Efficient Argumentation for Medical Decision-Making

Robert Craven and **Francesca Toni** and **Cristian Cadar** and **Adrian Hadad**
Department of Computing, Imperial College London,
South Kensington Campus, London SW7 2AZ, UK

Matthew Williams
University College Hospital,
235 Euston Road, London NW1 2BU, UK

Abstract

We describe the application of assumption-based argumentation (ABA) to a domain of medical knowledge derived from clinical trials of drugs for breast cancer. We adapt an algorithm for calculating the admissible semantics for ABA frameworks to take account of preferences and describe a prototype implementation which uses variant-based parallel computation to improve the efficiency of query answering.

1 Introduction

Some forms of decision-making (e.g., medical diagnosis and treatment) require decision-makers to consider different types of information (e.g., clinical trials, medical guidelines), often across various domains (e.g., branches of medicine), while taking into account requirements that cannot always be quantified (e.g., quality of life, survival benefit, patient preferences). This plethora of information and requirements may give rise to conflicts (e.g., two different clinical trials may be in contradiction, or medical guidelines may impose constraints inconsistent with patient preferences). In addition, the decisions need to be justified, in order to give assurance that those making them have considered and weighed all relevant factors.

Computational argumentation has been suggested for supporting this form of decision-making in medicine (e.g., in (Hunter and Williams 2010)). It computes *arguments* from knowledge represented by logical sentences, *attack* relationships between arguments, and the strength of arguments proportionally to their capability to defend themselves against attacks. In a medical context, logical sentences from which arguments and attacks are built can be extracted from medical sources, such as clinical trials. However, the number of sources is typically very large, and building argumentative justifications from them is a serious computational challenge. For example, more than 500 papers reporting the results of clinical trails of drugs on patients with breast cancer alone are published each year. Moreover, there are large numbers of choice points in the search for arguments, including many alternatives for the construction of arguments, different ways to defend against attacks and different ways to conduct the debates leading to the construction of the derivations. The use of preferences, say for one clinical trial over another on the basis that the latter trial is less relevant, or less well-conducted, than the former (Hunter and Williams 2010), can be used to reduce the search space, but poses computational challenges for obtaining justifications that take preferences into account.

The many choice-points in the decision-making we aim to support, combined with the large volume of information it needs to take into account, makes the automation of this decision-making a demanding task. We believe that none of the existing systems for computational argumentation can scale to the many choice points and the large amount of data that must be considered. In order for computational argumentation to deliver its promise and have a tangible impact and substantial practical benefits on medical decision-making, an efficient and scalable system is needed that can deal with large amounts of information and preferences, and that can return justifications in suitable formats. In order to realise this goal, in this paper we use parallel programming techniques to implement a variant of the AB-dispute derivations (Dung, Kowalski, and Toni 2006; Gaertner and Toni 2008) for Assumption-Based Argumentation (ABA), supporting the interleaved computation of arguments and attacks (as in standard AB-dispute derivations (Dung, Kowalski, and Toni 2006)), the computation of the dialectical trees combining these arguments and attacks (as in (Gaertner and Toni 2008)) as well as the computation of the full underlying argument trees. The latter extension is important to allow doctors to view in full the evidence underlying recommended decisions. The new form of AB-dispute derivations is tailored, via a specific selection function, to ABA frameworks compiling reasoning about preferences, in the manner of (Toni 2008).

We choose to use variant-based parallel programming techniques because we believe that single-threaded, centralised systems, performing each of the choices underlying debates sequentially (which include all existing argumentation systems) are inadequate as (1) the algorithmic complexity of different sequences of choices can vary dramatically—and certain sequences of choices may not even terminate in a practical amount of time; (2) this complexity cannot be determined in advance, and thus no intelligent heuristics can be adopted at design time. Instead, by executing different choices in parallel and choosing the most promising one af-

ter a given amount of time, problems which cannot be solved by single-threaded systems in a practical amount of time can be addressed (Cho 1997; Bordeaux, Hamadi, and Samulowitz 2009; Trachsel and Gross 2010). At the same time, the use of variant-based parallelisation can lead to the beneficial exploitation of new hardware platforms (Cadar, Pietzuch, and Wolf 2010)—ranging from multicore processors to large-scale data centres—which provide an abundance of computational resources and can support a high degree of parallelism effectively.

