Computational Models of Argument
F. Toni et al. (Eds.)

doi:10.3233/FAIA220134

Well, to Be Honest, I Wouldn't Start from Here at All

(A Personal View of Complexity in Argumentation After 20 Years)

Paul E. Dunne
Department of Computer Science, University of Liverpool, UK

Abstract. Computational complexity theory and the related area of efficient algorithms have formed significant subfields of Abstract Argumentation going back over 20 years. There have been major contributions and an increased understanding of the computational issues that influence and beset effective implementation of argument methods. My aim, in this article, is to attempt to take stock of the standing of work in complexity theory as it presently is within the field of Computational Argument, as well as offering some personal views on its future direction.

Introduction

There is an English tourist on a walking holiday in Eire who, while wandering around, becomes aware that he has no idea of what direction he ought to take. He sees a farmer and, approaching him, then says "I say, old chap, I'm in a bit of a frightful mess here. I'm trying to find The Old Manor House, and I don't have the deuce of a notion how to get there. Be most awfully grateful if you could help." The farmer looks up, saying in reply "So it be the Old Manor House ye're wantin' then? Sure an' that's nott a problem at all. What ye want to be doin' is this: ye go across thon field, ye turn left at the hayrick, left mind ye now, nott right, ye carry on till ye reach the cowshed and then . . . No, no no, that's not it, that's not it at all. Here what ye need to do is go down to the brook, follow it about a hunner yards an' ye'll come to a bridge, ye cross that an' straight on another hunner yards or so till ye see the windmill, ye go to the right (right mind ye right nott left) and, and . . ." The farmer's voice trails off and he hesitates a long while before continuing. "Sure and it's a divvil of a problem this, divvil of a problem. Ye know what I'm thinking? Ye know what it is I'm thinking? I'm thinking if I wanted to get to The Old Manor House, *well, to be honest, I wouldn't start from here at all*.".

All of which is to make the point that sometimes it feels as if Computational Complexity Theory (or, more accurately, the practice of Complexity Theory within Computational Argument) is in a similar position to that of the English tourist: vaguely aware of an end it wishes to achieve but unsure of how best to get there in the most direct way, and, in consequence, finding its path diverted along the scenic detours offered by brooks and windmills. My purpose, in this article, is to consider the extent to which this viewpoint is justified. In doing so, after the short recap of Section 1, I consider, in Section 2, the origins of Computational Argument in a form that gave a model suited to algorithmic and complexity study: this, of course, is the watershed approach of Dung [27]. I will look at what grew from Dung's work over the ten years between its appearance and the inaugural COMMA in 2006. This forms the basis of Section 3. What would prove to be a discovery of crucial importance to complexity analysis of Dung's model appeared in a different context: what is now dubbed *The Standard Translation* of Dimopoulos and Torres [16]. The basis of this and its importance are considered in Section 4. As complexity and algorithmic study of Dung's model proceeded the concept of *Canonical Decision Problem* emerged as a means of focusing issues with new models. In Section 5 I revisit this canon examining what its impact on Computational Argument has been in Section 6. In Section 7 the central theme is that of areas of neglect: which and to what extent such lacunae matter. Conclusions are offered in Section 8.

This may seem to be, as is probably apparent from its opening, a rather unusual paper: there are no intricate technical analyses of existing models, no new models being presented and justified. Its primary stance is that of a personal reflection on the status of a specialist field in which I have spent twenty years researching. As such its opinions about significant landmarks are highly subjective and should not be regarded as definitive factual assertions. It is also rather exceptional in containing no occurence of the word "*divers*", as in "*divers models*".

1. Prelude: Algorithms and Complexity

As a topic of research the study of algorithms has a history going back over 2,000 years: Euclid presents methods for geometric constructions in *The Elements*; Eratosthenes a technique for identifying Prime Numbers; Newton and his successors would offer approaches to finding so-called zeros of functions, generalizing the discoveries of Cardano and Ferrari respecting closed form solutions for roots of small degree polynomials [12]. Euclid, however, not only presents solutions but poses problems: one of these, "squaring the circle" would remain unresolved until the close of the 19th Century (Lindemann [37,38]). I mention this background to stress not only the historical depth of algorithm study but also to highlight some consequences already beginning to be apparent as a consequence of Lindemann's discoveries. Lindemann demonstrated that an algorithmic problem could not be solved *within the system allowed for its solution* (Ruler-and-compass). Fifty years later the discoveries of Gödel and Turing [31,44], would show that the underlying system was only part of the cause: the phenomenon of not being solvable by algorithmic means was pervasive and exhibited by all programming systems. One result is to split the world of function computation into two parts: some computational process (i.e. algorithm) exists and no such process is possible. Study of the latter, in the guise of Recursive Function Theory, would give rise to many ideas (e.g. degrees of computability, The Arithmetic Hierarchy) already beginning to lose any tangible link to computational concerns as faced in reality. In as much as Computational Argument is linked to proof theory within classical logic such non-computability issues are present in argumentation.

