Inverse Problems for Gradual Semantics
Nir Oren, Bruno Yun, Srdjan Vesic, Murilo Baptista

To cite this version:
Nir Oren, Bruno Yun, Srdjan Vesic, Murilo Baptista. Inverse Problems for Gradual Semantics. Thirty-First International Joint Conference on Artificial Intelligence IJCAI-22, Jul 2022, Vienna, Austria. pp.2719-2725, 10.24963/ijcai.2022/377. hal-03768129

HAL Id: hal-03768129
https://univ-artois.hal.science/hal-03768129
Submitted on 2 Sep 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
Inverse Problems for Gradual Semantics

Nir Oren\textsuperscript{1*}, Bruno Yun\textsuperscript{1}, Srdjan Vesic\textsuperscript{2}, Murilo Baptista\textsuperscript{1}

\textsuperscript{1}University of Aberdeen
\textsuperscript{2}CNRS, Univ. Artois, CRIL, France
{n.oren,b.yun,murilo.baptista}@abdn.ac.uk, vesic@cril.fr

Abstract

Gradual semantics with abstract argumentation provide each argument with a score reflecting its acceptability. Many different gradual semantics have been proposed in the literature, each following different principles and producing different argument rankings. A sub-class of such semantics, the so-called weighted semantics, takes, in addition to the graph structure, an initial set of weights over the arguments as input, with these weights affecting the resultant argument ranking. In this work, we consider the inverse problem over such weighted semantics. That is, given an argumentation framework and a desired argument ranking, we ask whether there exist initial weights such that a particular semantics produces the given ranking. The contribution of this paper are: (1) an algorithm to answer this problem, (2) a characterisation of the properties that a gradual semantics must satisfy for the algorithm to operate, and (3) an empirical evaluation of the proposed algorithm.

1 Introduction

Abstract argumentation semantics aim to identify the justification status of arguments by considering interactions between arguments. Such semantics typically operate over a directed graph, with nodes representing the (abstract) arguments, and directed edges denoting the interactions between them, e.g., attacks or supports among others. Standard argumentation semantics [Baroni et al., 2011; Dung, 1995; Caminada et al., 2012] identify sets of arguments which are considered justified (as well as unjustified and undecided). In contrast, ranking-based semantics seek to assign a ranking (or ordering) over arguments, with higher ranked arguments being considered more justified (or “less attacked”) than lower ranked arguments. Such rankings are — in most ranking-based semantics — determined by assigning numerical values (called acceptability degrees) to all arguments, with the ranking on arguments being computed based on the numerical ordering. Those ranking-based semantics are called gradual semantics. Note that not all ranking-based semantics follow this numerical approach. For instance, the ranking on arguments obtained from the burden-based or the discussion-based semantics [Amgoud and Ben-Naim, 2013], are computed using the lexicographical order on vectors of argument scores.

While some ranking-based semantics [Amgoud and Ben-Naim, 2013; Bonzom et al., 2016; Delobelle, 2017; Amgoud et al., 2016] only consider the structure of a standard Dung’s argumentation framework, others take in one or more additional features, such as a set of initial weights for each argument [da Costa Pereira et al., 2011; Amgoud et al., 2022]; weights for attacks between arguments [Coste-Marquis et al., 2012; Yun and Vesic, 2021]; a support relation [Mossakowski and Neuhaus, 2018; Mossakowski and Neuhaus, 2016; Rago et al., 2016]; or even set attacks [Yun et al., 2020]. In most gradual semantics, the final acceptability degree of an argument then depends on a range of parameters. Here, we focus on gradual semantics which take into account the structure of the graph, the initial weights of arguments, and the peculiarities of the semantics being used. Of course, the proposed approach could easily be generalised to other settings.

Rather than describing how an initial set of argument weights map to a ranking on arguments via some semantics, we instead consider the inverse problem. That is, given an abstract argumentation framework and a desired ranking on arguments, we seek to identify what initial weights should be assigned to arguments so as to obtain the desired argument ranking. We provide an algorithm for undertaking this task for a set of well-known gradual semantics which satisfy some basic properties, and evaluate the algorithm’s performance.

