On the performance of argument-based dialogue systems

Murphy, Josh

Awarding institution:
King's College London

On the Performance of Argument-Based Dialogue Systems

By
Josh Murphy

A thesis submitted in partial fulfilment for the
degree of Doctor of Philosophy

in the
Department of Informatics
School of Natural & Mathematical Sciences
King's College London

2019

Abstract

Argumentation is an approach to reasoning that can be implemented in machines because of its logical foundations, and which is easily understood by humans because of its dialectical nature. By enabling humans to reason with machines through dialogues, argumentation accommodates even non-experts to scrutinise and communicate with computational agents. Such technology is valuable at a time in which the need for accountability in intelligent machines is ever increasing, and as human-machine interaction becomes ever more commonplace.

However, if practical argument-based applications are to be realised, the technologies and systems that underpin them should be effective and efficient. Furthermore, applications should be optimised to run on the kinds of structures of argumentation which exist in the domain that they operate in.

In this thesis, therefore, we seek to understand the performance of computational argument-based dialogue systems. We begin by investigating the effect of domain on the performance of these systems, and then develop two approaches to strategic reasoning in argument-based dialogues that are computationally efficient and effective.

Acknowledgements

I express my deepest thanks to my supervisors, Elizabeth Black and Michael Luck. Liz's expertise, generosity, and encouragement have been central to my PhD. I greatly appreciate her endless help and encouragement, and for setting an excellent example to which I aspire to in my own work. Mike has always pushed me to do my best (and to do it faster!), while at the same time being understanding of my circumstances. I feel honoured to have had such outstanding mentors.

My thanks go to all the academics, professional services staff, friends, students, and colleagues in the Department of Informatics at King's College London, as well as the wider university community. The department has been my place of study throughout my higher education, and I have always felt happy there thanks to the people in it. Particular thanks go to: Peter McBurney, Simon Miles, Sanjay Modgil, Odinaldo Rodrigues, Peter Young, Steffen Zschaler, Samhar Mahmoud, Lina Barakat, Alex Burdusel, and many others whose time at King's I have had the pleasure of sharing. Thanks must also go to David Kohan Marzagão, whose enthusiasm, coffee, and punctuality have respectively inspired, sustained, and delayed me throughout my PhD studies.

Thank you to Federico Cerutti (Cardiff University) and Nir Oren (University of Aberdeen), for kindly agreeing to take the time to examine this thesis, and for their detailed comments on the work. Thanks also to the computational argumentation community, for the many stimulating ideas they have provided me with through reviews, presentations, and discussions.

Thank you to my parents for their support over the years, and for always giving me the freedom to follow my own goals. Without their encouragement I would not be where I am today.

My final acknowledgement is reserved for my wife, Rachel. I could not have completed this work without your love and care. I thank you for your patience, and for putting some of our aspirations on pause so that I could achieve one of mine. I owe you a good few holidays, and many more lost weekends.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Overview

Since ancient times [3], western logic has distinguished between *classical logic* that develops normative models of formal reasoning, and *dialectical* models that seek to describe reasoning between interested parties who hold divergent views. With the onset of computational machines in the modern era, the aim of logicians was to design machines capable of mathematical reasoning, such that they could automatically derive new formulae and proofs from a set of well defined initial axioms [96]. As such, a main part of the focus of early Artificial Intelligence (AI) research was on how to capture reasoning, based on classical logic, within a computational agent.

However, as computational machines have become commonplace in our day-to-day lives, existing in complex and human-oriented environments, they must increasingly deal with conflicting, incomplete, and inaccurate knowledge. Classical logic was not adequate to represent such informal reasoning, and struggled with modelling knowledge in the presence of uncertainty. Therefore, there has been a growing demand for computational systems that can deal with uncertain information, and for them to communicate with us in a natural way. Dialectical reasoning, with its ability to synthesise conflicting viewpoints, is an ideal approach to handle such information. Within the AI community, this has led to to an expanding interest in the field of argumentation: a social approach to dialectical reasoning, by which controversial standpoints can be reasoned about through a rational exchange of beliefs between intelligent entities.

Argumentation is a field that spans many disciplines. In ancient Greece for example, philosophers were interested in argumentation for its role in rhetoric [2], focusing on the ways in which arguments could best be delivered to influence audiences in public discourse. Human-oriented approaches to argumentation have been continued in fields such as linguistics [48], psychology [53], and neuroscience [97]. However, increasingly, argumentation is also studied from a computational perspective, seeking to establish strong, logical foundations for dialectical reasoning. Of specific importance is Dung's seminal work [33], which introduces an abstract approach to existing logics, allowing for them to be represented with a dialectical characterisation that makes them suitable for representing argumentative reasoning.

Dung's logical formalisation of argumentation provides a model of dialectical reasoning that is used to develop *argumentation systems* that inform and support human and machine reasoning. There are many types of argumentation systems that deal with different aspects of argumentation. A subset of these systems focuses on the exchange of arguments between agents, which, in this thesis, we refer to as *dialogue systems*.

An argument dialogue is a process by which agents exchange arguments in order to achieve their goals, either as individuals or as a group. Simply, dialogue systems aim to support agents in the dialogue process. These include systems that provide infrastructure, such as protocols and communication languages, to enable agents to exchange arguments. Dialogue systems also includes more strategic aspects of the dialogue, such as strategy generators that determine what utterances a dialogue participant should make in order to realise the participant's goals. Examples of applications of dialogue systems include software tools that support humans in conducting principled debates [26], computational agents that can act in dialogue situations on their human user's behalf [31], and the generation of strategies to encourage behaviour change in patients [58]. However, the computational complexity of strategic reasoning in dialogue is inherently non-trivial [34]. Indeed, current approaches to computing the optimal strategy for dialogues with a single agent have not been shown to scale well to increasingly large or complex domains.

The intuitively dialectical nature of argumentation allows even non-expert users to interact with and scrutinise argumentation systems. Argumentation is perhaps uniquely placed in being implementable in computational systems (because of its logical foundations) whilst also being naturally understandable by human users (because of the familiarity it has with common-sense reasoning). Thus, the true value of argumentation is in its ability to bridge the gap between human and machine reasoning, improving communication between users and machines, and increasing their understanding of one another [77].

However, humans do not always approach arguments consistently: who we are arguing with, what we are arguing about, and where we are arguing all influence the way in which we argue. We are adept at adjusting the way we argue to suit our current environment. For example, consider a group of politicians engaging in an argument as part of a parliamentary debate on a policy issue. Before a politician can speak in the debate, they must first *formulate* the arguments they wish to express. The formulation of their argument is a complex process. The politicians want to ensure their arguments are as effective as possible, and this might require the use of evidence such as statistics or an appeal to precedent. At the same time, they must ensure their arguments abide by the expected standard of the arena of discourse; for example, in the United Kingdom, parliamentary debates explicitly forbid participants from attacking the credibility of other participants. The politicians must follow strict rules in the way that they *exchange* arguments; typically they have a specified duration for which they are permitted to speak, and may only begin speaking when a moderator allows them to. Once the debate has come to an end, it may be appropriate to take some action based on the *evaluation* of the debate; for example, taking a vote on the policy to determine whether it is accepted.

Now consider a group of friends deciding on where they should eat dinner. Though explicit rules are not usually followed in such a discussion, there are still social norms that determine the way in which the discussion progresses. Thus, just like in the political debate example above, the formulation, exchange, and evaluation of arguments is influenced by the domain. In this informal example, participants are unlikely to have to provide evidence for the claims of their arguments. Furthermore, there is no strict turn-taking in friendly discussions, and it is more likely to be acceptable to interrupt another participant. At the end of the discussion, a formal vote will probably not be taken; rather the outcome for the argumentation will be some negotiated consensus.

