papers' Reputation**		**Error in Authors' Reputation**	
	α_R=5; β_R=1	α_R=1; β_R=5	α_R=5; β_R=1	α_R=1; β_R=5	α_R=5; β_R=1	α_R=1; β_R=5
$\#_{co-authors} = 0$	$\sim$ 11 %	$\sim$ 60 %	$\sim$ 2 %	$\sim$ 9 %	$\sim$ 22 %	$\sim$ 20 %
$\#_{co-authors} = 1$	$\sim$ 13 %	$\sim$ 57 %	$\sim$ 3 %	$\sim$ 9 %	$\sim$ 12 %	$\sim$ 15 %
$\#_{co-authors} = 2$	$\sim$ 13 %	$\sim$ 54 %	$\sim$ 3 %	$\sim$ 9 %	$\sim$ 2 %	$\sim$ 7 %

Table 2: The results of experiment 2, in numbers

4 Conclusion

We have presented the ARM reputation model for the academic world. ARM helps calculate the reputation of researchers, both as authors and reviewers, and their research work. Additionally, ARM also calculates the reputation of reviews.

Concerning the reputation of authors, the most commonly used reputation measure is currently the h-index [Hir05]. However, the h-index has its flaws. For instance, the h-index can be manipulated through self-citations [BK10, FR13]. A study has also found the h-index as not providing a significantly more accurate measure of impact than the total number of citations [Yon14]. ARM, on the other hand, bases the reputation of authors on the opinions that their papers receive from other members in their academic community. We believe this should be a more accurate approach, though future work should aim at comparing both approaches.