While we do not discuss the applications of our results, we note that potential areas in which they can be used include persuasion [Polberg and Hunter, 2018] and preference elicitation [Mahesar et al., 2018].

The remainder of this paper is structured as follows. In Section 2, we give the needed background to understand our approach. In Section 3, we describe our algorithm. Section 4 highlights the properties of the semantics over which the algorithm operates. Our empirical evaluation is detailed in Section 5, and we discuss potential applications of this work as well as avenues for future research in Section 6.

2 Background

We begin this section by providing a brief overview of abstract argumentation, as well as several gradual semantics.

*Contact Author
Following this, we introduce the bisection method, a simple technique for finding the roots of an equation which lies at the heart of our approach.

2.1 Argumentation

We situate our approach in the context of abstract argumentation. Arguments are thus atomic entities which interact with each other via an attack relationship. Such systems are encoded as directed graphs (c.f., [Dung, 1995]). Since our departure point here involves assigning each argument an initial weight, we instead consider weighted argumentation frameworks (WAFs) [Dunne et al., 2011; Amgoud et al., 2017].

Definition 1 (WAF) A weighted argumentation framework (WAF) is a triple \( \mathcal{F} = (\mathcal{A}, \mathcal{D}, w) \), where \( \mathcal{A} \) is a finite set of arguments, \( \mathcal{D} \subseteq \mathcal{A} \times \mathcal{A} \) is a binary attack relation, and \( w : \mathcal{A} \to [0,1] \) is a weighting function assigning an initial weight to each argument.

The set of attackers of an argument \( a \in \mathcal{A} \) is denoted as \( \mathcal{A} \mathcal{t}(a) = \{ b \in \mathcal{A} | (b,a) \in \mathcal{D} \} \).

A ranking-based semantics allows us to move from a WAF to a ranking over arguments. While myriad semantics have been proposed, we consider the gradual semantics described in [Amgoud et al., 2022] due to this work’s recency and the popularity of the semantics described therein. We note in advance that some of these semantics do not work with our approach, but we will use these to help explain the properties of those semantics to which our approach applies. Furthermore, while [Bonzon et al., 2016] describes 13 ranking-based semantics, only these gradual semantics allow for an initial weight to be assigned to an argument.

Definition 2 (Gradual Semantics) A gradual semantics \( \sigma \) is a function that associates to each weighted argumentation graph \( \mathcal{F} = (\mathcal{A}, \mathcal{D}, w) \), a scoring function \( \sigma^\mathcal{F} : \mathcal{A} \to [0,1] \) that provides an acceptability degree to each argument. In this paper, we consider the gradual semantics \( \sigma_x \), for \( x \in \{\mathcal{T}B, \mathcal{I}S, \mathcal{M}B, \mathcal{C}B, \mathcal{H}C\} \), defined as follows.

- **Trust-based semantics** \( \sigma^\mathcal{T}B \) [da Costa Pereira et al., 2011] are defined so that the acceptability degree of an argument \( a \in \mathcal{A} \) is \( \sigma^\mathcal{T}B(a) = \mathcal{T}B_\infty(a) \), where \( \mathcal{T}B_i(a) = 1/2 \cdot \mathcal{T}B_{i-1}(a) + 1/2 \cdot \min(w(a),1 - \max_{b \in \mathcal{A} \mathcal{t}(a)} \mathcal{T}B_{i-1}(b)) \) and for all \( b \in \mathcal{A}, \mathcal{T}B_0(b) = w(b) \).

- The iterative-schema semantics \( \sigma^\mathcal{I}S \) [Gabbar and Rodrigues, 2015] is defined such that the acceptability degree of an argument \( a \in \mathcal{A} \) is \( \sigma^\mathcal{I}S(a) = IS_\infty(a) \), where \( IS_i(a) = (1 - IS_{i-1}(a)) \cdot \min(1/2,1 - \max_{b \in \mathcal{A} \mathcal{t}(a)} IS_{i-1}(b)) + IS_{i-1}(a) \cdot \max(1/2,1 - \max_{b \in \mathcal{A} \mathcal{t}(a)} IS_{i-1}(b)) \) and for all \( b \in \mathcal{A}, IS_0(b) = w(b) \).