Given that humans adjust the way in which they argue depending on the context and the domain, we would expect computational systems that do not also adjust their approach to argument appropriately to have varying success in different domains: whether you are a computational or human agent, approaching a political argument in the same way as a deliberation between friends is unlikely to result in success. Thus, we should ensure that computational systems involved in the argumentative process behave in a way that is optimised to the domain in which they operate. By understanding how different domain properties impact on the performance of dialogue systems, and seeing what properties an actual domain has, we can develop more efficient and effective systems for that domain.

Therefore, in this thesis, in light of the above, we seek to examine the extent to which the performance of argumentation systems can fluctuate depending on the varying characteristics of argumentation domains. Furthermore, we propose two approaches to strategic reasoning within dialogue systems that use approximation techniques to find acceptable (possibly non-optimal) solutions, which scale to domains beyond what is possible for current approaches that guarantee to find the optimal solution.

1.2 Motivation

We now explore further the motivation of this thesis. We begin by discussing the potential of argumentation to address two current hurdles for AI technology. Specifically in Section 1.2.1 we examine how argumentation can be used for communication between and with computational agents, and in Section 1.2.2 we examine how argumentation may also be used to enable better scrutiny of intelligent entities.

1.2.1 Interaction with computational systems

The increasing affordability of computer hardware has led to computational systems becoming ubiquitous in many aspects of our lives. As such, computer systems are no longer just stand-alone, but are typically made up of physically-distributed yet interconnected units. The field of distributed artificial intelligence considers how problems that are difficult for a single unit to solve can be overcome by several separate units working together [118]. As distributed systems become more complex, it is helpful to represent them as *multi-agent systems*.

Multi-agent systems consider individual units of distributed systems as having their own intentions, able to act on their beliefs and desires to achieve their own goals. Attributing such intent to computational systems can help us to understand the behaviour of such seemingly social and autonomous systems. Multi-agent systems have thus become a common paradigm in modern day software engineering, used to develop practical systems in a range of domains [68]. Indeed, the internet hosts many examples of multi-agent systems, from marketplaces and personal assistants, to social media and forums.

An inherent characteristic of agents is that they use social behaviours in order to achieve their goals in situations that they are not able to efficiently achieve them independently. Argumentation can be an effective formalisation for interaction between agents. In particular, argument dialogues are a suitable way for agents with possible conflicting viewpoints, and divergent interests, to exchange their beliefs about the environment. The ability for argumentation to be intuitively understood by humans makes it an especially powerful representation for agent interaction. It allows humans to communicate in a natural way as a member of an agent system comprised of computational agents.

1.2.2 Scrutability of computational systems

Early practical AI systems were based on the model of a single computational agent, endowed with expert knowledge in a specific domain. The canonical example of such a system is Mycin [23], an AI system used to aid physicians in the diagnosis of infections. These *expert systems* are able to offer novel information to the user, which typically takes the form of recommendations for action. Mycin enquires about the patient by engaging in a dialogue with the user, and then provides recommendations for appropriate prescriptions. An important aspect of expert systems is that they perform simple exchanges with the user which would help to provide the user with step-by-step information on how the system had arrived at its recommendation. The exchanges allow the user to scrutinise the reasoning process. This not only reassures the user that the process is sound, but also that it can allow feedback into the system to improve future reasoning. Indeed, Teach and Shortcliffe [105] revealed that the most important requirement for a user to accept a system is not how well it performs, but rather that it should "be able to justify its advice in terms that are understandable and persuasive...", and that "a system that gives dogmatic advice is likely to be rejected entirely". To ensure its reasoning was understandable, during its enquiry, Mycin would allow the user to ask for justification for why the current line of questioning would help to diagnose the problem. Furthermore, once Mycin had given a recommendation, the user could ask to see the full diagnostic trace of its reasoning.

More recent AI techniques, such as neural networks and machine learning, have demonstrated an ability to excel in a wide range of tasks [99]. However, one of the criticisms of such systems is that it is difficult to extract the reasoning processes by which the computational agent has arrived at their conclusions [49], in the way that expert systems were able to. This makes it difficult to justify taking important decisions based on the output of such systems, and makes it harder to convince a non-expert that such a system is truly trustworthy and reliable.

Such a limitation is especially troublesome in the modern era. Computational systems are now becoming sufficiently sophisticated that they may soon pose a threat to humans [119], and as such it has been argued that it has become a moral imperative that intelligent machines are endowed with the ability to explain their reasoning to human users [19]. Further, from a legal standpoint, as computational systems get more complex it becomes harder to identify who or what is at fault when things go wrong [8]. Therefore, it is vital that as modern AI techniques have a greater role in human society, their ability to justify their output in a human understandable way is significantly improved, as more traditional expert systems were required to do.

Given that argumentation allows human users to engage naturally with computational agents, it is a well-placed technology to fulfil this role in AI systems, so that they can be adequately scrutinised by their users. Indeed, argumentation systems have been developed for many real-world domains where the need for human scrutiny is required, such as in healthcare [40], law [9], and eGovernance [5].

1.3 Research aims

As discussed above, argumentation provides an intuitive approach to human-machine interaction, which is capable of becoming a key technology in the realisation of increasingly sophisticated and increasingly social computational agents. This has led to many dialogue systems being developed; however, they have been limited in several respects. Specifically, when evaluating dialogue systems, little regard has been given to the structural particularities of the domains in which they are being used. As a result, the relationship between the argument domain and the performance of dialogue systems is not well known.

A limitation of strategic reasoning in dialogue systems is that, due to their focus on generating only the optimal strategy for a dialogue, the current approaches do not scale computationally to larger domains. Thus, in this thesis, we intend to demonstrate the impact that the domain can have on dialogue system. Furthermore, we present two examples of how, through use of approximate techniques, we can generate efficient dialogue strategies that are still effective.

1. **Can we identify a domain characteristic that correlates with performance of a dialogue system, are we able to quantify the effect?**
   This questions is answered by investigating the extent to which there is a relationship between the similarity of agents' beliefs and the likelihood they can reach

consensus by engaging in an argument dialogue (Chapter 3).

2. **Is there a measurable relationship between the structural properties of a problem domain and their performance-related properties?**
   This question is answered by an investigation of structural classes of argumentation, derived from realistic instantiations (Chapter 4). It has been shown that the structures of these classes have a profound impact on a number of key properties relevant to dialogue systems.

3. **How can we develop effective and efficient approaches to strategic reasoning in dialogue systems?**
   We demonstrate that by considering the structural properties of argumentation we can develop systems that are advantageous when compared to the current state of the art. We do this by presenting novel examples of dialogue systems that do exploit the above-stated relationship. We present the two following examples.

   - A heuristic approach for determining beneficial arguments to put forward in a dialogue that considers only the structural properties of the argumentation framework. This is presented in detail in Chapter 5.
   - A evolutionary search approach to finding strategies for persuasion dialogues. This is presented in details in 6.

1.4 A note on methodology

The research presented in this thesis largely follows an empirical methodology: the results rely on observations derived from simulations and experiments. However, unlike other sciences that study the natural world, empirical research in AI requires a certain amount of engineering. Since the systems we study are on some level artificial, the experimentation relies on the suitability of the logical models and programs that have been engineered. This is both a benefit (the models we work with are often simple, especially compared to natural cognition), and a detriment (models must be a meaningful representation of a phenomena) for research. We must therefore be wary when conducting empirical research in the field of AI to ensure that the engineered systems of study are indeed consequential phenomena.