2 Background

2.1 Assumption-based Argumentation

An ABA framework (Dung, Kowalski, and Toni 2009) is a tuple $\langle \mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{\ } \rangle$ where

- $\langle \mathcal{L}, \mathcal{R} \rangle$ is a deductive system, with $\mathcal{L}$ the *language* and $\mathcal{R}$ a set of *rules* $\delta_0 \leftarrow \delta_1, \ldots, \delta_m (m \geq 0)$ with each $\delta_i \in \mathcal{L}$;
- $\mathcal{A} \subseteq \mathcal{L}$ is a (non-empty) set, known as the *assumptions*;
- $\overline{\ }$ is a total mapping from $\mathcal{A}$ into $\mathcal{L}$ where $\overline{\delta}$ is the *contrary* of δ.

Given a rule $\delta_0 \leftarrow \delta_1, \ldots, \delta_m$, δ_0 is referred to as the *head* and $\delta_1, \ldots, \delta_m$ as the *body*. An ABA framework is *flat* if and only if no assumption is the head of a rule.

Informally, following (Dung, Kowalski, and Toni 2009):

- *an argument for (the claim)* $\delta \in \mathcal{L}$ *supported by* $A \subseteq \mathcal{A}$ ($A \vdash \delta$, in short) is a (finite) tree with nodes labelled by sentences in $\mathcal{L}$ or by τ[1], root labelled by δ, leaves either τ or assumptions in A, and non-leaves δ' with as children the elements of the body of some rule with head δ';
- *an argument* $A_1 \vdash \delta_1$ attacks an argument $A_2 \vdash \delta_2$ if and only if δ_1 is the contrary of one of the assumptions in A_2.

With argument and attack defined for a given $\langle \mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{\ } \rangle$, standard argumentation semantics can be applied in ABA, e.g.:

- a set of arguments is *admissible* if and only if it does not attack itself and it attacks all arguments that attack it.

For an ABA framework and a claim, a *structured AB-dispute derivation* (Gaertner and Toni 2008) is a way of computing a set of admissible arguments, with one of the arguments having the claim in question. Derivations are modelled as a series of steps taken either by a *proponent*, building arguments in favour of the claim, or an *opponent*, building arguments that attack those of the proponent. Such derivations gradually construct arguments, creating new attacking arguments from the other player when an argument is found to depend on an assumption. (The details are in (Gaertner and Toni 2008).)

ABA does not directly deal with preferences, but some kinds can be accommodated by translating them into ABA. In particular, (Toni 2008) shows how certain sets of defeasible and strict rules with additional defeasible and strict rules defining preferences over them (called *epistemic frameworks*), can be translated into ABA frameworks. We make use of that work in the following, where preferences over rules will represent the different weights given to clinical trials of drugs in medicine, but again pass over the details.

[1] $\tau \notin \mathcal{L}$ stands for "true"/the empty body of rules.

2.2 Variant-Based Parallel Computation

Parallelisation can be relatively straightforward if a problem is decomposable into entirely independent parts, which only need to be recombined after each has been solved (known as "problem splitting"). Unfortunately, many problems cannot be so easily decomposed. There are several broad approaches to the parallelisation of problems when simple problem splitting is infeasible, but in the current work we apply variant-based competitive parallel execution (Cho 1997; Bordeaux, Hamadi, and Samulowitz 2009; Cledat et al. 2009; Vajda and Stenstrom 2010; Trachsel and Gross 2010).

Multiple versions of a possible sequential process are created, which will all give the same eventual result, but take different times to do so. Variants are started in parallel, and when the 'winner'—the first to find a solution to the problem—announces its triumph, the other variants are killed. The number of variants run by a system can vary with the number of available resources (e.g., CPU cores), which is useful in a flexible computation environment, where resources are dynamically added or removed. Communication and synchronisation are very simple with this technique—variants operate independently, and as soon as one of them completes execution, the others are killed by the system.