- The weighted max-based semantics \( \sigma^\mathcal{M}B \) [Amgoud et al., 2022] is defined such that the acceptability degree of an argument \( a \in \mathcal{A} \) is \( \sigma^\mathcal{M}B(a) = MB_\infty(a) \), where \( MB_i(a) = \frac{w(a)}{\frac{1}{\mathcal{T}B_i(a)} + \max_{b \in \mathcal{A} \mathcal{t}(a)} MB_{i-1}(b)} \) and for all \( b \in \mathcal{A}, MB_0(b) = w(b) \).

- The weighted card-based semantics \( \sigma^\mathcal{C}B \) [Amgoud et al., 2022] is defined such that the acceptability degree of an argument \( a \in \mathcal{A} \) is \( \sigma^\mathcal{C}B(a) = CB_\infty(a) \) where \( CB_i(a) = \frac{w(a)}{1 + \max_{b \in \mathcal{A} \mathcal{t}(a)} \mathcal{C}B_{i-1}(b)} \) for all \( b \in \mathcal{A}, CB_0(b) = w(b) \), and \( \mathcal{C}B_{\mathcal{I}S}(a) = \{ b \in \mathcal{A} \mathcal{t}(a) | w(b) > 0 \text{ if } \mathcal{I}S^* (a) \neq \emptyset \text{ and } w(a) \text{ otherwise.} \)

- The weighted h-categorizer semantics \( \sigma^\mathcal{H}C \) [Amgoud et al., 2022] is defined such that the acceptability degree of an argument \( a \in \mathcal{A} \) is \( \sigma^\mathcal{H}C(a) = HC_\infty(a) \) where \( HC_i(a) = \frac{w(a)}{1 + \sum_{b \in \mathcal{A} \mathcal{t}(a)} HC_{i-1}(b)} \) and for all \( b \in \mathcal{A}, HC_0(b) = w(b) \).

With the exception of \( \sigma^\mathcal{I}S \), the ranking on arguments is obtained from the acceptability degree assigned to them. For \( \sigma^\mathcal{I}S \), the semantics returns those arguments whose acceptability degree is set to 1. As usual, for every \( a, b \in \mathcal{A} \), we write \( a \succ b \) iff \( a \succeq b \) and \( b \not\succ a \), \( a \succeq b \) iff \( a \not\succ b \) or \( a \prec b \), \( a \prec b \) iff \( a \not\succ b \) but \( a \not\succeq b \), and \( a \simeq b \) iff \( a \not\succ b \) and \( b \not\prec a \).

Example 1 Let \( \mathcal{F} = (\mathcal{A}, \mathcal{D}, w) \) be a WAF, where \( \mathcal{A} = \{ a_0, a_1, a_2, a_3 \}, \mathcal{D} = \{ (a_0, a_2), (a_1, a_3), (a_1, a_2), (a_2, a_1), (a_3, a_2) \}, w(a_0) = 0.43, w(a_1) = 0.39, w(a_2) = 0.30, w(a_3) = 0.39 \). The WAF is represented in Figure 1 and the acceptability degrees for the gradual semantics of Definition 2 are shown in Table 1.

We note that the semantics described above are able to operate on cyclic graphs. Semantics such as DF-Quad [Rago et al., 2016], while popular, are designed to operate on acyclic graphs only, and we therefore ignore them in this work.

2.2 The Bisection Method

The algorithm we describe in Section 3 requires us to find the roots of a continuous function. While many techniques for doing so exist [Dekker, 1969; Brent, 2013; Verbeke and Cools, 1995], the bisection method is easily understood and numerically stable, and is therefore used in our experiments, though more advanced methods could also be used.