The concern of non-computability while present is, however, not where the focus of algorithm study has been regarding Computational Argument. The objects we wish to identify *can* be discovered: the question arising from the investigations of algorithms and Computational Complexity theory is whether it is possible to do so "efficiently". The notion of what is meant by efficiently is that the resources used by the algorithm, be it some measure of run time or of memory demands, increase proportionately to some slowly growing function of the input size. Run time uses polynomial growth rate (formally n^k for some constant k on inputs with n items). Algorithms address specific computational problems. Computational Complexity asks questions concerning what are the best algorithms possible for a given problem. Here we find a common misusage: algorithms have resource demands, they do *not* have complexity; problems have an associated complexity (formally a problem belongs to a complexity *class* membership of which is witnessed by an algorithm whose requirements are captured through that class).

There are problems for which efficient algorithms have been discovered and there are those for which no efficient algorithm is possible. Since its formal development in the mid 1960s with Hartmanis and Stearns [32][1] Computational Complexity theory has faced an open issue of some considerable magnitude: we know for any resource demand that there are functions whose computation must use at least this bound, however, there are no typically encountered problems for which such bounds have been *formally proved*. This leaves many of the problems often met in practical settings in an uncertain state: it is believed, albeit on the basis of purely circumstantial indicators, that usable algorithmic solutions cannot be found but there is no definite proof of this. These circumstantial indicators are based on the argument that an efficient solution for one yields efficient methods for all. By far the most common measure of intractability arises from the technical theory of NP–completeness, see Garey and Johnson [30]: a defence of the claim “The problem P is intractable” being made by demonstrating that P is NP–complete. Such demonstrations involve taking a known NP–complete problem, Q say, and phrasing algorithms for it in terms of algorithms for P: if the rephrasing or *reduction* is efficient (written $Q \leq_p P$) then any fast algorithm for P perforce yields a fast algorithm for Q. Computational Complexity theory offers NP–completeness as one basis of intractability, there are, however, many others, cf. Johnson [33].

In summary a formal proof that some problem is NP–complete is viewed as sufficient to take that problem out of the realm of those for which efficient algorithmic methods are possible. This, of course, should not lead to giving up on solution methods or becoming resigned to inordinate performance demands. A demonstration of NP–completeness is just the start: having accomplished such, attention turns to a whole arsenal of mechanisms which have been proposed to force the over-demanding within tractable limits. Some of these approaches are reviewed within Section 7.

Before any of these researches – algorithm efficiency and problem complexity – can have a foundation established in argumentation a common basis for their investigation is essential. This is found not in propositional logic nor predicate calculus, not in some vague notion of natural language interpretation and reasoning: it is found in the classical structure of directed graphs. It is the landmark in the study of Computational Argument: the Abstract Argumentation Frameworks of Dung [27].

[1]There is an argument for Shannon [43], but the basic model is very different from that of classical complexity theory.

2. In the beginning

In directed graphs we have *nodes* and a relationship between pairs of nodes which is not required to be symmetric: the *links* or edges. The model of abstract argument proposed by Dung [27] views nodes as arguments and a link from argument p to argument q as expressing the fact "argument p *attacks* argument q". Its key assumption is that the arguments, within this structure, are atomic and indivisible. It does not attempt to rationalize *why* argument p attacks argument q but simply states it as an aspect of whatever scenario is being modelled. This separation of *what* is being described from *how* it is described avoids concerns with soundness and rationality: the inner level intricacies underlying the actual structure of a stated position are hidden. One *can*, of course, drill down into the formal structure of an atomic argument within Dung's formalism but in doing so the result will not be to treat a single node differently but to substitute a new collection of atomic arguments presenting the detail of what is being replaced. It is, however, in the reduction of argument relationships to *solely* the concept of attack that Dung's model achieves much of its power. If an argument attacks another then an immediate consequence is that the two cannot simultaneously be accepted by a rational agent: there is an inherent conflict in belief. But we can go further, if a chain of three arguments is such that x attacks y which attacks z then, in principle, both x and z may be rationally accepted since should an adversary dispute the validity of z by proposing y then y can in turn be disputed by advancing x. It is this interplay of attack and defence (although this term is not used by Dung) that positions Dung's abstraction as a powerful dialectical modelling system.