3 Modified AB-dispute Derivations

In the present paper, we make two modifications to the definition of an AB-derivation, to make it more suited to supporting medical treatment recommendation. First, we alter the way in which arguments are represented. In the original formulation (Gaertner and Toni 2008), arguments are recorded as structures $[S_m, S_u] \vdash \sigma$ where S_m is known as the *marked support* and S_u the *unmarked support*. The marked support are assumptions required to support the argument's conclusion σ. The unmarked support contains assumptions and other sentences that may be relevant to supporting the conclusion σ. When an argument represented in this form is *complete* (the unmarked support is empty), the chain of reasoning that led back, via intermediate sentences and any number of rules, is lost. For arguments presented to clinicians, this is clearly inadequate: in order to evaluate the arguments an argumentation-based support system provides, and satisfy the requirement of transparency, the full chain of reasoning should be shown.

We replace the existing representation of arguments by triples $(\mathcal{T}, S_m, S_u)$, where $\mathcal{T}$ is a partly-completed argument tree for the argument's conclusion, and S_m and S_u are *sets of nodes* of the tree containing the argument's marked and unmarked support.

Secondly, the nature of the medical domain (see section 4.1) requires that we take account of preferences over rules in our derivations. The medical knowledge is represented as defeasible rules, with several rules obtained as the results of a given clinical trial. Some clinical trials are weighted more strongly than others by clinicians, and these

preferences are inherited by the rules derived from the trials; they should be taken into account by the argumentation.

When making an AB-derivation for an ABA framework which has been derived from a defeasible epistemic framework according to the transformation in (Toni 2008), the way in which rules are selected to be used in proofs should be sensitive to the preference information. We now give priority to the proof or disproof of information relating to preferences, so that if a rule is used which, because of its low placing in the preferential ordering, should be disregarded, then this is found as quickly as possible. In effect, we constrain the selection function used in AB-dispute derivations to choose those members of S_u which relate to establishing what the preferential ordering over rules is.

A full technical specification of the algorithms for the derivations is omitted owing to space constraints.

4 Evaluation

We implemented our approach in `sxdd`, a system targeting multicore CPUs, written in C++. In this section, we describe our experience applying `sxdd` in a medical setting.

4.1 Medical Data

The medical data we used describes the results of clinical trials of treatments for early-stage breast cancer. It was initially collected and analysed as part of one author's PhD thesis, with experiments being reported in (Williams and Hunter 2007). Fifty-seven papers referred to in the National Cancer Institute's breast cancer guidelines (NCI 2007) were examined. The papers report randomised trials of drugs for breast cancer, meta-analyses, and clinical guidelines. An OWL ontology of 190 classes was designed to represent the medical domain.

We transformed this ontology into the Prolog-style syntax accepted by our argumentation tool, resulting in 947 ground rules. The results of the clinical trials, reported in the papers, can be represented in the form of rules such as:

1. From (EBCTCG 2005): *Women who have early ER-positive breast cancer and are given a 2-year course of tamoxifen have, in general, an increased disease-free survival rate of 1.21%.*
2. From (Rutqvist et al. 1995): *Women with early breast cancer given a course of Tamoxifen have an increased risk of endometrial cancer of 4.1%.*

Increased risks of cancer or other diseases and decreases in disease-free survival rate would argue against a given treatment; decreased risks of disease and increased survival rates would argue for a given treatment. The two rules above can accordingly be transformed into the following defeasible rules:

$$\mathsf{Woman}(W) \wedge \mathsf{hasDisease}(W, D) \wedge \mathsf{BreastCancer}(D) \quad (1)$$
$$\wedge\, \mathsf{ERPositive}(D) \wedge \mathsf{TamoxifenTwoYear}(T)$$
$$\Rightarrow \mathsf{haveTreatment}(W, T)$$

$$\mathsf{Woman}(W) \wedge \mathsf{hasDisease}(W, D) \wedge \mathsf{EarlyStageBC}(D) \quad (2)$$
$$\wedge\, \mathsf{Tamoxifen}(T) \Rightarrow \neg\mathsf{haveTreatment}(W, T)$$

The weight such rules carry in an argument for or against a given treatment depends on how trusted the clinical trial from which the knowledge was extracted is. Many different criteria are typically applied to studies in order to evaluate the quality of evidence they provide, including such measures as the number of patients the study involved and the length of time the patients' histories were followed after treatment ended. Other measures depend on information about the patient for whom treatment is being decided: if the patient is post-menopausal, for instance, then a trial directed on pre-menopausal women would, *ceteris paribus*, be valued less highly than one on post-menopausal women.