![Figure 1: Graphical representation of a WAF](image-url)
Algorithm 1 The bisection method.

function BISECT(f, α, β, ϵ)
    μ ← \frac{α + β}{2}
    if \(f(μ) < ϵ\) then return μ
    if \(f(μ) > 0\) then return BISECT(f, μ, β, ϵ)
    else return BISECT(f, α, μ, ϵ)
end function

Algorithm 1 details the bisection method. As input, the method takes in a function \(f\), a tolerance \(ϵ\), and upper and lower bound values \((α, β)\) respectively, such that \(f(β) < 0 < f(α)\). A single iteration of the algorithm identifies the midpoint \(μ = \frac{α + β}{2}\). If \(f(μ) > 0\), \(α\) is set to \(μ\); if \(f(μ) < 0\), \(β\) is set to \(μ\), tightening the upper and lower bounds. The process then repeats until the absolute value of the image of the midpoint is sufficiently small, i.e., \(\left|f\left(\frac{α + β}{2}\right)\right| < ϵ\). Note that one can choose to also stop when the distance between \(α\) and \(β\) is small. The number of iterations required to achieve an error \(ϵ\) is bounded by \(\log_2\left(\frac{|α - β|}{ϵ}\right)\). Note that for the bisection method to work correctly and return a unique root, the function \(f\) must be continuous and monotonic in the interval \([α, β]\).

3 The Inverse Problem

Our aim in this work is to identify a set of initial weights to obtain some desired final ranking on arguments. More formally, we take as input: (1) an unweighted argumentation framework \((A, D)\), (2) a gradual semantics \(σ\), and (3) a desired preference relation \(≻\) on \(A\). Our aim is to find a weighting function \(w\) such that in the resultant WAF \(F = (A, D, w)\), for all \((a, b) \in A, σ^F(a) ≥ σ^F(b)\) iff \(w(a) ≥ w(b)\).

In Sections 3.2 and 3.3, we describe a two phase algorithm to identify an appropriate weighting function. In phase 1, we identify an achievable acceptability degree for an argument, taking into account the desired ranking on arguments. In phase 2, we undertake a search — using the bisection method — for the initial weights necessary to achieve this desired acceptability degree. We begin however by considering several special cases of the inverse problem.

3.1 Special Cases

We begin by considering the trust-based semantics. If \(w(a) < 0.5\) for all \(a \in A\), then \(σ^TB(a) = w(a)\), making the inverse problem trivial to solve in this case. While such a solution satisfies the inverse problem, it is at odds with the intuition behind trust-based semantics as described in [da Costa Pereira et al., 2011]. In cases where, for all \(a \in A\), \(w(a) ≥ 0.5\), the presence of cycles can mean that no solution exists for the inverse problem under the \(σ^TB\) semantics. As an example of this, consider the standard 3-cycle WAF: \(\langle\{a, b, c\}, \{(a, b), (b, c), (c, a)\}, w\rangle\). If \(w(a), w(b), w(c) ≥ 0.5\), the acceptability degrees of all arguments will be 0.5.

Turning to the \(σ^{15}\) semantics, we observe that it was designed to have acceptability degrees converge to either 1, 0.5, or 0. This means that the inverse problem is not always applicable to this semantics as it can only accommodate three levels of acceptability. Moreover, there are rankings which cannot be achieved, e.g., consider the simple WAF: \(\langle\{a, b\}, \{(a, b)\}, w\rangle\), it is not possible to get \(a < b\) as \(σ^TB(a) = 1\) and \(σ^TB(b) = 0\), for any weighting \(w\).

Finally, consider a fully connected graph. We can easily prove the following proposition, which makes the solving the inverse problem on such graphs trivial.

Proposition 1 For a fully connected WAF \(F = (A, D, w)\), semantics \(σ\in\{σ^MB, σ^CB, σ^HC\}\) and any arguments \(a, b \in A\), \(σ^F(a) ≥ σ^F(b)\) iff \(w(a) ≥ w(b)\).

Given these special cases, in the remainder of the paper we consider only \(σ^MB, σ^CB\) and \(σ^HC\). We can trivially solve the inverse problem for fully connected graphs as all the semantics will converge quickly, even in the presence of a significant number of arguments and edges.

3.2 Phase 1: Computing Acceptability Degrees

We partition the set of arguments \(A\) into a sequence of non-empty sets of arguments \([Ar_0, \ldots, Ar_n]\) such that for any \(a, b \in Ar_i, 0 ≤ i ≤ n, a ≈ b\), and for any \(a \in Ar_i, b \in Ar_j\), where \(0 ≤ i < j ≤ n, a > b\). Now consider an argument \(a \in Ar_0\). For each semantics, we can reason as follows.