Look again at the two very basic examples of attack structure presented in the preceding paragraph: "x attacks y" and "x attacks y attacks z". In the former it is reasoned that at most one of the pair can be held; in the latter that because the attack on z has been countered the position described by the set $\{x, z\}$ is tenable. In these are found the other significant contribution of abstract argumentation frameworks. The interpretation of a collection of mutually endorsed positions as a subset of graph nodes satisfying given criteria: what has become known as abstract argument *semantics*.

In Dung's paper a succession of these is built up from the most basic (*conflict-free*) through more refined concepts (*admissible*, *complete*) culminating in quite sophisticated ideas (*preferred*, *stable*, *grounded*). As I recap in the next section these basic six would soon be added to in conjunction with the practice of developing new models that continues to be a feature of Computational Argument in the present day.

3. Schism: semantics and models

Dung's paper [27] appeared in 1995: introducing the basic graph model, methods for interpretation (i.e. argument semantics), properties of these semantics, more advanced notions (coherence, controversial arguments, infinite structures). By the time the first Computational Models of Argument conference was held in 2006 [18] not to mention the seminal special issue of the leading Artificial Intelligence research journal [7], Dung's basic model and half dozen semantics had burgeoned into such as semi-stable semantics (Caminada [11]), bipolarity (Cayrol and Lagasquie-Schiex [13]), preference-based argument (Amgoud and Cayrol [1]), symmetric frameworks (Coste-Marquis *et al.* [17]), Value-based argument frameworks (Bench-Capon [6]). Later yet would see Extended Argument Frameworks (Modgil [40]), half a dozen forms of weighted and probabilistic schemes, and, already in 2006 we had hypergraph structures offered as an alternative to the simple directed graph form (Nielsen and Parsons [42]).

Now it is really a matter of no importance *what* these objects are and how they are defined: all that is relevant is that they *are at all*. For what does the existence of a dozen different variant semantics and models say about Dung's approach?

One might claim that this proliferation indicates and attempts to correct some "inadequacy" or failure in Dung's model, but I would consider this view to over simplify. Take the case of semi-stable semantics. On the surface this deals with a problem with Dung's stable semantics: there are systems for which no stable semantics exists whereas every framework has defined semi-stable solutions and, furthermore, these coincide with stable sets should such be present. One can debate the extent to which the semi-stable approach affords a solution to this non-existence issue (I have, however, no intention of doing so). The point is simply that without exception all of the proposed new models and new semantics serve a purpose in that these have been anchored within some *perceived* problem arising in Dung's model, and irrespective of how such problems are addressed typically the basic formalism remains graph-theoretic and the accompanying semantics set-theoretic. While cases such as Modgil's Extended Argumentation Framework [40], let alone the recursive attack and infinite frameworks of Baroni *et al.* [3,4,5] stretch the notion of "directed graph" and "subset semantics" considerably these act in a manner which I would say remains recognizable.

In as much as there may be issues with, to use Simari's evocative phrase, "a plethora of semantics" (and models)[2] I think it is more a pedagogical concern than an inherent structural weakness in Dung's model: the effort of sifting and distinguishing the vast panoply of different suggestions and ideas presents steep demands for neophytes to the world of Computational Argument. This is potentially confusing and debatably unfortunate: what it is *not*, however, is an indication that its base is fundamentally flawed. In fact, when we look at Computational Complexity in Argument, as I shall now turn to, we find a remarkable unity in how these different methods can all be analysed. That such unity is possible is, in no small degree, due to the astonishing versatility of a quite fundamental mechanism: the Standard Translation of Dimopoulos and Torres [16].

4. Breakthrough: The Standard Translation

When Cook in [15] presented the first[3] NP–complete problem he adopted the problem of propositional satisfiability for this end: given an arbitrary propositional formula $\varphi(X)$ over a variable set X determine if there is a setting of X that will make φ **true**. In essence Cook showed that the behaviour of any reasonable computational system, such as a Turing machine, could be mimicked by asking about the satisfiability of an efficiently constructed propositional formula. Cook was able to refine the construction so that it continued to be valid for a special class of formulae: those presented in *Conjunctive Nor-*

[2]The distinguished contributor to work on Computational Argument, Guillermo Simari, coined the phrase "*plethora of semantics*" in his talk at the inaugural COMMA [39] describing the state that had already been reached.