For the current paper, we have included preferences between different papers reporting clinical trials, but have abstracted away from the details of how such preferences are generated. Each rule such as (1) or (2) above was given a unique ID, and a preference ordering over the IDs was randomly generated. The rules were then ground, with the preferences applying to the non-ground IDs being inherited by the ground versions. Where there was a preference for a rule over a related rule with contradictory head, this preference was itself represented as a rule: the result is an *epistemic framework*, in the sense of Section 2.1. We generated a corresponding ABA framework, and the results were added to the (ground) rules from the medical ontology. The resulting ABA framework had 2345 rules, with a language ($\mathcal{L}$) size of 2469 sentences, of which 408 were assumptions.

We included example patient data for a hypothetical patient 'MsJones', who was stipulated to be over 50, post-menopausal, and to have proto-ER-positive, LN-positive, Stage 2 breast cancer. We then asked our system to find arguments for a sample query, namely whether a daily course of Tamoxifen (used in the treatment of early-stage breast cancer) for two years was justified for MsJones. Arguments were successfully generated respecting the preferential structure, meaning that rebuttals to counterarguments were only successfully employed when the clinical trial from which the rebuttal claim was obtained was ranked more highly in the preference ordering that for the counterargument. The following section discusses our results.

4.2 Experimental Results

For parallelisation, the important parts of the algorithm for making AB-derivations are the *choice points*. These choice points do not affect the overall outcome of the derivation (although they may lead to widely different execution times), so they can be effectively used in variant-based parallelisation.[2] The parameters involved in these choice points, together with their possible values, are as follows:

- **turn choice**: either (i) the proponent develops an argument, if possible; or (ii) the opponent does, if possible;
- **proponent's argument choice** and **opponent's argument choice**: either (i) the newest argument added is

[2] In contrast, other decisions, such as which rule in the ABA framework to use next in developing an argument, do affect the final outcome of a derivation and one needs to backtrack over them.

chosen; (ii) the oldest argument is chosen; (iii) the argument with the smallest set of non-processed leaf nodes is chosen; or (iv) the argument with the largest set of non-processed leaf nodes is chosen;

- **proponent's node choice** and **opponent's node choice**: either (i) choose a non-assumption node from the argument to develop, if possible; or (ii) choose an assumption node from the argument, if possible.

The parametrisation on strategies involving these choice points means there are a total of $(2 \times 4 \times 4 \times 2 \times 2) = 128$ possible variants. Figure 1 shows, for each variant, the time

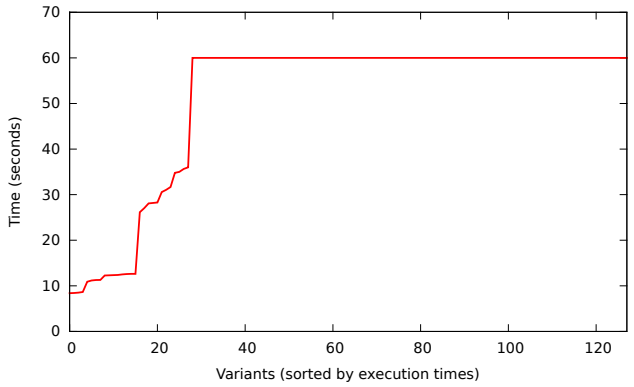

Figure 1: CPU time for breast cancer data on sample query, with cut-off time of 60 seconds.

taken by `sxdd` to answer the query described in Section 4.1. The minimum CPU time taken by a variant was 8.39 seconds, and 28 variants out of 128 completed before the chosen cut-off time of 60 seconds. (In order to check that the cut-off time was not unnecessarily strict, we ran 10 randomly selected variants, out of those that took longer than 60 seconds, for 10 minutes each; none of them answered the query in under 10 minutes.)

We first defined a 'successful' variant as one which completed in under 20 seconds; this seems a reasonable responsiveness given that the system is envisaged as being used in the context of a patient consultation, where several queries may need to be posed and answers obtained quickly. There were 16 variants that completed in less that 20 seconds.