(a) $\alpha_R = 1$ and $\beta_R = 1$

(b) $\alpha_R = 0.1$ and $\beta_R = 0.1$

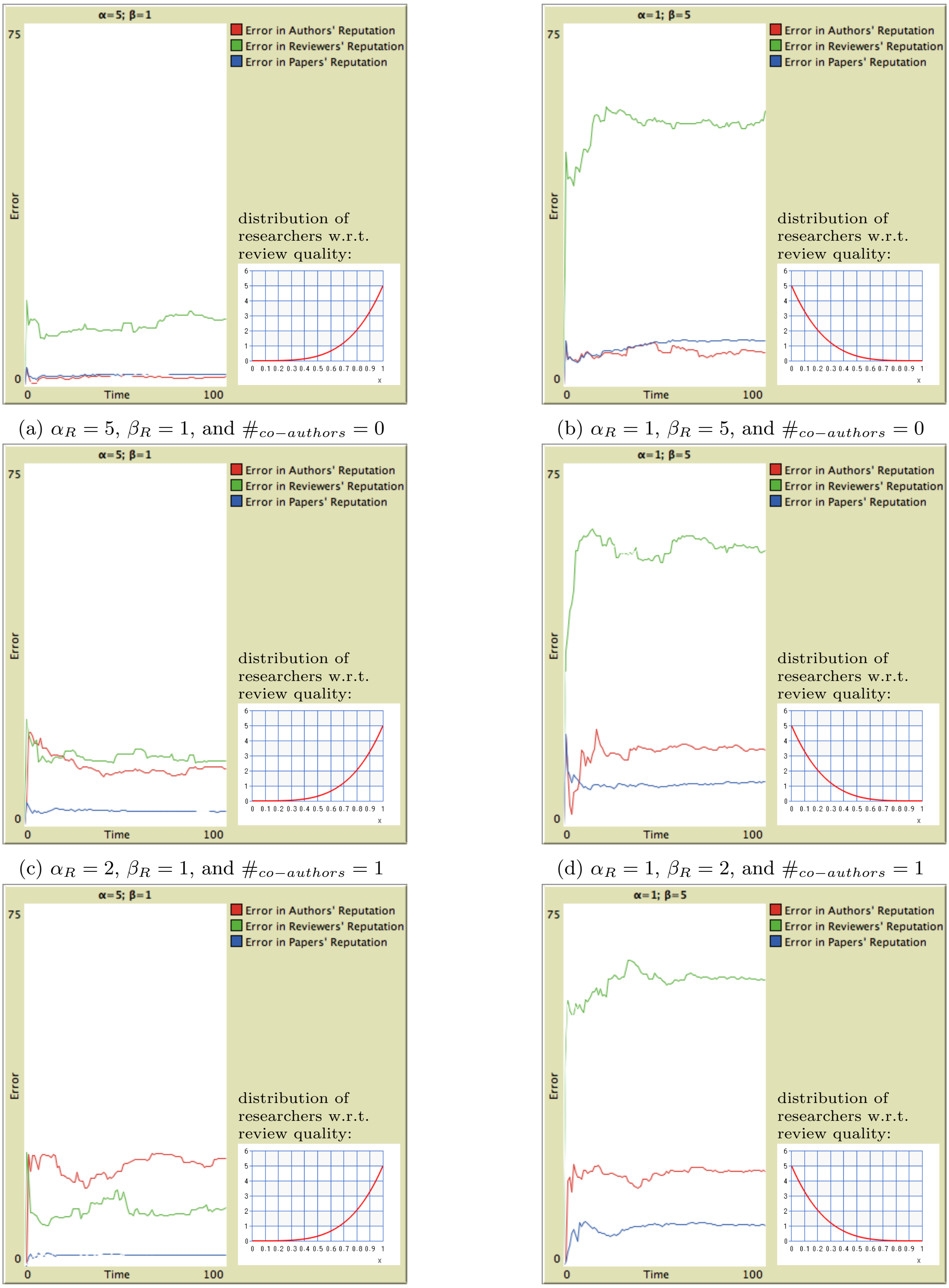
(c) $\alpha_R = 2$ and $\beta_R = 1$

(d) $\alpha_R = 1$ and $\beta_R = 2$

(e) $\alpha_R = 5$ and $\beta_R = 1$

(f) $\alpha_R = 1$ and $\beta_R = 5$

Figure 2: The impact of reviewers' quality on reputation measures. For each set of results, the distribution of the reviewers' true quality is presented to the right of the results.



(a) $\alpha_R = 5$, $\beta_R = 1$, and $\#_{co-authors} = 0$

(b) $\alpha_R = 1$, $\beta_R = 5$, and $\#_{co-authors} = 0$

(c) $\alpha_R = 2$, $\beta_R = 1$, and $\#_{co-authors} = 1$

(d) $\alpha_R = 1$, $\beta_R = 2$, and $\#_{co-authors} = 1$

(e) $\alpha_R = 5$, $\beta_R = 1$, and $\#_{co-authors} = 2$

(f) $\alpha_R = 1$, $\beta_R = 5$, and $\#_{co-authors} = 2$

Figure 3: The impact of co-authorship on reputation of authors. For each set of results, the distribution of the reviewers' true quality is presented to the right of the results.

Concerning the reputation of papers, the most common measure currently used is the total number of citations a paper gets. Again, this measure can easily be manipulated through the self-citations. [OSSM10] presents an alternative approach based on the propagation of opinions in structural graphs. It allows papers to build reputation either from the direct reviews it receives, or inherit reputation from the place where the paper is published. In fact, a sophisticated propagation model is proposed to allow reputation to propagate upwards as well as downwards in structural graphs (e.g. from a section to a chapter to a book, and vice versa). Simulations presented in [OSMSMC12] illustrate the potential impact of this model. ARM does not have any notion of propagation. The model is strictly based on direct opinions (reviews and judgements), and when no opinions are present, ignorance is assumed (as in the default reputation of authors and papers).

Concerning the reputation of reviews and reviewers, to our knowledge, these reputation measures have not been addressed yet. Nevertheless, we believe these are important measures. Conference management systems are witnessing a massive increase in paper submissions, and in many disciplines, finding good reviewers is becoming a challenging task. Deciding what papers to accept/reject is sometimes a challenge for conference and workshop organisers. ARM is a reputation model that addresses this issue by helping recognise the good reviews/reviewers from the bad.

The obvious next steps for ARM is applying it to a real dataset. In fact, the model is currently being integrated with two Spanish repositories: DIGITAL.CSIC (https://digital.csic.es) and e-IEO (http://www.repositorio.ieo.es/e-ieo/). However, these repositories do not have any opinions or judgements yet, and as such, time is needed to start collecting this data. We are also working with the IJCAI 2017 conference (http://ijcai-17.org) in order to allow reviewers to review each other. We will collect the data of this conference, which will provide us with the reviews and judgements needed for evaluating our model. We will also continue to look through existing datasets.