[3]*pace* the claims of Levin [36].

mal Form (CNF). In what is now referred to as *The Standard Translation* a CNF-formula is changed into a Dung-style argumentation framework in which a named argument is accepted under some set-theoretic semantics if and only if the source CNF is satisfiable.

I will not repeat the details of this translation here: the interested reader may find these described in a number of general survey works. e.g. [25] and also within relevant specialist articles. It suffices to observe that the Standard Translation describes any CNF formula as a tripartite structure: an argument representing the formulae itself; a second set describing the clauses[4]; a final set presenting the individual literals ($\{x_i, \neg x_i : 1 \leq i \leq n\}$) used in the formula. Its nature is such that the formula argument will belong to a set satisfying the criteria of a given semantics if and only if a selection of the literal arguments can be made such that setting these to **true** provides a satisfying assignment of the original formula. From this elegant and basic construction the first demonstrations of intractable behaviour in abstract argumentation frameworks are obtained: credulous acceptance under Dung's admissibility semantics and the existence of a stable system of arguments are both shown to be NP–complete. The device also suggests some algorithmic directions. For just as a CNF-formula can be transmuted into an argument framework so too some properties of an argument framework can be encapsulated within propositional and thence CNF structures, e.g. Egly and Woltran [29].

With the sole exception of preference-based schemes the view provided by the Standard Translation has been adopted to demonstrate intractability results in all of the variant forms I mentioned earlier (value-based, weighted, extended); with respect to novel semantics (semi-stable, resolution-based, cf2, ideal, stage, naive) and with respect to different graph-theoretic restrictions (planar, acyclic, binary tree). It finds new applications within the study of argument semantics signatures and the concept of realizability (Dunne *et al.* [23]; Dvořák *et al.* [26]). It is not limited to demonstrations of intractability at the level of mere NP-completeness being powered up to Π_2 and Σ_2 completeness results in the cases of sceptical acceptance in Dung's preferred semantics (Dunne and Bench-Capon [21]), and, as demonstrated by Dvořák and Woltran [28], with respect to semi-stable questions. It links basic argumentation dialogue games to a very primitive (but sound and complete) proof calculus (Vreeswijk and Prakken [46]), opening up another avenue for complexity-theoretic studies (Dunne and Bench-Capon [22]).

5. The concept of Canonical Problem

In comparing one approach and semantics against an alternative technique – especially in the context of complexity and algorithmic study – we need to have some benchmark collection of ideas for such comparison. The formulation of 4 standard problems[5] provides this.

Hence given any proposed semantics, σ and graph-based model $M(\mathcal{X}, \mathcal{A})$ we have:

A. Existence (does any non-empty collection of items within M satisfy the criteria set by σ?)

[4] A clause being a disjunction of literals.

[5] The list presented in [25, Table 1.1] separates the existence problem into two: one allowing empty sets to satisfy criteria and another (called non-emptiness in [25]) phrased identically to that of our Existence problem. I have chosen to eschew considering the former problem as canonical.

B. Credulous Acceptance (is there some collection satisfying the conditions of σ and containing this argument, p?)
C. Sceptical Acceptance (does every collection satisfying σ have p as a member?)
D. Verification (does this collection meet the conditions prescribed by σ?)

From such a basis is obtained the standard agenda for complexity-theoretic analysis of new semantics and models: for each of the canonical problems determine exact bounds on its complexity. Notice this involves both algorithmic construction (witnessing that a problem *can* be solved within a particular resource bound) and demonstration that such algorithms cannot be significantly improved (categorization within a specific complexity class). Having addressed the basic questions attention often turns to variations within the model itself, e.g. graph-theoretic restrictions such as planarity, k-colourability, acyclicity.

6. What have Algorithmic Study and Complexity given to Argument?

Some of Dung's semantics can be dealt with efficiently for all of the canonical problems: for example the grounded semantics. Others, notably the preferred, not known to have efficient methods for any and, in fact, have strong indicators that no such methods are possible. Other mix the trivial (sceptical acceptance in the conflict free semantics) with straightforward efficient methods (verification in the naive semantics).