Thus, in this thesis the following steps are taken in the presentation of experiments. First and most importantly, the motivation is given for why the investigated phenomenon is relevant and interesting. Second, the logical models that underpin the experiment are presented. Third, we give the parameters that are manipulated and measured by the experiment. And finally, we present the results, with accompanying statistical analysis where appropriate.

1.5 Thesis structure

The remainder of this thesis is structured as follows.

- **Chapter 2, Background** Introduces the necessary technical background for the following chapters. We review the background literature.

- **Chapter 3** Demonstrates a relationship between the domain and the performance of dialogue systems. Specifically, we investigate how the similarity of agents' beliefs (e.g., the way in which arguments in an argumentation framework are distributed between agents) correlates with the likelihood that the agents are able to reach an agreement by engaging in a deliberation dialogue. The content of this chapter has been published in [78].

- **Chapter 4** Reinforces the existence of the relationship evidenced in the previous chapter by considering how the the low-level properties of a framework can affect the performance of an argumentation system. Specifically, we consider the effect of structures of generalised argumentation frameworks on key properties which are known to affect argumentation systems. We investigate structures of Dung-style frameworks, as well as two generalisations: extended argumentation frameworks that allow arguments to attack attack relations in order to express preferences, and collective-attack frameworks that allow sets of attacking arguments. We consider two case-studies based on existing argumentation systems, relating to statistical-model selection and clinical trials. The content of this chapter has been published in [81].

- **Chapter 5** Recent works consider mechanisms for determining an optimal strategy for persuading an agent of some particular goal argument. However, computing such optimal strategies is expensive, swiftly becoming impractical as the number of arguments increases. In response, we present a strategy that uses heuristic information of the domain arguments and can be computed with high numbers of arguments. Our results show that not only is the heuristic strategy fast to compute, it also performs significantly better than a random strategy. The content of this chapter has been published in [79].

- **Chapter 6** We explore a one-to-many persuasion setting, where a persuader presents arguments to a multi-party audience, aiming to convince them of some particular goal argument. The individual audience members each have differing personal knowledge, which they use, together with the arguments presented by the persuader, to determine whether they are convinced of the goal. The persuader must, therefore, carefully consider which arguments to assert, in order to maximise the number of convinced audience members. For reasonably sized problems with multiple audience members, it is computationally infeasible to search the space of all possible strategies. Instead, we use techniques from search-based model engineering to allow us to find an effective strategy for the persuader. We investigate performance of our approach on a range of settings, and in our evaluation we consider different structures and sizes of argumentation framework as well as varying the size of the audience and of the audience members' personal knowledge bases. Furthermore, we show that the approach is flexible enough to support multiple persuader objectives, allowing us to find persuader strategies that aim to minimise the number of arguments that are asserted while still maximising the number of convinced audience members. The content of this chapter is to appear in [80].

- **Chapter 7, Conclusion** A discussion of the main contributions of this thesis, and directions for further study.

Chapter 2

Background

2.1 Introduction

This chapter is divided into two parts. In the first, we present the technical background for the remainder of the thesis, including Dung's argumentation frameworks with accompanying semantics, and an overview of argumentation systems. In the second part, we review the relevant literature on computing argument dialogue strategies as well as work that has investigated the effect of domain on argumentation systems.

2.2 Technical background

2.2.1 Argumentation frameworks

An argument is defined as a set of *grounds* in support of a *conclusion*. Arguments can take the form "conclusion *because* grounds", where the grounds of the argument constitute the evidence in support of the conclusion. The conclusion may be derived from the grounds using deductive reasoning, as in the syllogistic example "Socrates is mortal *because* Socrates is a man and all men are mortal". Arguments can also rely on inductive reasoning; for example consider the argument "all ravens are black *because* all observed ravens have been black". In fact, an argument can be built upon any type of reasoning, including abductive, analogistic, and even fallacious reasoning. This is a desirable property as it allows formal reasoning to be represented alongside more common-sense reasoning. However, for this reason, a key aspect of arguments is their *defeasibility*, meaning that an argument can be invalidated if another argument is in conflict with it.

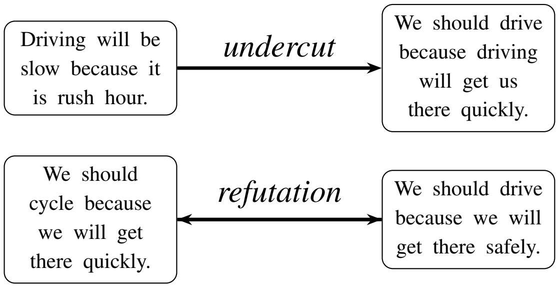

Driving will be slow because it is rush hour.
undercut
We should drive because driving will get us there quickly.
We should cycle because we will get there quickly.
refutation
We should drive because we will get there safely.

Figure 2.1: Instantiated examples of an *undercut* and a *refutation*.

An argument conflict is referred to as an attack relation. We can read an attack from argument a to argument b as “argument a is a reason against argument b”. It entails that it would be rationally incoherent to find both arguments acceptable at the same time. Note that if argument a attacks argument b, it does not necessarily imply that b attacks a. We consider two types of attack: the refutation and the undercut. If argument a undercuts argument b, then either the conclusion of argument a is not rationally coherent with one of the grounds of argument b, or the conclusion of argument a is not rationally coherent with the means by which the conclusion of argument b has been derived from its grounds. If argument a refutes argument b, then the conclusion of argument a is not rationally coherent with the conclusion of argument b. Examples of the undercut and refutation are shown in Figure 2.1.

Since Dung’s seminal work [33], the dominant approach to argumentation-based reasoning is to represent arguments as abstract entities in an argumentation framework (AF). Commonly represented topologically as directed graphs, AFs are comprised of a set of arguments and the attacks between them. See Figure 2.2 for an instantiated argumentation framework represented as a directed graph, and Figure 2.3 for an abstracted version of the same framework.

Definition 1. *An* ***argumentation framework*** *is a tuple $AF = \langle A, R\rangle$, such that A is a set of arguments*[1]*, and $R \subseteq A \times A$, is a set of attacks where $(x, y) \in R$ is an attack, x to y.*

The conflict relationship is central to the approach, and allows uncertain and possibly inconsistent knowledge to be formally represented. Since some of the arguments in a framework may be invalid, a key question is, given an argumentation framework, which arguments can be deemed to be valid? There are in fact multiple approaches, or

[1] In this thesis, we will consider only finite argumentation frameworks, where A is a finite set.

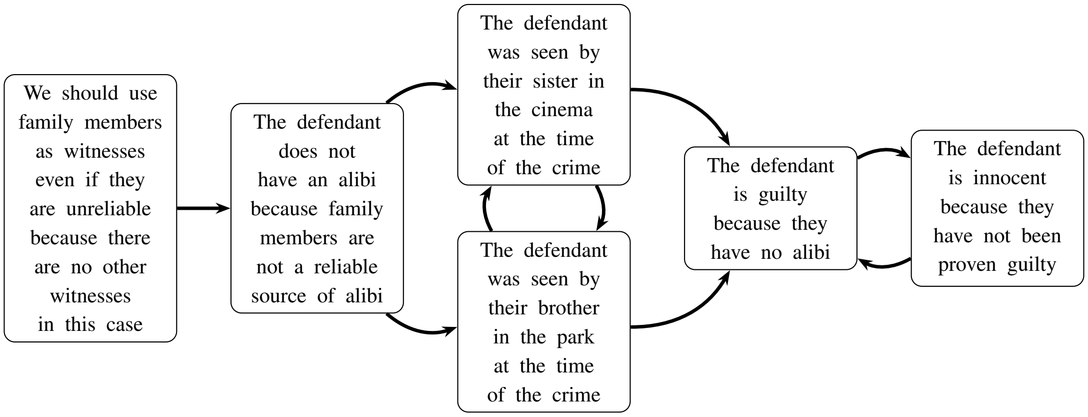

We should use family members as witnesses even if they are unreliable because there are no other witnesses in this case
The defendant does not have an alibi because family members are not a reliable source of alibi
The defendant was seen by their sister in the cinema at the time of the crime
The defendant was seen by their brother in the park at the time of the crime
The defendant is guilty because they have no alibi
The defendant is innocent because they have not been proven guilty

Figure 2.2: An instantiated argumentation framework.