To assess the probabilities of picking at least one successful strategy, for different numbers of variants, we used the hypergeometric distribution (e.g., see (Johnson, Kemp, and Kotz 2005)). To achieve a 90% probability of picking a successful strategy, at least 17 variants are needed; for 80% probability, the number of variants needed falls to 12, and for 70% probability, 9 variants are needed.

Note that the distribution of the execution times for different variants cannot be predicted in advance from the query and argumentation framework used for constructing arguments, and running variants sequentially, one after another, may prove impractical for many data sets and queries, including the case considered here. However, by spawning a large number of variants to run in parallel, we can significantly increase the likelihood of choosing at least one variant that completes within the necessary cut-off time.

We believe the results obtained so far suggest that variant-based parallelisation can offer significant benefits in achieving acceptable speeds of query-answering in an argumentation framework in the medical domain. However, these are only preliminary results, and more experiments are needed to better quantify the benefits of this approach on different data sets and different queries.

In addition to running these sequential tests, we also successfully deployed our system on a 16-core machine, and we are currently using this platform to conduct more extensive experiments.

5 Related Work

Our proposed derivations are an extension of the AB-dispute derivations of (Dung, Kowalski, and Toni 2006), by (i) rendering explicit the computation of the dialectical tree of arguments and counter-arguments and of the argument trees, implicit in the original AB-dispute derivations, and (ii) rendering explicit different computational choices that affect the search for AB-dispute derivations (e.g., the turn-making choice, or the selection of which argument to expand). The need for/benefits of rendering computational choices such as the turn-making explicit has been also advocated in (Gaertner and Toni 2007), with software engineering motivations. Here, we have shown how rendering these choices explicit can support useful experimentation.

Several tools exist in support of argumentation-based reasoning. Most focus on abstract argumentation, but some proposals exist for tools for concrete argumentation frameworks. For example, Gorgias (Demetriou and Kakas 2003), for credulous argumentation in logic-programming-based argumentation frameworks with preferences amongst defeasible rules; the ASPIC system (Fox et al. 2007) dealing with quantitative uncertainty; DeLP (Garcia and Simari 2004) for defeasible logic programming.

Several authors have suggested the use of argumentation for decision making and decision support in medicine, e.g., (Fox et al. 2010; Williams and Hunter 2007). Our work builds on these earlier approaches.

Variant-based parallelisation has been effectively used in the past to improve application performance (Cho 1997; Bordeaux, Hamadi, and Samulowitz 2009; Cledat et al. 2009; Vajda and Stenstrom 2010; Trachsel and Gross 2010).. For example, (Cho 1997) introduces the idea of using competitive execution to speed up distributed programs. More recently, (Bordeaux, Hamadi, and Samulowitz 2009) optimise SAT solving by automatically generating multiple variants ("portfolios") of a given solver, which are then run competitively in parallel. (Trachsel and Gross 2010) propose a general framework for competitive execution that targets multicore and multiprocessor systems, in which sequential applications are optimised by introducing competitive variants for parts of the program.

6 Conclusion

We have presented a novel argumentation system, implementing a new form of dispute derivations in ABA and making use of variant-based parallelisation techniques. We have applied our system in a concrete medical domain, where it has shown promising results. The implementation we have developed supports, in addition to AB-dispute derivations, also the GB-dispute derivations of (Dung, Mancarella, and Toni 2007). Further work is needed to extend the implementation to an adaptation of the IB-dispute derivations of (Dung, Mancarella, and Toni 2007), but this is straightforward. More work is also needed to study the formal properties of our extensions of dispute derivations. Since these derivations extend the standard AB-dispute derivations conservatively, they can easily be proven to compute the same sets of assumptions as standard AB-dispute derivations, giving soundness as a direct consequence. In order to prove that the computed dialectical structures are 'correct' further work is required, possibly along the lines of (Toni 2011).

For our medical experiments, the medical ontology required by the clinical trials modelled was added by hand. Translation into a suitable ABA representation was semi-automated. Further work is required to automate these steps fully, as far as possible.

Our initial experiments with variant-based parallelisation have shown the technique to be well-suited to argumentation, where the impact of different choice points is difficult to determine in advance. However, further experimental analysis is needed to understand the distribution of variant execution times in different application domains, and whether some choice points perform consistently better on certain data sets.