Such rough division may seem already to provide a case for advocating one semantics in preference to another. This, however, if applied naively does not give a good basis. One might put forward (at least) one reason as to why an efficiently solvable semantics (for example the grounded) is not necessarily to be favoured over one less so (for example the preferred semantics). Often it is found that the more tractable cases are only so on account of their limited expressive ability: hence the grounded semantics offers useful outcomes only within the sub-class of argument frameworks defined by directed graphs having at least one source node. Conflicting viewpoints, however, tend to produce models in which every argument attacks and is attacked by another: here the grounded semantics offers no information. Thus in total there is a balance between expressibility in modelling terms and algorithmic tractability with respect to the canonical problems.

While complexity-theoretic analysis does not realistically inform the *choice* of a semantics, it does provide an *awareness* of potential issues: the knowledge that the verification problem for preferred semantics is coNP–complete may be insufficient to reject it, however, it may, having identified some admissible set, discourage attempts to push that set to maximality.

The significant contribution that complexity theory has made to the study of argument is, I would say, in this notion of awareness. Irrespective of the semantic formalism, irrespective of the graph-theoretic model each of the four canonical problems raises a single question: is it possible to solve this problem efficiently? Of course I would not claim the related questions had been ignored prior to the first complexity studies of abstract argumentation, but I would claim that an awareness of the underyling issues is heightened by considering matters in a uniform style: that of the four canonical questions with respect to semantic criteria.

With respect to algorithmic contributions, it is appropriate at this point to raise one method which, in principle, offers an alternative semantic treatment: the labelling approach championed, most notably by Verheij [45] and Caminada (see, e.g. Modgil and

Caminada [41]). The basic labelling semantics uses three labels called IN, OUT, UNDEC with different semantics captured by the configurations allowed within legal labellings. Such approaches offer a solution to questions of semantics (via those arguments legally assigned the label IN) and algorithms (in processes for allowing a final labelling to evolve from an initial default labelling). In [41, Section 7, 127–8], Modgil and Caminada offer a cogent summary of the gains achieved through labelling semantics and, of interest from the perspective of the current topic, algorithms. Labelling is not, of course, some latter day Philosopher's Stone, turning the base metal of intractability to desired efficient solutions. It does, however, as noted by Modgil and Caminada offer useful insight into dialectical processes thereby linking esoteric algorithmic matter to how argument may be recognized in the "real world".

The algorithmic contribution to argument therefore consists not only of the tangible gains in computational efficiency but also in the rather more subtle effect of providing insight into the mechanism of argument itself.

In total, computational complexity theory provides evidence (albeit circumstantial) that looking for efficient under all conditions algorithms for a number of argumentation problems will be fruitless. Formal algorithm study has contributed efficient methods, and for many special cases, useful practical methods. Very often these stray into quite advanced algorithmic ideas, e.g. fixed-parameter tractability and its specific case of Courcelle's Theorem. Nonetheless I would claim that there are many avenues, well studied in classical algorithm theory as an angle on intractability, the investigation of which in argumentation has been barely touched. What are these? That is the question considered in the next section.

7. Omissions and Neglect

I remarked earlier that in the classical study of algorithms and complexity a demonstration that a problem is likely to be unsolvable does not signal the end of further investigation. Instead a whole range of possible means of coping come into play: randomized and probabilistic methods; approximation techniques, special cases, average case efficiency, fixed parameter tractable representations, backdoor techniques. Now it is certainly true that a number of these have been considered in computational argument. For example special case study dates back at least as far as the work of Coste-Marquis *et al.* [17] on symmetric frameworks; bipartite forms are shown to be tractable in Dunne [19]. Similarly a reasonable volume of work has accrued in the study of fixed parameter and backdoor methods.

Despite these, there are areas which have been at best neglected, at worst overlooked entirely. This may partly be a matter of fashion (for example the once very thriving area of so-called phase-transition phenomena, e.g.[14], where there have been only superficial studies). Phase-transition phenomena, in much as there is algorithmic potential, have always seemed to me, personally, to have elements of "smoke-and-mirrors", some of these aspects being discussed in Dunne, Gibbons and Zito [24]. So possibly it is not surprising that a full scale study of threshold phenomena in argumentation has yet to be undertaken: here is an approach becoming a historical curiosity, that promises much (efficient algorithmic solutions for generally intractable problems) but in reality actually delivers little of practical use (that is to say, there *are* efficiently *on average* fast methods but these are for cases on the verge and not for the classes of instance that will be seen in real contexts). So the famous "conjuring trick"[6] of Angluin and Valiant [2] is dependent on a combinatorial property of random graph structures: as with all good conjuring tricks the effect at first surprising loses interest once seen how it is achieved.