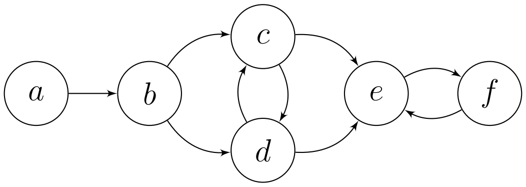


Figure 2.3: An abstract argumentation framework.

semantics, by which the *acceptable arguments*, those that can be justified with respect to the rest of the framework, can be inferred. However, each approach considers only the abstract arguments and the attack relations between them, and does not examine the internal details of the arguments. The acceptable arguments are thus determined by the underlying structure of the framework. The following section provides an overview of the semantics considered in this thesis.

2.2.2 Argumentation semantics

Argumentation semantics are based on the intuitive principles that it is not rational to accept any two conflicting arguments, and that an argument which is attacked can only be accepted if all of its attacking arguments are themselves attacked by an accepted argument [33].

The notion that a set of valid arguments should be internally-consistent, in that no argument in that set should attack any other argument in the set, is covered by the property of being *conflict free*.

Definition 2. *Let $\langle A, R\rangle$ be an AF and $S \subseteq A$, S is* **conflict-free** *iff $\forall x, y \in S$: $(x, y) \notin R$.*

Example 1. *Consider the AF in Figure 2.3, the sets $\{a, c, f\}$ and $\{b, f\}$ are both conflict free because there are no attacks between the arguments in them. The set $\{a, c, d, f\}$ is not conflict free, because there is an attack between arguments c and d.*

We can capture the intuition that a set of valid arguments should also be able to defend itself from any external counter-arguments. Therefore, any argument not in the set that attacks an argument within the set, should itself be attacked by an argument in the set.

Definition 3. *Let $\langle A, R\rangle$ be an AF and $S \subseteq A$, $x \in A$ is* **acceptable** *with respect to S iff, for all y such that $(y, x) \in R$, there $\exists z \in S$ such that $(z, y) \in R$.*

An argument x in an AF $\langle A, R\rangle$ is acceptable with respect to a set of other arguments $S \subseteq A$ if all arguments in A that attack x are attacked by an argument in S.

Example 2. *Consider the AF in Figure 2.3, and the set $S = \{a, b\}$. Argument c is acceptable with respect to the AF and S, as although there is an attack from argument $b \in S$ to c, there is an argument $a \in S$ which attacks b; we say that a effectively* defends *c.*

Definition 4. *Let $\langle A, R\rangle$ be an AF and $S \subseteq A$, S is* **admissible** *iff S is conflict-free and each argument in S is acceptable w.r.t. S.*

Example 3. *Consider the AF in Figure 2.3. The set $S = \{a, c, f\}$ is admissible since it is conflict free, and a, c, and f, are all acceptable w.r.t. S and the AF.*

There are a range of different semantics that build on these principles and determine sets of arguments that can rationally be presented as coherent [7]. These sets are known as *extensions*. Below, we survey some of these semantics (all from [33]), and highlight the ones that are used in the rest of this thesis.

Grounded semantics

An argument is acceptable under the **grounded semantics** if it is in the smallest set S such that every argument that is acceptable with respect to S is in S. The grounded extension is necessarily unique.

Example 4. *Consider the framework in Figure 2.3. The only argument acceptable under the grounded extension is a.*

Intuitively, the grounded semantics can be characterised as being cautious since only a relatively small number of arguments are acceptable under them. As such, sometimes the semantics may be considered too restrictive in the determination of acceptable arguments. However, the grounded semantics do provide a level of certainty in the arguments determined to be acceptable.

Example 5. *Consider the framework in Figure 2.2. Using the grounded semantics, we are not able to infer whether the defendant is guilty or not guilty, since the arguments with these as their conclusions are both excluded from the grounded extension. This is because the conflict between the two alibis prevents us from finding either of them acceptable under the grounded semantics.*

Preferred semantics

The use of the *preferred semantics* gives a lower threshold for acceptability when compared to the grounded semantics, potentially allowing more arguments to be inferred as acceptable than the grounded semantics. An extension is preferred if it is *maximally admissible*. A set S is maximally admissible if any of the arguments not in S were to be added to S then it would no longer be admissible.

Definition 5. *Let $\langle A, R\rangle$ be an AF and $S \subseteq A$, S is* **maximally admissible** *iff $\nexists e \in (A - S) : S \cup \{e\}$ is admissible.*

Example 6. *Consider the framework in Figure 2.3. There are two maximally admissible sets: $\{a, c, f\}$ and $\{a, d, f\}$.*

Since there are possibly multiple preferred extensions for an AF, to determine which arguments are acceptable we use some inference. An inference can be either *credulous* or *sceptical*.

An argument is acceptable under the preferred semantics with a credulous inference if it is part of at least one maximally admissible sets. Note that the set of arguments that are acceptable under the preferred credulous semantics is not necessarily conflict-free.

Example 7. *Consider the framework in Figure 2.3. The set of arguments acceptable under the preferred credulous semantics is $\{a, c, d, f\}$.*

An argument is acceptable under the preferred semantics with a sceptical inference if it is part of all maximally admissible sets.

Example 8. *Consider the framework in Figure 2.3. The set of arguments acceptable under the preferred sceptical semantics is* $\{a, f\}$.

We motivate the demonstrate the practical difference between the preferred credulous and preferred sceptical semantics in the example below, and show why each may be used over the other.

Example 9. *Consider the framework in Figure 2.2. Under the preferred credulous semantics, we accept both alibis and thus we find the defendant innocent; however, this introduces a conflict in our acceptable arguments as the alibis are contradictory. Under the preferred sceptical semantics, we are able to find the defendant innocent without introducing any conflict, but it is then less clear the reasons for why we have done so, since we find neither of the alibis acceptable.*

2.2.3 Argumentation systems

The development of logical formalisms for the representation of dialectical reasoning allows for the application of argumentation systems to a broad range of problems. Some of these argumentation systems, such as argument solvers, focus on the abstract computational challenges associated with argumentative reasoning. Other argumentation systems, such as those used for dialogues, are designed to support the social aspects of interaction.

Argument solvers

Many tasks in argumentation are computationally difficult [34], and therefore require the development of algorithms specialising in performing them efficiently. Argument solvers are programs that are designed to perform these tasks. There are many solvers, each using different techniques; some of them use techniques especially designed for argumentation (e.g. [93]), while others translate the problem to another domain and then solve it with an existing technology (e.g. [69]).

Examples of typical tasks that solvers perform are given below, but note this is not an exhaustive list, and some solvers may only perform a subset of the tasks [34].

- Compute an extension for a given framework and semantics.
- Compute all extensions for a given framework and semantics.
- Determine whether a given argument is credulously inferred for a given framework and semantics.

- Determine whether a given argument is sceptically inferred for a given framework and semantics.