- \(σ^MB\): Assume that \(a\) is attacked by an argument with acceptability degree 1. If \(w(a) = 1\), its acceptability degree can be at most 0.5.
- \(σ^CB\): Assume that \(a\) is attacked by \(n\) other arguments with degree 1. Then its acceptability degree can be at most \(1/(2 + n)\). If \(a\) is the most attacked argument in \(Ar_0\), then all other arguments in \(Ar_0\) will have an acceptability degree equal to or greater than this value.
- \(σ^HC\): Assume that \(a\) is attacked by \(n\) other arguments with degree 1. Then its acceptability degree can be at most \(1/(1 + n)\). If \(a\) is the most attacked argument in \(Ar_0\), then all other arguments will have an acceptability degree equal or greater to this value.

We refer to the aforementioned values as the minimal upper bounds for the arguments in \(Ar_0\), as this is the lowest value we are guaranteed to be able to achieve (with the corresponding semantics) if the arguments in \(Ar_0\) have an initial weight of 1. Similarly, the maximal upper bound for arguments in \(Ar_0\) is 1, achievable if all attackers of arguments in \(Ar_0\) have an acceptability degree of 0. The idea is to make sure that all the arguments from \(Ar_0\) have acceptability degree in the interval \([mup, 1]\), where \(mup\) is the minimal upper bound corresponding to the semantics in question, e.g., all the arguments from \(Ar_0\) are within \([1/(1 + n), 1]\) for \(σ^HC\).

Now, consider \(Ar_1\). If the maximal upper bound of the acceptability degree of these arguments is lower than the minimal upper bound for the arguments in \(Ar_0\), then we will comply with our desired ranking on arguments. To achieve this, we set the initial weights of arguments in \(Ar_1\) to (just below) the minimal upper bound of \(Ar_0\). We can repeat this, computing initial weights, and concomitant maximal upper bounds for \(Ar_i\) by considering the minimal upper bounds of \(Ar_{i-1}\). Algorithm 2 describes this process. Note that a small constant \(ζ\) is added to the denominator in all cases to ensure that the minimal upper bound is still reduced for the special case where all arguments in some \(Ar_i\) are unattacked.
Algorithm 2 Computing arguments’ minimal upper bounds

function COMPUTEBOUNDS(\{\}, \ldots ,\ldots )
  return \{\}
end function

function COMPUTEBOUNDS([Ar0, \ldots , Arn], max, \sigma )
  switch \sigma do
    case \sigma_{MB}: \min \leftarrow \max/(1 + \max + \zeta )
    case \sigma_{HC}: \min \leftarrow \max/(1 + \max \max a_{Ar0} |Att(a)| + \zeta )
    case \sigma_{CB}: \min \leftarrow \max/(2 + \max a_{Ar0} |Att(a)| + \zeta )
  return \{(Ar0, \min )\} \cup COMPUTEBOUNDS([Ar1, \ldots , Arn], min, \sigma )
end function

function COMPUTEBOUNDS([Ar0, \ldots , Arn], \sigma )
  return COMPUTEBOUNDS([Ar0, \ldots , Arn], 1, \sigma )
end function

3.3 Phase 2: Finding the Initial Weights

Having identified appropriate minimum upper bounds for all Ar0 to Arn, we now turn our attention to finding initial weights for each argument in these sets so as to have that argument’s acceptability degree equal to the corresponding set’s minimum upper bound. By doing this, we obtain our desired ranking on arguments.

Our approach to achieving this involves picking an argument and modifying its initial weight (using the bisection method), causing it to approach its minimum upper bound value. We then pick another argument and repeat this process, until all arguments reach their desired values. There are several choices we must consider, and optimisations possible, when instantiating this approach. The most obvious choices we face revolve around selecting an argument for modification, and the decision of how much to modify the selected argument by. Myriad strategies for argument selection are possible, and in this work we consider 5 simple strategies:

S1: Select more preferred arguments for modification first. The rationale here is that such arguments have higher acceptability degrees, and fixing their values will cause fewer perturbations in the remainder of the process.

S2: Select less preferred arguments for modification first. Such arguments, with their small degree, would have little influence on the network.

S3: Select arguments further from their target degree first. By selecting arguments with the largest error first, we may perturb the network less.