If phase transition phenomena are ideas whose time has passed the rather more focussed and related concern of average-case study is a very different matter. This, again, is an approach for which only superficial studies (if that) have been undertaken. Average-case studies work from a base in which input instances are chosen at random according to some probability distribution. The forms being studied in argumentation would thus be treated as random directed graph structures. In order to avoid the combinatorial legerdemain underpinning the performance in [2] (typical random directed graphs contain many links and this specific method performs well on such graphs), reliable evidence for usable average case argumentation algorithms needs to focus on graphs typical of those seen in reality. I would claim that the following problems have yet to be fully considered:

T1. Develop a model of random argumentation frameworks that reflects the characteristics of typical frameworks.
T2. Develop methods for generating random representatives within this model.
T3. Extend these to value-based (VAF) and abstract dialectical frameworks (ADFs).

Average-case investigation is well established as a field within algorithmics and has attracted a modicum of interest in argumentation. There is, however, another well studied (in graph theoretic terms) approach whose relevance to argumentation has received minimal attention. I refer here to the study of spectral properties. Directed graphs may be described as $(0, 1)$-matrices and the analysis of the eigenvalues and eigenvectors of $(0, 1)$-matrices has historically offered some insight into graph-theoretic problems, e.g. node colouring, Wilf [47]. One significant use of spectral analysis in other areas has been its application to ranking problems, most notably in the mechanics underpinning Google's search algorithm, see [9] but also other ranking environments, e.g. Keener [34], Kleinberg [35]. Such approaches and the importance of ranking as an issue in argumentation, e.g. Bonzon *et al.* [8] leave, to my mind, the comparative neglect of spectral analysis rather puzzling. A very basic and preliminary investigation is reported in [10], however its findings are very inconclusive. The following questions are, I think, worthy of more detailed study:

S1. Google's ordering approach involves identifying an eigenvector of a dominant eigenvalue within a rational valued matrix defined from webpage linkages. The linkage structure is a directed graph. In principle treating an argument framework in such a manner might give some insight into argument "importance". There is, however, a complication: in simple terms Google's page significance function is cumulative (if page X links to page Y which links to page Z, the score assigned Z will have positive contributions from the scores of X and Y). A naive translation to argument runs into an immediate problem: a higher score for X should result in a lower score for Y and thence a higher score for Z. The side-effect of non-monotonicity raises the question of formulating scoring functions for argument (akin to Google's technique) that might allow analysis of argument ranking as the ordering of components in an eigenvector.

[6] This description is a little bit unfair, however not one that I suspect the authors would dispute.

S2. The study in Butterworth and Dunne [10] looks at potential relationships between the eigenvalues associated with an argument and its acceptability under different semantics. A full comparative study of this has yet to be carried out.

I choose these simply as significant broad areas which, in my view, have yet to be the subject of sustained and systematic study. There are other specialist techniques found helpful in algorithm yet untried in argument. It may, also, be the case that such detailed studies will yield nothing of interest. We will not know this, however, unless we try.

8. Conclusion: "are we there yet?"

The trite and obvious answer to this question is, of course, not by some margin. What, however, do I intend precisely by "there"? A reasonable view would be to equate this with the general agenda of algorithmic and computational complexity studies. That is to say the classification of problem difficulty (computational complexity theory) combined with concerted attacks aimed at exorcising the worst side-effects of intractable behaviour. I think there can be no question that with respect to the first of these there has been notable success: one struggles to think of any significant argumentation problem (certainly with respect to the classical semantics of Dung [27]) whose complexity status remains open. Similar comprehensive achievements have been delivered with respect to new semantics (semi-stable, resolution-based, and, with one niggling gap, ideal) and also within alternative models (value-based, extended argumentation frameworks).

Against these contributions, my feeling is that too little attention has been given to the nature of efficient algorithmic methods. This is understandable, the analytic acrobatics brought to bear in engineering some intractability proofs (I trust the reader will excuse my citation of [19, Thm. 12], let alone [20, Corollary 5]) can be hard to resist. This, however, should not detract from the fact that the decision problems addressed arise from a real application setting, and thus effective solution, in at least as much as such can be developed, is a necessity. If there is one urgency I would identify from the body of work produced so far it is that of addressing algorithmic approaches.