Dialogue systems

The second type of argumentation system we consider are dialogue systems. Argument dialogues are a structured approach to rational interactions between parties. Participants in an argument dialogue engage in an exchange of arguments in order to achieve some collective goal, while at the same time seeking to achieve their own personal goals. Argument dialogues are of particular interest as they provide a structured means of communication for social agents [77], allowing agents not only to state their beliefs but also the reasons they have for holding those beliefs. Consider the following dialogue, in which two agents discuss how to travel to the park.

Edward: The weather is nice today, so we should cycle to the park.
Sophie: I don't want to cycle. If we take our bikes the traffic will slow us down, it will be faster to take the tube.
Edward: There is no traffic at this time of the day, so cycling will be faster than the tube.
Sophie: I do not have a helmet.
Edward: The segregated cycle paths will keep us safe.

We can see that by offering arguments and counter-arguments to one another, the agents are able to reason collectively. Each agent offers arguments that are personal to themselves, offering potentially novel information to the other. Through the exchange of new knowledge, the participants are able to influence the beliefs of one another.

Computational studies of dialogue began with Hamblin, in which he defines a set of dialogue games in logic (these games are simple, two-player, turn-based dialogues) [52]. The focus of Hamblin's work on dialogue games was not to investigate the behaviour of agents in dialogues, nor to design efficient mechanisms for dialogue, but rather to identify the logical circumstances that lead to *fallacies* occurring through the course of formal reasoning.

Much of the recent work on argument dialogues is centred around the exploration of different types of dialogue. Walton and Krabbe's initial typology of the various kinds of dialogue [115] is commonly used as a basis for these investigations. They define dialogue types based on the initial starting condition of the dialogue, the individual goals of participants, and the goal of the dialogue (in the context of agents, this is better thought of as the goal ascribed to all participants of the dialogue [71]). Table 2.1 summarises the canonical dialogue types set out by Walton and Krabbe.

Table 2.1: A typology of dialogues

Dialogue type	**Initial conditions**	**Individual goals**	**Group goal**
Information seeking	Individual ignorance	Acquire knowledge; spread knowledge	Spread of knowledge
Persuasion	Conflict in points of view	Convince others of your view	Resolve conflicting views
Inquiry	General ignorance	Obtaining new knowledge	Discovery of new knowledge
Deliberation	Need for group action	Obtain a favourable outcome	Reach a joint decision for action
Negotiation	Cooperation between self interested parties	Obtain a favourable deal	Arrive at an agreement
Eristic	Antagonistic parties	Inflict harm to others	Arrive at a provisional accommodation

We can classify the dialogue between Edward and Sophie as a deliberation dialogue. There is an initial need for group action, since the agents must decide on how to get to their destination. They both wish to arrive at a joint decision in order to get to the park. However, at the same time they both want to influence the decision towards a favourable outcome for themselves: Sophie has a preference to take the tube whereas Edward has a preference for cycling.

Although Walton and Krabbe's typology provides a useful way to categorise and discuss different kinds of dialogue, the way in which they are classified is informal. Dialogue games are a way to formally define instances of these types of dialogue [71]. They introduce the key components of a dialogue system, which are as follows. The valid utterances are formalised in a *set of locutions* agents can make during a dialogue game, alongside a *commitment store* that defines the meaning of each utterance. For example, a valid utterance may be to assert an argument known by the agent, which the agent is then committed to defend for the duration of the dialogue. *Combination* rules detail the sequences of locutions which are acceptable, and under what conditions they can be made. *Commencement* and *termination* rules specify how a game begins, and under what conditions, as well as to what outcome, it finishes.

Dialogue games have been defined for each of the dialogue types specified by Walton and Krabbe: for example, inquiry dialogues [17], persuasion dialogues [87], negotiation dialogues [73], and deliberation dialogues [104]. Dialogue games have been applied to real-world use in multiple domains, including medicine, eGovernance, and commerce. In the medical domain, argument-based dialogues have been used as a process by which a computational agent can help diagnose an illness by supporting a doctor [40]. The PARMENIDES (Persuasive ARguMENt In DEmocracieS) system designed by Atkinson *et al* [5] uses argumentation to support the public in consulting a computational agent representing a government body on political issues. Delecroix *et al.* have developed a virtual selling agent, that attempts to profile the customer over the course a persuasion type dialogue [31].

While dialogue games specify which moves are valid to be made by an agent, they do not address the issue of which of the valid moves should be made in order to maximise the chance of an agent to achieve their goals. This decision makes up part of an agent's *dialogue strategy*. Though some mechanism design seeks to diminish the role of strategy in dialogue [90], in more open settings of dialogues where agents can act on their self-interest, strategic argumentation is a central to an agent achieving their goals [106].

In general, strategic argumentation does not just focus on which moves are optimal to make in a dialogue, but also the way in which they are made. We can use the notions of *logos* (appealing to logic and reason), *pathos* (appealing to emotions of the listener), and *ethos* (appealing to the authority of the speaker) to distinguish between the different aspects of strategy. However, in this thesis as with the majority of computational research in strategic argumentation, the focus will be on the logos aspect of strategy, though a computational approach to ethos [88] and pathos [12] may also be valuable.

Most work in strategic argumentation considers a setting in which a proponent attempts to convince an opponent of the acceptability of a particular argument (commonly referred to as the *topic* argument, as it is the focus of the dialogue). However, assumptions about details of the domain greatly vary between work. In Section 2.3.1, we provide a critical review of the prominent approaches to strategic reasoning in dialogue systems.

2.3 Literature review

In this section we review the current state of the art in strategic argumentation, as well as literature that has considered different structures of argumentation frameworks in the evaluation of argumentation systems.

2.3.1 Strategic argumentation

In this section we review the state of the art in approaches to determining participant strategies for persuasion dialogues. Though direct comparison of the approaches is not possible due to the differences in assumptions, where the computational performance of the approach has been investigated, we give an indication of its scalability in terms of the number of arguments in the domain.

Hunter considers a setting in which the proponent has a model of the opponent's initial beliefs, which they can use to formulate the optimal strategy for the proponent by maximising expected utility [57]. In this setting, the persuasion is asymmetric, and the opponent's only behaviour is to indicate honestly whether they find the topic argument acceptable at certain stages of the dialogue. However, the approach has not been investigated through empirical evaluation, nor has the computational efficiency of the approach been evalauted.

Black *et al.* propose an approach to generating a dialogue strategy in persuasion dialogue through the use of AI planning techniques, to which they have applied to a range of different settings [16]. They model the dialogue as a planning problem, which can be given to a planner to generate the optimal strategies that an agent can follow. The planning problem consists of the initial situation (the arguments known by each dialogue participant), the possible actions available to the persuader (the set of assertions), and a set of goal states (those in which the persuadee is convinced). Similar to Hunter's approach, Black *et al.* assume that the persauder has a probabilistic model of the persuadee's beliefs, but also consider that asserting an argument may induce new knowledge in the persudee's beliefs. The strategies produced by the approach are given alongside a probability that the strategy will be successful in convincing the opponent agent. The approach has been shown to scale to domains of at most 13 arguments in the domain.

Hadoux *et al.* [50] support richer models argument dialogue, by using Mixed Observability Markov Decision Problems to model the dialogue that, after some minimisation, can be used to generate dialogue policies for the proponent. The policies give the proponent an optimal action for any possible point in the dialogue. Their approach has been shown, through empirical evaluation, to scale up to 8 arguments in the domain. It should be noted that bipartite frameworks are probably simpler to develop a strategy over, since arguments would either be fully attacking or fully defending the topic, and so the approach does not need to consider more nuanced arguments.