Future work can investigate a number of additional issues. For instance, we plan to provide data on the convergence performance of the algorithm. One can also study the different types of attacks that could impact the proposed computational model. While similarity of reviews is now computed based on the similarity of the quantitative opinions, the similarity between qualitative opinions may also be used in future work by making use of natural language processing techniques. Also, while we argue that direct opinion can help the model avoid the pitfalls of the literature, it is also true that direct opinions are usually scarce. As such, if needed, other information sources for opinions may also be considered, such as citations. This information can be translated into opinions, and the equations of ARM should then change to give more weight to direct opinions than other information sources.

4.0.1 Acknowledgements

This work has been supported by CollectiveMind (a project funded by the Spanish Ministry of Economy & Competitiveness (MINECO), grant # TEC2013-49430-EXP), and Open Peer Review Module for Repositories (a project funded by OpenAIRE, which in turn is an EU funded project).

References

[BK10] Christoph Bartneck and Servaas Kokkelmans. Detecting h-index manipulation through self-citation analysis. *Scientometrics*, 87(1):85–98, 2010.

[Eys06] Gunther Eysenbach. Citation advantage of open access articles. *PLoS Biology*, 4(5):e157, 05 2006.

[FR13] Emilio Ferrara and Alfonso E. Romero. Scientific impact evaluation and the effect of self-citations: Mitigating the bias by discounting the h-index. *Journal of the American Society for Information Science and Technology*, 64(11):2332–2339, 2013.

[Hir05] J. E. Hirsch. An index to quantify an individual's scientific research output. *Proceedings of the National Academy of Sciences of the United States of America*, 102(46):16569–16572, 2005.

[KD12] Nikolaus Kriegeskorte and Diana Deca, editors. *Beyond open access: visions for open evaluation of scientific papers by post-publication peer review*, Frontiers in Computational Neuroscience. Frontiers E-books, November 2012.

[OSMSMC12] Nardine Osman, Jordi Sabater-Mir, Carles Sierra, and Jordi Madrenas-Ciurana. Simulating research behaviour. In *Proceedings of the 12th International Conference on Multi-Agent-Based Simulation*, MABS'11, pages 15–30, Berlin, Heidelberg, 2012. Springer-Verlag.

[OSSM10] Nardine Osman, Carles Sierra, and Jordi Sabater-Mir. Propagation of opinions in structural graphs. In *Proceedings of the 2010 Conference on ECAI 2010: 19th European Conference on Artificial Intelligence*, pages 595–600, Amsterdam, The Netherlands, The Netherlands, 2010. IOS Press.

[TW04] Seth Tisue and Uri Wilensky. Netlogo: Design and implementation of a multi-agent modeling environment. In *In Proceedings of the Agent Conference*, pages 161–184, 2004.

[Yon14] Alexander Yong. Critique of hirschs citation index: A combinatorial fermi problem. *Notices of the American Mathematical Society*, 61(11):1040–1050, 2014.

A The Algorithms

Algorithm 1: Reputation of a paper

```
Function ReputationPaper(p : P):[0,100]=
    Data: $p : P$ /* a paper identifier */
    Data: $aut : P \to R\ list$ /* function returning the list of authors of papers */
    Data: $o : (R \times E)\ list$ /* list of evaluations of reviewers over paper $p$ */
    Data: $k : integer$ /* minimum number of reviewers to compute non-default reputation $k > 1$ */
    Result: $RepPaper : [0, 100]$ /* the reputation value of paper $p$ */
    /* This function computes the reputation of a paper for a given dimension.  It must be
       called every time a new review is created for this paper, and every time the
       reputation of one of the paper's reviewers is modified. */
    $rev = \emptyset$;
    for $(r, e) \in o$ do
        if $R_R(r) \neq null$ then
            $rev = rev \cup (r, e)$;
        end
    end
    if $length(rev) < k$ then
        $RepPaper \longleftarrow null$;
    else
        $normal \longleftarrow 0$;
        for $(r, e) \in rev$ do
            $normal \longleftarrow normal + ReputationReviewer(r)$;
        end
        $num \longleftarrow 0.0$;
        for $(r, e) \in rev$ do
            $num \longleftarrow num + ReputationReviewer(r) * e$;
        end
        $RepPaper \longleftarrow num/normal$;
    end
    return RepPaper;
```