References

[1] L. Amgoud and C. Cayrol. A reasoning model based on the production of acceptable arguments. *Annals of Mathematics and Artificial Intelligence*, 34(1):197–215, 2002.
[2] D. Angluin, L.G. Valiant. Fast probabilistic algorithms for hamiltonian circuits and matchings. *Journal of Computer and System Sciences*, 18(2):155–193, 1979.
[3] P. Baroni, F. Cerutti, P.E. Dunne, and M. Giacomin. Computing with infinite argumentation frameworks: the case of AFRAs. In *Proc. 1st Intnl. Workshop on Theory and Appl. of Formal Argumentation (TAFA)*, pages 197–214, 2011.
[4] P. Baroni, F. Cerutti, M. Giacomin, and G. Guida. AFRA: argumentation framework with recursive attacks. *Int. J. Approx. Reason.*, 51(1):19–37, 2011.
[5] P. Baroni, F. Cerutti, P. E. Dunne, and M. Giacomin. Automata for infinite argumentation structures. *Artificial Intelligence*, 203:104-150, October 2013
[6] T.J.M. Bench-Capon. Persuasion in practical argument using value-based argumentation frameworks. *Jnl. Logic and Computation*, 13(3):429–448, 2003.
[7] T. J. M. Bench-Capon and P. E. Dunne. Argumentation in artificial intelligence. *Artificial Intelligence*, 171(10-15):619–641, 2007.

[8] E. Bonzon, J. Delobelle, S. Konieczny, and N. Maudet. A comparative study of ranking-based semantics for abstract argumentation. In *Proc. of the 30th AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA.*, pages 914–920, 2016.

[9] K. Bryan and T. Leise. The $25,000,000,000 eigenvector: The linear algebra behind Google. *Siam Review*, 48(3):569–581, 2006.

[10] J. Butterworth and P. E. Dunne. Spectral techniques in argumentation framework analysis. In *Proc. 6th COMMA*, volume 287 of *FAIA*, pages 167–178. IOS Press, 2016.

[11] M. Caminada. Semi-stable semantics. In *Proceedings of COMMA 2006*, vol. 144 Frontiers in Artificial Intelligence and Applications, IOS Press, 2006, pages 121–128.

[12] G. Cardano. *Artis Magnae, Sive de Regulis Algebraicis*. 1545.

[13] C. Cayrol and M. C. Lagasquie-Schiex. On the acceptability of arguments in bipolar argumentation frameworks. In *European Conference on Symbolic and Quantitative Approaches to Reasoning and Uncertainty (ECSQARU)*, pages 378–389, 2005.

[14] P. Cheeseman, B. Kanefsky, and W. Taylor. Where the really hard problems are. *Proc. of the 12th IJCAI*, pages 331–337, 1991.

[15] S. A. Cook. The complexity of theorem-proving procedures. In *Proc. of the 3rd Annual ACM Symposium on Theory of Computing*, pages 151–158. ACM, 1971

[16] Y. Dimopoulos and A. Torres. Graph theoretical structures in logic programs and default theories. *Theor. Comput. Sci.*, 170(1-2):209–244, 1996.

[17] S. Coste-Marquis, C. Devred, and P. Marquis. Symmetric argumentation frameworks. In L. Godo, editor, *Proc.* $8^{\rm th}$ *European Conf. on Symbolic and Quantitative Approaches to Reasoning With Uncertainty (ECSQARU)*, volume 3571 of *LNAI*, pages 317–328. Springer-Verlag, 2005.

[18] P. E. Dunne and T. J. M. Bench-Capon (editors) *Computational Models of Argument: Proceedings of COMMA 2006*, vol. 144 Frontiers in Artificial Intelligence and Applications, IOS Press, 2006.

[19] P. E. Dunne. Computational properties of argument systems satisfying graph-theoretic constraints. *Artificial Intelligence*, 171:701–729, 2007.

[20] P. E. Dunne. The computational complexity of ideal semantics, *Artificial Intelligence*, 173(18):1559–1591, 2009.

[21] P. E. Dunne and T. J. M. Bench-Capon. Coherence in finite argument systems. *Artificial Intelligence*, 141(1/2):187–203, 2002.

[22] P. E. Dunne and T. J. M. Bench-Capon. Two party immediate response disputes: properties and efficiency. *Artificial Intelligence*, 149:221–250, 2003.