Rienstra *et al.* [92] build on the work of Oren *et al.*, and provide an approach to strategising in dialogue that involves sophisticated modelling of the opponent. Their approach allows the persuader to consider arguments that it itself is not aware of but believes are known to the opponent (referred to as virtual arguments); this is in contrast to other approaches that assume the proponent is aware of all the arguments that are known by the opponent. The approach of Rienstra *et al.* [92] was evaluated on frameworks of size 10 (in which the topic argument is always in the grounded extension), but specific investigation of the computational performance of the approach has not been presented.

Hadoux and Hunter use decision trees to model persuasion dialogues between two participants [51]. They represent all possible dialogues from an initial situation in a tree structure: an edge of the tree represents an asserted argument by either participants, nodes represent decision points where one of the participants must take an action, leaf nodes represent dialogue outcomes, and paths from root to leaf represent a possible dialogue. They seek to generate a policy, that can tell a proponent the best action in any node of the decision tree based on a probabilistic model of the opponents behaviour and beliefs. Rules from decision theory are used to determine optimal actions once they have optimised the tree structure. Their approach scales to domains of 8 arguments for frameworks that do not contain cycles, but they suggest this performance may be improved by further optimising the tree structure.

Focusing on persuading human users rather than computational agents, Rosenfeld and Kraus use machine learning techniques to predict the most beneficial arguments to assert during a dialogue [95]. They evaluate the effectiveness of their approach with experiments involving over 100 dialogues with human participants. Their results showed that human participants were at least as well convinced of the topic argument by their approach as they would be by another human.

Proof dialogues are similar to persuasion dialogues, in that they involve two agents asserting arguments in turn to one another to convince the other of the (un)acceptability of a particular topic argument. However, a key difference to typical persuasion dialogues is that in proof dialogues, agents argue over a shared argumentation framework whereas in persuasion dialogues agents are normally assumed to have their own personal argument frameworks. Caminada proposes an approach to a proof dialogue in which an agent may generate a sequence of arguments (similar to a strategy) in order to convince the opponent that an argument that is justified under the grounded semantics is, indeed, justified [24]. Moreover, the agent will not assert an argument that is not justified. The key difference to this approach from other work on strategy is that it restricts the agent to honest behaviours: an agent is only able to convince an opponent of what it truly believes to be the case by using arguments that it also believes. The strategy is therefore suitable for an explanation system, but cannot be used for more general dialogue goals where deceptive behaviours may be beneficial. Other approaches to proof dialogues exist for different semantics (e.g. preferred sceptical [100], credulous sceptical [113], and stable semantics [25]), and do not generally assume agents engage only in honest behaviours. Nevertheless, the proof dialogue approaches are still limited by the assumption of a shared argumentation framework. So while proof dialogues are appropriate for establishing the acceptability of arguments in a framework, or demonstrating its acceptability through dialogue, they cannot be applied to more general persuasion scenarios.

A common feature of the above mentioned argument-based approaches to determining dialogue strategies is that they all attempt to find the optimal strategy. Since the strategic problems are inherently computationally complex [34], approaches that attempt to find an optimal solution will, under the $P \neq NP$ assumption, have their worst-case performance become exponentially worse as the number of arguments in the domain increases. This is problematic because it limits the kinds of scenarios in which strategic reasoning is computationally viable, and therefore the number of real-world domains. An alternative approach would be to forgo the optimality of the found strategy, and instead attempt to find an acceptable solution more efficiently. Approximate techniques could be used to determine a dialogue strategy in domains in which would be infeasible to find optimal solutions, or just situations where computational resources are limited or expensive.

The use of an approximate heuristic when strategising in persuasion dialogues has been considered by Oren *et al*, who present a heuristic for minimising the amount of knowledge that is revealed by the persuader to the opponent during a dialogue [84]. The heuristic is not designed to be used to determine strategies that are necessarily effective at convincing the persuadee, nor does the approach seek to address performance issues with strategic reasoning and so the scalability of the approach is not investigated.

The role of strategy in dialogue has been considered independently of argumentation theory. For example, Shin has applied game theory to the domain of dialogue in order to analyse strategising with respect to the relationship between the quality of information available to the opponent and proponent, what information is withheld to each party, and the allocation of the burden of proof [101, 102]. Glazer and Rubinstein have applied mechanism design to generate rules on the process of strategising that optimises for the success of a desirable outcome [44, 45]. However, neither of these approaches directly tackle the problem of selecting which utterances should be made by participants in a dialogue setting.

2.3.2 How domain can impact performance of argument systems

In Section 2.2.3, we have presented an overview of argumentation systems. As discussed, argumentation systems are typically evaluated on arbitrary structures of framework, which therefore leads to limited results. However, some systems have been evaluated with different structures in mind.

We now review the current state of the art, by critiquing notable examples of evaluations that specifically consider argumentation framework structure. In the domain of dialogue systems, we review Black *et al*'s evaluation of their dialogue system. Further, we also consider, though not the focus of this thesis, evaluations in the domain of argument solvers.

Ladders and cycles in persuasion dialogues

Recall the approach to computing a dialogue strategy, proposed by Black *et al.* [16] from the previous section. Their approach is evaluated empirically by generating randomised initial starting scenarios to provide as input into the planner. The evaluation considers the effectiveness of the strategies that are generated, as well as the time taken to compute them. In initialising a starting scenario, the argumentation framework being argued over by the agents is randomly generated, and the arguments available to each agent is distributed. In the evaluation, three types of framework are considered. Bipartite frameworks are used as one type of framework, where each partition is distributed to a separate agent. These frameworks are considered to be strategically easier because the agents do not need to be concerned about undermining their own arguments. The other two framework types are non-bipartite. These two framework types are based on topological notions of *cycles* and *ladders*, and are devised to be especially challenging benchmarks. They are considered to be challenging because both types of structures contain arguments that may be both beneficial or detrimental for a persuader, depending on the persuadee's beliefs. They use a naive strategy as a comparison for the performance of their planning approach.

The results of Black *et al.*'s evaluation shows that while the planning approach is generally faster and more effective than a naive approach, and the performance of both approaches is at least partially determined by which type of framework is used in the initial scenario. More generally, the work demonstrates how the underlying framework structure can influence the performance of a particular dialogue system.

However, the two challenging benchmarks are particular to this approach to generating a persuasion strategy using AI planning, in that it is currently unclear whether these structures would be similarly challenging to other approaches to generating a strategy, or whether they are challenging for argumentation systems in general. While the structures have allowed some further insight into the evaluation of this argumentation system, other structures are likely to be more relevant for the evaluation of different argumentation systems.

Structural properties of frameworks on argument solvers

The impact of the structure of argumentation frameworks on the performance of argument solvers is recognised [77], but the relationship between the two is still not fully understood. Below, we review work that has used frameworks of different structures in the evaluation of the performance of argument solvers. Though the work below specifically investigates argument solvers, as examples of argumentation systems, they are still somewhat relevant to the consideration of dialogue systems in this thesis due to their use of argumentation frameworks as a domain. The empirical evaluation of the performance of argument solvers is a more mature area of investigation compared to the evaluation of the performance of dialogue systems and so they can offer insight for the work undertaken in this thesis.

- **The International Competition on Computational Models of Argumentation**

  The 2nd International Competition on Computational Models of Argumentation measures the efficiency of argument solvers at performing a number of different reasoning problems, across a selection of the most common semantics [43]. The competition requires argumentation framework instances on which the solvers run.