Algorithm 2: Reputation of an author

```
Function ReputationAuthor(r : R):[0,100]=
    Data: $r : R$ /* a researcher identifier */
    Data: $pap : R \to P\ list$ /* function returning the list of papers of authors */
    Data: $aut : P \to R\ list$ /* function returning the list of authors of papers */
    Data: $alpha : real$ /* tuning factor for coefficient gamma */
    Result: $RepAuthor : [0, 100]$ /* the reputation value of author $r$ */
    /* This function computes the reputation of an author. It must be called every time a
       new paper is created for this author, and every time the reputation of one of the
       author's papers is modified. */
    $pap2 = \emptyset$;
    for $p \in pap(r)$ do
        if $R_P(p) \neq null$ then
            $pap2 = pap2 \cup p$;
        end
    end
    $num \longleftarrow 0.0$;
    if $pap2 \neq \emptyset$ then
        for $p \in pap2$ do
            $gamma \longleftarrow 1/length(aut(p))$ /* length gives the length of a list */
            $num \longleftarrow num + exp(gamma, alpha) * ReputationPaper(p) + (1 - exp(gamma, alpha)) * 50$
        end
        $RepAuthor = num/|pap2|$;
    else
        $RepAuthor = null$
    end
    return $RepAuthor$
```

Algorithm 3: Auxiliary functions, used by Algorithms 4 and 5

```
Function v*(ri : R, rj : R, p : P):[0,100]+null=
    Data: $ri : R, rj : R$ /* researcher identifiers */
    Data: $p : P$ /* a paper identifier */
    Data: $obar : (R \times E^k)$ list /* list of vector evaluations of reviewers over $p$ */
    Data: $v : (R \times R \times E)$ list /* list of judgments over paper $p$ */
    Result: $extjudge : [0, 100] + null$ /* extended judgment of $ri$ on $rj$'s opinion of $p$ */
    /* This function computes extended judgments.  It must be called every time a new judgment is made, and
       every time a new review is added on a paper which already has reviews by others.  It is also called by
       the AverageJudgment function below and the ReputationReview function of Algorithm 5. */
    if $\exists\ e : (ri, rj, e) \in v$ then
        $extjudge \longleftarrow e$
    else
        if $\exists\ ebar, ebar' : (ri, ebar) \in obar$ and $(rj, ebar') \in obar$ then
            $extjudge \longleftarrow sim(ebar, ebar')$
        else
            $extjudge \longleftarrow null$
        end
    end
    return $extjudge$
Function sim($e : E^k, e' : E^k$):[0,100]=
    Data: $e : E^k, e' : E^k$ /* evaluation vectors */
    Result: $similar : [0, 100] + null$ /* difference */
    /* This function computes the similarity between two vectors.  It is only called by the $v^*$ function above.
       */
    $num \longleftarrow 0$;
    $num' \longleftarrow 0$;
    $den \longleftarrow 0$;
    $den' \longleftarrow 0$;
    for $i \in [1, k]$ do
        if $e[i] \neq null$ then
            $num \longleftarrow num + e[i]$;
            $den \longleftarrow den + 1$;
        end
        if $e'[i] \neq null$ then
            $num' \longleftarrow num' + e[i]$;
            $den' \longleftarrow den' + 1$;
        end
    end
    if $den \neq 0$ and $den' \neq 0$ then
        $x \longleftarrow num/den$;
        $x' \longleftarrow num'/den'$;
        $similar \longleftarrow 100 - |x - x'|$;
    end
    $similar \longleftarrow null$;
    return $similar$
Function AverageJudgment($r : R, r' : R$):[0,100]+null=
    Data: $r : R, r' : R$ /* two research identifiers */
    Result: $AvgJudge : [0, 100] + null$ /* the average judgment of $r$ over $r'$'s opinions */
    /* This function computes the average judgment of one reviewer over another.  It is only called by the
       ReputationReviwer function Of Algorithm 4. */
    $judgements \longleftarrow 0.0$;
    $num \longleftarrow 0.0$;
    for $p \in P$ do
        if $v*(r, r', p) \neq null$ then
            $judgements \longleftarrow judgements + 1$;
            $num \longleftarrow num + v*(r, r', p)$
        end
    end
    if $judgements \neq 0.0$ then
        $AvgJudge \longleftarrow num/judgements$
    else
        $AvgJudge \longleftarrow null$
    end
    return $AvgJudge$
```