[23] P. E. Dunne, W. Dvořák, T. Linsbichler, and S. Woltran. Characteristics of Multiple Viewpoints in Abstract Argumentation. *Artificial Intelligence*, 228:153-178, November 2015

[24] P. E. Dunne, A. Gibbons, and M. Zito. Complexity-theoretic models of phase transitions in search problems. *Theoretical Computer Science*, 249(2):243–263, 2000,

[25] P. E. Dunne and M. Wooldridge. Complexity of abstract argumentation. In I. Rahwan and G. R. Simari, editors, *Argumentation in Artificial Intelligence*, Springer, Berlin, pages 85–104, 2009.

[26] W. Dvořák, P. E. Dunne and S. Woltran. Parametric Properties of Ideal Semantics. *Artificial Intelligence*, 202:1-28, September 2013

[27] P. M. Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial Intelligence*, 77:321–357, 1995.

[28] W. Dvořák and S. Woltran. Complexity of semi-stable and stage semantics in argumentation frameworks. *Inf. Process. Lett.*, 110(11):425–430, 2010.

[29] U. Egly and S. Woltran Reasoning in Argumentation Frameworks using Quantified Boolean Formulas In *Proceedings of COMMA 2006*, vol. 144 Frontiers in Artificial Intelligence and Applications, IOS Press, 2006, pages 12 133–144.

[30] M. R. Garey and D. S. Johnson. *Computers and Intractability*. W. H. Freeman New York, 2002.

[31] K. Gödel. Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme, I. *Monatshefte für Mathematik und Physik*, v. 38 n. 1, 173–198, 1931

[32] J. Hartmanis and R. E. Stearns, On the Computational Complexity of Algorithms. *Transactions of the American Mathematical Society*, 117:285–306, 1965.

[33] D. S. Johnson, A Catalog of Complexity Classes, Algorithms and Complexity (editor: J. van Leeuwen), Handbook of Theoretical Computer Science, Elsevier, 67–161, 1990.

[34] J. P. Keener. The Perron-Frobenius theorem and the ranking of football teams. *SIAM Review*, 35(1):80–93, 1993

[35] J. M. Kleinberg. Authoritative sources in a hyperlinked environment. *JACM*, 46(5):604–632, 1999
[36] L. A. Levin. Universal sequential search problems. *Problemy Peredachi Informatsii*, 9(3):115–116, 1973.
[37] F. Lindemann. Über die Ludolph'sche Zahl. *Sitzungsberichte der Königlich Preussischen Akademie der Wissenschaften zu Berlin*, 2:679–82, 1882.
[38] F. Lindemann. Über die Zahl π. *Math. Ann.*, 20:213–225, 1882.
[39] D. C. Martínez, A. J. García and G. R. Simari. On acceptability in abstract argumentation frameworks with an extended defeat relation. In *Proceedings of COMMA 2006*, vol. 144 Frontiers in Artificial Intelligence and Applications, IOS Press, 2006, pages 273–278.
[40] S. Modgil. Reasoning about preferences in argumentation frameworks. *Artificial Intelligence*, 173(9–10):901–934, 2009.
[41] S. Modgil and M. Caminada. Proof theories and algorithms for abstract argumentation frameworks. In I. Rahwan and G. R. Simari, editors, *Argumentation in Artificial Intelligence*, pages 105–129. Springer, 2009.
[42] S. H. Nielsen and S. Parsons. Computing preferred extensions for argumentation systems with sets of attacking arguments. In *Proceedings of COMMA 2006*, vol. 144 Frontiers in Artificial Intelligence and Applications, IOS Press, 2006, pages 97–108.
[43] C. E. Shannon. The synthesis of two-terminal switching circuits. *Bell System Technical Journal*, 28(1):59–98, 1949.
[44] A. M. Turing. On computable numbers, with an application to the Entscheidungsproblem. *Proceedings of the London Mathematical Society*, 2(1):230–265, 1937.
[45] B. Verheij. A labeling approach to the computation of credulous acceptance in argumentation. In *Proc. 20th IJCAI*, pages 623–628, 2007.
[46] G. Vreeswijk and H. Prakken. Credulous and sceptical argument games for preferred semantics. In *Proceedings of JELIA'2000, The 7th European Workshop on Logic for Artificial Intelligence.*, pages 224–238, Berlin, 2000. Springer LNAI 1919, Springer Verlag.
[47] H. S. Wilf. The eigenvalues of a graph and its chromatic number *Journal of the London Mathematical Society*, s1-42 (1):330–332, 1967