S4: Select arguments nearest their target degree first. These, due to needing only minor perturbations, would have minimal effect on the rest of the argumentation system.

S5: Pick arguments at random. This is the baseline strategy.

Observe that additional strategies can be used, e.g. picking arguments with most, or fewest attackers, or which attack most or fewest arguments, first. We leave consideration of such strategies to future work.

The bisection method is only guaranteed to work for a function with a single variable, and selecting an appropriate strategy is therefore critical to our algorithm’s success. As discussed in Section 5, some of these strategies work much better than others, but we are unable to provide an analytical proof of correctness for any of the strategies.

With regards to how much we should modify a selected argument, we could do so until it is within some tolerance \epsilon of its acceptability degree, or until a certain number of iterations of the bisection method have been carried out. The rationale behind the second approach is that it allows us to respond to changes in acceptability degrees of other arguments due to our modifications more rapidly than if we modify only a single argument at a time.

If \beta is the desired acceptability degree for an argument a, then we can use the bisection method to find a new initial weight \wma for a such that |\sigma^2(a) - \beta| \leq \epsilon where \sigma^2(a) is the acceptability degree of a in the WAF where the weight of a is now \wma. To apply the bisection method we need to also identify an initial upper and lower bound. While we can use the values 1 and 0 for this, we can also identify tighter bounds, leading to a small improvement in performance. First, consider the lower bound \alpha passed to the bisection method. Since our denominator is at least 1, we can set \alpha to the minimal upper bound. For \beta, assume we wish to achieve a minimal upper bound of m(a) for argument a, which has n attackers. Now consider \sigma_{MB}, and assume that the strongest attacker has acceptability degree 1. We have that m(a) = w(a)/(2 + \zeta) and so can set \beta to min\{(2 + \zeta) \cdot m(a), 1\}. Using this idea, for \sigma_{HC}, we can set \beta to min\{m(a) \cdot (1 + n + \zeta), 1\}, and for \sigma_{CB} to min\{(2 + n + \zeta) \cdot m(a), 1\}.

From a practical point of view, observe that the target acceptability degree computed in Phase 1 may be very small. The stopping condition of our bisection method should therefore use a relative error \{|(\alpha + \beta)/2 - m(a)|/m(a) < \epsilon rather than an absolute error. Since we evaluate acceptability degrees as part of the bisection method, we can also terminate our algorithm early if the acceptability degrees returned in the evaluation match our desired ranking, even if they have not yet converged to the desired minimum upper bound.

4 Properties

We now examine properties of our approach and the underlying semantics, identifying necessary conditions over the latter which are needed for the former to work. Given the iterative nature of the underlying semantics, proving that some of these properties hold is difficult, and in Section 5, we carry out an empirical evaluation which strongly suggests that the \sigma_{MB}, \sigma_{HC} and \sigma_{CB} semantics respect these properties. Properties which we are able to demonstrate are identified as propositions, with associated proofs in the supplementary material, while those we are unable to analytically demonstrate are labelled as conjectures.

The first property we consider involves the weights obtained in Phase 1 (Section 3.2). We need to demonstrate that the computed weights are achievable. While we can easily demonstrate that the computed weight is achievable in isolation, doing so for the entire system is more difficult.
Conjecture 1 (Weighting Validity) For any unweighted argumentation graph \( \langle A, D \rangle \) and \( \sigma \in \{ \sigma_{MB}, \sigma_{HC}, \sigma_{CB} \} \), there is a weighting function \( w \) such that for all \( 0 \leq i \leq n \), for all \( a \in A_r \), \( \sigma^F(a) \) is equal to its minimum upper bound (as computed by Algorithm 2), where \( F = \langle A, D, w \rangle \).

In Phase 2, for the bisection method to operate, we must demonstrate that our semantics is continuous (otherwise we may be unable to converge to a solution); and that these changes are (strongly) monotonic (as otherwise we may have any number of solutions). Thus, our solution satisfies uniqueness (though uniqueness does not imply monotonicity).