Most of the experimentation we have done so far has been to test whether a specific treatment is recommended or not. There is a strong need, however, for the system to be able to reason efficiently about which of a number of treatments is recommended; this will be the first step in making the work more clinically relevant.

References

Bordeaux, L.; Hamadi, Y.; and Samulowitz, H. 2009. Experiments with massively parallel constraint solving. In *Proc. IJCAI'09*, 443–448. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Cadar, C.; Pietzuch, P.; and Wolf, A. L. 2010. Multiplicity computing: a vision of software engineering for next-generation computing platform applications. In *Proc. FOSER'10*, 81–86.

Cho, S. H. 1997. Competitive execution: a method to exploit idle workstations. In *Proc. ICPADS'97*, 382 –391.

Cledat, R.; Kumar, T.; Sreeram, J.; and Pande, S. 2009. Opportunistic computing: A new paradigm for scalable realism on many-cores. In *Proc. HotPar'09*. Berkeley, California: USENIX Association.

Demetriou, N., and Kakas, A. C. 2003. Argumentation with abduction. In *Proc. the Panhellenic Symp. on Logic*.

Dung, P. M.; Kowalski, R. A.; and Toni, F. 2006. Dialectic proof procedures for assumption-based, admissible argumentation. *Artificial Intelligence* 170(2):114–159.

Dung, P. M.; Kowalski, R.; and Toni, F. 2009. Assumption-Based Argumentation. In *Argumentation in Artificial Intelligence*. Springer. 199–218.

Dung, P. M.; Mancarella, P.; and Toni, F. 2007. Computing ideal sceptical argumentation. *Artificial Intelligence* 171(10–15):642–674.

EBCTCG. 2005. Effects of chemotherapy and hormonal therapy for early breast cancer on recurrence and 15-year survival: an overview of the randomised trials. *The Lancet* 365(9472):1687–1717.

Fox, J.; Glasspool, D.; Grecu, D.; Modgil, S.; South, M.; and Patkar, V. 2007. Argumentation-based inference and decision making–a medical perspective. *Intelligent Systems, IEEE* 22(6):34–41.

Fox, J.; Glasspool, D.; Patkar, V.; Austin, M.; Black, L.; South, M.; Robertson, D.; and Vincent, C. 2010. Delivering clinical dec. support services: There is nothing as practical as a good theory. *J. of Biom. Inf.* 43(5).

Gaertner, D., and Toni, F. 2007. Computing arguments and attacks in assumption-based argumentation. *IEEE Intelligent Systems* 22(6):24–33.

Gaertner, D., and Toni, F. 2008. Hybrid argumentation and its properties. In *Proc. COMMA'08*.

Garcia, A., and Simari, G. 2004. Defeasible logic programming: An argumentative approach. *Journal of Theory and Practice of Logic Programming* 4(1-2):95–138.

Hunter, A., and Williams, M. 2010. Using clinical preferences in argumentation about evidence from clinical trials. In *Proc. IHI'10*, 118–127.

Johnson, N. L.; Kemp, A. W.; and Kotz, S. 2005. *Univariate discrete distributions*. Wiley-Blackwell.

NCI. 2007. Breast Cancer PDQ (Stage I, II, IIA, and operable IIIC Breast Cancer).

Rutqvist, L. E.; Johansson, H.; Signomklao, T.; Johansson, U.; Fornander, T.; and Wilking, N. 1995. Adjuvant tamoxifen therapy for early stage breast cancer and second primary malignancies. *J. of the Natl. Cancer Inst.* 87(9):645–651.

Toni, F. 2008. Assumption-based argumentation for epistemic and practical reasoning. In *Computable Models of the Law, Languages, Dialogues, Games, Ontologies*, volume 4884 of *LNCS*. Springer. 185–202.

Toni, F. 2011. A unified framework for dispute derivations in assumption-based argumentation. Technical Report.

Trachsel, O., and Gross, T. R. 2010. Variant-based competitive parallel execution of sequential programs. In *Proc. CF'10*, 197–206.

Vajda, A., and Stenstrom, P. 2010. Semantic information based speculative parallel execution. In *Proc. PESPMA'10*.

Williams, M., and Hunter, A. 2007. Harnessing Ontologies for Argument-Based Decision-Making in Breast Cancer. In *ICTAI (2)*, 254–261. IEEE Computer Society.