  For the 2nd International Competition on Computational Models of Argumentation, there was an open call for frameworks and framework generators, that could be used as input for the solvers. A total of six benchmarks were used. Most of the benchmarks used were randomly generated frameworks that were meant to be especially challenging for solvers. Only one benchmark came from real-world sources: a set of directed graphs from mass transit data that were interpreted as argumentation frameworks.

It is not clear whether the randomly generated frameworks have relevance to frameworks of real-world argumentation. Indeed, even the frameworks based on real-world structures are not within the context of argumentation or reasoning. However, this range of benchmarks do provide a range of differing structures on which the solvers were successfully benchmarked. From the results of the competition, it would appear that some of the types of structures of framework proved to be more challenging for solvers than other types. Further, not all solvers found the same structures challenging.

- **Features of frameworks and difficulty of problems**

  Given that the structure of framework used is known to affect the difficulty of computing extensions over that framework, a valuable direction of work is to identify which structural features of framework produce especially difficult problems. Identifying such frameworks can inform the design of solvers that are specialised in solving frameworks with those features more efficiently than a general solver is able to.

  The effect of framework features on the performance of solvers computing the preferred extension has been investigated by Cerutti *et al.* [39]. They use a range of topographic features relating to the structure of framework to predict which solver out of a set of four will be the most efficient in computing the preferred extension. In order to identify which are the most relevant features they use *predictive performance models*. Their results demonstrate which features are the best indicators of performance for the solvers.

  Rodrigues *et al.* investigate the computational complexity associated with computing the complete extensions across the domains that were used in the 2nd International Competition on Computational Models of Argumentation [94]. They found that certain domains can generate a very large number of extensions and this can be problematic for certain solvers, which fail to return a solution within the time limit set by the competition. Further, from their analysis, they propose a measure to estimate the difficulty of a framework based on its structural properties. The measures includes the size and number of strongly-connected components within the framework, the attack density of the strongly connected components, and the arrangement of strongly-connected components across the framework. They found that the measure was a fair indicator of framework difficulty for many of the domains used.

These approaches could be adapted to identify features of frameworks that affect the performance of dialogue systems. However, currently, such an analysis would be difficult due to the issues discussed in the Section 2.3.1 relating to the lack of a set of benchmark problems and domains for dialogue systems.

- **Social networks and argument solvers**

  Bistarelli and Santini [13] also consider how the underlying structure of a framework can affect argument solvers. Specifically, they have investigated the efficiency of argumentation solvers on different frameworks derived from the structures of social networks.

  They used randomly generated Erdős-Renyi networks, Watts-Strogatz networks, and Kleinberg networks as their models of small-world, directed, social networks.[2] They then interpret these as argumentation frameworks, and use them as the benchmarks for three solvers: Aspartix, Dung-O-Matic, and their own solver ConArg. Each solver was tasked with completing enumeration tasks for a range of semantics on each type of framework. They found that the performance of the solvers to complete these tasks varied substantially between the different types of framework that they investigated. Moreover, it was found that there was no solver that outperformed the other solvers for all the framework types, and that a solver that performs well on one type of framework may not perform well on other types.

  Bistarelli and Santini's results support the hypothesis that framework structure has a profound effect on the performance of argumentation systems. The authors' motivation for the use of social networks as benchmarks for argumentation systems is that argumentation is a social approach to reasoning. While the social aspect of argumentation is certainly important in the creation of argumentation systems [42], it is not evident that the structure of the social network of arguing agents would be retained in the structure of argumentation frameworks produced. It is therefore not obvious how relevant the structures of framework that Bistarelli and Santini investigate are to frameworks that would be encountered in real-world domains.

- **Semantic web instantiations and argument solvers**

  Yun *et al.* generate argumentation frameworks from knowledge bases based on existential rules in order to benchmark argumentation solvers [120]. The existential rules

[2] Erdős-Renyi networks were also used as a benchmark in the argument solver competition.

that they generate are based on the type of reasoning that occurs on the semantic web when in the presence of inconsistencies.

Though not based on real-world examples, the generator produces randomly generated knowledge bases that would be similar to those found on the semantic web. The knowledge bases are then translated to equivalent argumentation frameworks, ready to be used in the evaluation of solvers.

The evaluation itself mirrors the aforementioned argument solver competition, using a number of the same solvers, checking them with the same problems, and uses the same scoring system, but instead uses the new argumentation frameworks. The results show that, for many of the problems, when using the new frameworks, a significantly different ordering over the performance of the argument solvers is found.

- **Portfolios of argument solvers**

  By investigating the performance of solvers from the 1st International Competition on Computational Models of Argumentation, Cerutti *et al.* show that, in general, there is not a single argument solver that is best for all computational problems, but rather that the performance of solvers are complementary [37, 38]. They establish that collections of solvers (*portfolios*) can be employed to solve problems over a framework, where some initial analysis of the framework is undertaken to find which solver in the collection is likely to be most efficient at solving the problem for the given framework. They demonstrate through empirical evaluation that portfolios of solvers typically outperform standalone solvers.

  The results suggest that the structure of argumentation framework not only affects the performance of argument solvers, but may affect the performance of different solvers in different ways. This may be true of dialogue systems as well. It would be interesting to investigate whether framework structure affects the performance of dialogue systems in different ways, and indeed whether it is worthwhile to construct portfolios of dialogue strategies depending on the domain. However, portfolios are only possible if the systems that make them up can be directly compared. This is not currently the case for dialogue strategy generators as they are all developed for different forms of persuasion dialogue.

- **Framework representation**

  Not only can the structure of argumentation framework affect the performance of solvers, but the way in which the frameworks are represented can also have an impact of the speed of computation [36]. Cerutti *et al.* demonstrate that a framework with the same structure, but represented in file format in a different way (for example with a different ordering of arguments) can impact the performance of a solver.

Given that there are is standard representation of dialogue problems, this is not a priority for investigation within the dialogue domain. Nevertheless, if such a representation develops within the community, the impact of the representation on performance should be taken into account since it has the potential to affect the performance of dialogue systems given that it can affect the performance of argument solvers. Ideally, a representation of the problem should be optimised to improve the performance of all systems that use it, but such a representation may not exist.

2.4 Summary

In this chapter we have introduced the necessary technical background for the remainder of this thesis. We have reviewed argumentation systems, and we state that performance is an important aspect of dialogue systems and argumentation systems in general.

We reviewed recent work that considers framework structure in the evaluation of argumentation systems. While there is a competition for evaluating the performance of argument solvers that uses a range of framework structures, the vast majority of work on dialogue systems uses arbitrary frameworks and domains in their evaluation, and as a result, in many cases, there is not a strong understanding of how structure and domain affect the performance of dialogue systems. As a consequence, in the next two chapters of this thesis, we undertakes further analyses of how domain may affect the performance of dialogue systems. We begin by investigating deliberation dialogues, and whether the similarity of participants' initial beliefs correlates with the probability that agents are able to reach an agreement. Further, we generate a benchmark library of frameworks derived from popular generalisations of Dung's argumentation framework, and investigate the relationship between their low-level structure and their emergent semantic level properties.

We also reviewed work in strategic argumentation. In general, the problem of determining an agent strategy in a persuasion dialogue for effectively convincing a persuadee of a particular topic argument is far from trivial, and numerous approaches have been investigated. The approaches that directly address the issue all attempt to find optimal strategies. This comes at the cost of computational efficiency. So, while current state of the art solutions can produce optimal strategies, they do not scale to larger domains. This is potentially problematic if we expect practical applications to be developed for complex and human-oriented domains. In response to this issue, we propose two approximate solutions to strategic reasoning with a focus on scalability and performance. Specifically, we first propose a heuristic for estimating how beneficial arguments would be to assert in a persuasion dialogue, and use the heuristic to compute a proponent strategy. In our second approach, we use evolutionary search to find effective strategies. We demonstrate that both of these approaches are able to generate strategies in domains that would be likely computationally infeasible for current approaches that generate optimal strategies.