**Property 1 (Uniqueness) Given two WAFs \( F = \langle A, D, w \rangle \), \( F' = \langle A, D, w' \rangle \) and some \( a \in A \) such that \( w(a) \neq w'(a) \) and for all \( b \neq a \in A \), \( w(b) = w'(b) \). It holds that \( \sigma^F(a) \neq \sigma^{F'}(a) \), for \( \sigma \in \{ \sigma_{MB}, \sigma_{CB}, \sigma_{HC} \} \).

**Property 2 (Continuity) A gradual semantics \( \sigma \) satisfies continuity iff for any WAF \( F = \langle \{a_1, a_2, \ldots, a_n\}, D, w \rangle \), \( X^F = (\sigma^F(a_1), \sigma^F(a_2), \ldots, \sigma^F(a_n)) \), we can find \( F' = \langle \{a_1, a_2, \ldots, a_n\}, D, w' \rangle \) (with unbounded weights) such that there is at least one \( a \in A \) s.t. \( w(a) \neq w'(a) \) and \( |X^{F'} - X^F| < \delta \) for any positive \( \delta \). We say that \( \sigma \) satisfies bounded continuity iff it satisfies continuity and the initial weights \( F \) are restricted to \([0, 1]\).

**Property 3 (Strong Monotonicity) A gradual semantics \( \sigma \) satisfies strong monotonicity iff for any two WAFs \( F = \langle A, D, w \rangle \), \( F' = \langle A, D, w' \rangle \) for which there is some \( a \in A \) such that \( w(a) = w'(a) + \delta, \delta > 0 \) and for all \( b \in A \setminus \{a\} \), \( w(b) = w'(b) \), it holds that \( \sigma^F(a) > \sigma^{F'}(a) \).

This in turn yields the following proposition.

**Proposition 2** A gradual semantics \( \sigma \) which does not satisfy strong monotonicity or bounded continuity could have multiple, or no solutions to the inverse problem. In other words, both are necessary conditions for a unique solution for the inverse problem to exist.

We conjecture that \( \sigma_{MB}, \sigma_{HC} \) and \( \sigma_{CB} \) meet these conditions. We are unable to demonstrate strong monotonicity and bounded continuity though we show uniqueness and continuity for them in our supplementary material. The empirical evaluation suggests our approach operates successfully.

**Proposition 3** The gradual semantics \( \sigma \) satisfies continuity and uniqueness, for \( \sigma \in \{ \sigma_{MB}, \sigma_{HC}, \sigma_{CB} \} \).

Conjecture 2 The gradual semantics \( \sigma \) satisfies strong monotonicity and bounded continuity for \( \sigma \in \{ \sigma_{MB}, \sigma_{HC}, \sigma_{CB} \} \).

### 5 Evaluation

We evaluated each of the strategies discussed in Section 3.3 over directed scale-free, small world (Erdos-Renyi), and complete graphs of different sizes (number of arguments).

As part of our evaluation, we ran 10, 100 and 2000 iterations of the bisection method for each argument before using a strategy to pick the next desired argument. Table 2 describes the remaining parameters used in our evaluation.

We created a simple target preference ordering for our experiments, randomly placing each argument within the graph into one of 5 levels of preference. This meant that in all cases, at least some arguments had equal desired preference levels.

Our experiments evaluated the runtime of the different strategies, the number of times the bisection method was invoked, and the number of times the total iterations required exceeded the permitted maximum number of iterations. Given the number of dimensions across which our evaluation took place, we present only a subset of our results here; full results can be found in the supplementary material.

Our main criterion for evaluation revolves around the number of times the bisection method was called by our approach. As shown in Figure 2, which is representative of the results for most graph topologies, our runtime grows in a super-linear manner, due — as shown in [Amgoud et al., 2022] — to the increased time taken to evaluate a semantics on larger graphs. The number of bisection method iterations is shown in Figure 2 for our different semantics and graph types when selecting the next argument based on largest relative error. This value grows linearly (with a gradient between 1 and 2 depending on topology and semantics) for all semantics considered \((R^2 > 0.99\) for all cases). Arguments are thus only recomputed at most twice before our approach converges. These results not only demonstrate the feasibility of our approach for large argumentation graphs, but also highlight the effectiveness of this specific argument selection strategy. We also note that there is little variance in our results for \( \tilde{CB} \). We believe that this is due to the extra \(|Att|\) term in the semantics; this term overweights the term which depends on other arguments’ final degrees, making the result more dependent on the topology of the graph than in the case of the other semantics.