Algorithm 4: Reputation of a reviewer

```
Function ReputationReviewer(r : R):[0,100]=
    Data: $r : R$ /* a researcher identifier */
    Data: $RepReviewer(r) : [0, 100]$ /* the reputation value of author $r$ */
    Result: $RepReviewer(r) : [0, 100]$ /* the new reputation value of author $r$ */
    /* This function computes the reputation of a single reviewer. It is only called by
       the function ReputationReviewers and itself, ReputationReviewer. */
    $den \longleftarrow 0.0$;
    $num \longleftarrow 0.0$;
    for $r' \in R$, $r' \neq r$ do
        if $AverageJudgment(r', r) \neq null$ then
            $den \longleftarrow den + ReputationReviewer(r')$;
            $num \longleftarrow num + ReputationReviewer(r') * AverageJudgment(r', r)$;
        end
    end
    if $den > 0.0$ then
        $RepReviewer(r) \longleftarrow num/den$;
    else
        $RepReviewer(r) \longleftarrow 50$;
    end
    return $RepReviewer(r)$;
Function ReputationReviewers : [0,100] list=
    Data: $\epsilon : [0, 100]$ /* a threshold specifying when is the difference in reputation value
       considered negligible */
    Data: $r : R$ /* a researcher identifier */
    Data: $RepReviewer(r) : [0, 100]$ /* the reputation value of an author $r$ in $R$. Initially it
       is set to $RepReviewer(r) = null$ */
    Result: $RepReviewers : [0, 100]$ list /* returns the list of updated reputation value for all
       authors $r$ in $R$; that is, $RepReviewers = \{RepReviewer(r)\}_{\forall r \in R}$ */
    /* This function computes the reputation of all reviewers. It must be called every
       time an extended judgment over an opinion of $r$ is created or modified (calculated
       by the function $v^*$ of Algorithm 3). Alternatively, this might be called once a
       day. */
    $repeat \longleftarrow true$;
    $RepReviewers \longleftarrow \{RepReviewer(r)_{\forall r \in R}\}$;
    while $repeat \neq false$ do
        $repeat \longleftarrow false$;
        for $r \in R$ do
            $RepReviewer(r)_{OLD} \longleftarrow RepReviewer(r)$;
            $RepReviewers \longleftarrow \{RepReviewers - RepReviewer(r)\} \cup ReputationReviewer(r)$;
            if $|RepReviewer(r)_{OLD} - RepReviewer(r)| > \epsilon$ then
                $repeat \longleftarrow true$;
            end
        end
    end
return RepReviewers;
```

Algorithm 5: Reputation of a review

```
Function ReputationReview($r : R, p : P, k : integer$):[0,100]=
    Data: $r : R$ /* a researcher identifier */
    Data: $p : P$ /* a paper identifier */
    Data: $k : integer$ /* minimum number of judgments to compute non-default reputation review
        value, $k > 0$ */
    Result: $RepReview : [0, 100]$ /* the reputation value of the review of $r$ over $p$ */
    /* This function computes the reputation of a particular review. It must be called
        every time an extended judgment over that opinion of $r$ is created of modified
        (calculated by the function $v^*$ of Algorithm 3), and every time the reputation of
        the author of the review is modified. */
    $jud = \emptyset$;
    for $r' \in R, r' \neq r$ do
        if $v^*(r', r, p) \neq null \wedge R_R(r') \neq null$ then
            $jud = jud \cup r'$;
        end
    end
    $den \longleftarrow 0.0$;
    $num \longleftarrow 0.0$;
    if $jud \neq \emptyset$ then
        for $r' \in jud$ do
            $den \longleftarrow den + ReputationReviewer(r')$;
            $num \longleftarrow num + ReputationReviewer(r') * v^*(r', r, p)$
        end
        $RepReview \longleftarrow num/den$;
    else
        $RepReview \longleftarrow ReputationReviewer(r)$
    end
    return $RepReview$;
```