Due to space, we omit several detailed results. In summary, (a) from all argument selection strategies, only selecting arguments from largest to smallest relative error resulted in always finding a solution to the inverse problem; (b) allowing for partial convergence (via fewer iterations per argument before moving to the next one) decreased performance, often failing to find a solution; (c) Optimising \( \alpha \) and \( \beta \) bounds (cf. Section 3.3) had almost no influence on runtime. This unsurprising due to the speed of bisection method convergence.

Note that in the absence of equivalent arguments with the same acceptability degree, one could allow for early termination without getting to the target acceptability degrees, but that was not investigated in the current paper.

<table>
<thead>
<tr>
<th>( \zeta )</th>
</tr>
</thead>
<tbody>
<tr>
<td>10, 20, ..., 150</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Graph Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Erdos-Renyi probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1, 0.3, 0.5, 0.7</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Maximum relative error</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.001</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Bisection method iterations</th>
</tr>
</thead>
<tbody>
<tr>
<td>10, 100, 2000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Bisection method ( \epsilon )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.001</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Maximum bisection method calls</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000</td>
</tr>
</tbody>
</table>

Table 2: Parameters used in our evaluation

---

1Source code for our algorithm and evaluation can be found on GitHub at https://github.com/jhudy/numerical_inverse.
6 Conclusion & Discussion

We considered the inverse problem for a WAF. We demonstrated that a solution to this inverse problem exists for at least one family of semantics, and that a solution does not exist for at least some semantics. We then described an algorithm to solve this problem, and empirically evaluated its performance. Our results show the viability of our approach. When selecting arguments for initial weight perturbation based on relative error, each argument is typically perturbed at most twice (depending on semantics and graph topology).

Our approach was able to find weightings over all evaluated graphs and semantics, suggesting that $HC, CB$ and $MB$ meet all the requirements described in Section 4. Some of our results rely on empirical analysis, and providing analytical proofs for these is a critical avenue of future work.

We note in passing that our approach assumes that our argument ranking is total. Extending our scheme to admit partial rankings is — in a sense — trivial, but would result in a combinatorial explosion as we would need to consider multiple possible argument rankings as inputs.

To our knowledge no work has explicitly considered the inverse problem applied to gradual semantics as described in this paper, but several works have examined related concepts under different guises. The work of [Dunne et al., 2015] is perhaps closest to ours. They consider the case where one is given a set of extensions and a semantics, and need to decide whether there exists an argumentation framework that induces the given set of extensions. Unlike our work, they situate their approach in standard abstract argumentation semantics. Work on the epistemic approach to probabilistic argumentation [Hunter and Thimm, 2017] describes properties which probabilistic argumentation semantics should satisfy. Selecting some of these properties constrains the probabilities which arguments can have. Similarly, in fuzzy argumentation [Wu et al., 2016] calculates legal ranges of fuzzy degrees for arguments based on initial weights and the semantics underpinning the fuzzy argumentation system. Finally, argumentation dynamics examines what arguments or attacks should be introduced to strengthen or weaken an argument, somewhat analogous to our changing of initial argument weight.

There has been some work on sensitivity analysis within argumentation [Tang et al., 2016]. This work considers whether (small) changes in argument weights will affect the conclusions that can be drawn from an argumentation framework. The results reported on in this paper are a first step towards our long-term goal to provide a formal analysis of sensitivity to initial weights in $MB, CB$ and $HC$ style semantics.

The conditions specified in Prop. 2 are necessary for our algorithm to operate. As mentioned above, we are still investigating whether these conditions are also sufficient, or whether additional properties need to be identified. Once this is done, we will be able to categorise other weighted semantics unrelated to those discussed in the current work (e.g., the constellation-based probabilistic semantics [Li et al., 2011]) and consider whether our approach can be applied to them.

While we believe analytical solutions exist for the inverse problem for some semantics, one advantage of the numerical approach proposed in this work is that it is more generally applicable to a wide range of semantics. The current work therefore describes an easily applied approach to solving the inverse problem in argumentation.

Figure 2: Runtime (in seconds) and number of iterations for the different semantics and graph types
References