Chapter 3

The impact of domain on dialogue systems

3.1 Introduction

Autonomous agents must often collaborate with other agents to achieve their goals, for example when it is impossible or inefficient to achieve them as individuals. One way for a group of agents to coordinate their actions is to participate in a dialogue. Argument-based dialogues are structured interactions between participants, involving the exchange of formal arguments (e.g., [72]). There are many classes of argument dialogues, one such class being the deliberation dialogue, in which participants attempt to agree on an action. Such dialogues are a rational approach for agents to come to an agreement on how to act, allowing the opportunity for an agent not only to express their preferences over possible actions, but also to express the reasons they have for those preferences. Thus, deliberation dialogues are important as possible collaboration and coordination mechanisms that can be used in agent systems.

The complexities of agent-based argument dialogues mean that often only a limited number of properties can be studied formally without making overly restrictive simplifications to the problem domain [89]. This can make formal analysis of agent performance in dialogues difficult. A complementary approach is to use simulation and empirical analysis. An example of such an empirical evaluation is Black and Bentley's experiments on simulations with a deliberation dialogue system that found that the use of argument-based deliberation dialogues typically outperforms a basic consensus forming algorithm [15]. However, while their experiments explore a large and sensitive parameter space, they do not consider how the similarity of agents' initial arguments affects the dialogues (in their experiments they assume that agents have disjoint sets of initial arguments), which could be a contributing factor to the outcome of the dialogue.

In this chapter we also study the behaviour of deliberation dialogues using empirical methods. We investigate the dialogue system studied by Black and Bentley [15], first presented by Black and Atkinson [14]. We extend Black and Bentley's analysis by considering whether the similarity of the sets of arguments known by participants at the start of the dialogue affects the likelihood of whether agents successfully reach an agreement. The similarity of the arguments of an agent at the start of the dialogue can vary in real-world domains and so it is especially pertinent to understand how this property affects the outcomes of dialogues.

Our results demonstrate that the similarity of initial arguments has a statistically significant correlation with the likelihood of dialogue success for the investigated deliberation dialogue. We find that, in contrast to our intuition, the higher the similarity of initial arguments the lower the likelihood of success. We provide a justification for this relationship, and moreover, we analyse the extent of the relationship across the parameter space, helping to identify cases where the use of this specific deliberation dialogue can be used effectively. In the wider context of this thesis, through this example, we demonstrate that dialogue systems can have significant and surprising correlations with the outcome of dialogues.

The chapter is structured as follows. In Section 3.2 we recapitulate the model of the dialogue system originally presented by Black and Atkinson [14]. In Section 3.3 we describe our implementation and method of experimentation, including how we varied the similarity of the sets of arguments agents initially know about. In Section 3.4 we present the results of our experiments, including an analysis of observed trends and a detailed description of the relationships between variables. We discuss related work and other deliberation dialogue evaluations in Section 3.5. Finally, we conclude with a discussion in Section 3.6.

3.2 Deliberation dialogues

In this section we describe the model that defines the deliberation dialogues investigated in this chapter. This model is the same as that described by Black and Bentley [15], first presented by Black and Atkinson [14], which is based on the popular argument scheme and critical questions approach [114]. We use their model here because of its emphasis on practical reasoning. First we give details of the argumentation model that agents use to generate and evaluate arguments for and against different actions. We then describe the dialogue system used by agents to exchange these arguments, including the dialogue protocol that defines the structure of a deliberation dialogue, and the strategy that agents use to determine which of their arguments they will exchange.

3.2.1 Argumentation model

Our key concern is with the performance of the system specified in [14], in which agents have knowledge about the state of the world, about the preconditions and effects of actions they can perform, and about values that are either promoted or demoted by particular changes to the state of the world (these values represent qualitative social interests that an agent wishes to uphold; for example, fairness, health benefit, or personal privacy) [10]. An agent can use its knowledge to construct arguments for or against actions by instantiating a scheme for practical reasoning [4]: in the current circumstances R, we should/should not perform action A, which will result in new circumstances S, which will achieve goal G, which will promote/demote value V.

As a running example, we will consider the domain of two agents deliberating on how to travel to their shared destination of the local park. The current and new circumstances are models of what is currently true of the world, such as proximity of the agents to the park and weather conditions. There are a set of possible actions to perform, such as taking a car or cycling to the destination. The goal for the agents is to arrive at the destination. Values in the example are affordability, well-being, and timeliness; under typical circumstances, we could expect the action of cycling to promote the values relating to affordability and well-being (since cycling is free and is exercise), and demote the value of timeliness (since cycling is a relatively slow way to travel). An agent in this domain may be able to construct the following arguments for and against actions to achieve its goal (note that we omit the current and new circumstances from these arguments, assuming the reader can envisage appropriate instantiations).

- **A1:** We should *cycle* (action) because it promotes *well-being* (value) in achieving *getting to the park* (goal).
- **A2:** We should not *drive* (action) because it demotes *affordability* (value) in achieving *getting to the park* (goal).

- **A3:** We should *drive* (action) because it promotes *timeliness* (value) in achieving *getting to the park* (goal).

The scheme for practical reasoning is associated with a set of characteristic critical questions (CQs), which can be used to identify challenges to proposals for action that instantiate the scheme. These critical questions each relate to one of three reasoning stages: *problem formulation*, which considers the knowledge agents have about the problem domain (e.g., whether the preconditions and effects of actions are correct, whether state transitions promote or demote particular values); *epistemic reasoning*, where agents determine the current circumstances; and *action selection*, where agents construct and evaluate arguments for and against different action options. The deliberation dialogues we study here consider only action selection, assuming that the other stages have been dealt with previously with other types of dialogue; this action selection stage determines three CQs for consideration (we use the numbering of CQs used in [4]; see [14] for a more detailed justification of the appropriateness of these CQs).

- **CQ 6:** Are there alternate ways of realising the same goal? Two arguments that promote different actions attack one another.
- **CQ 9:** Does doing the action have a side effect which demotes some other value? An argument that demotes an action attacks an argument that promotes the same action.
- **CQ 10:** Does doing the action have a side effect which promotes some other value? An argument that promotes an action for some value attacks an argument that promotes the same action for a different value.

From these CQs we can identify attacks between arguments *for* and *against* actions to achieve a particular goal: two arguments for different actions attack one another (CQ6); an argument against an action a attacks another argument for the same action a (CQ9); two arguments for the same action that each promote different values attack one another (CQ10). Considering the example arguments given above, A1 attacks A3, A3 attacks A1, and A2 attacks A3.

Each agent has a (total-order) ranking over the values, referred to as its *audience*, which represents the importance it assigns to them. An agent uses its audience to determine the relative strength of arguments according to the values they each promote/demote, and thus whether an attack succeeds as a defeat. In the example above, an agent who finds well-being to be a more important value than timeliness will find argument A1 to be stronger than A3 and so will determine that A1 defeats A3, while A3's attack on A1 does not succeed as a defeat.

Given a set of arguments, the attacks between those arguments (determined by the CQs above), and a particular agent's audience, we evaluate the acceptability of an argument with respect to that agent with a Value Based Argumentation Framework (VAF) (introduced in [6]), an extension of the argumentation frameworks (AF) of Dung [8].

Recall from Definition 4, that in an AF, an argument is acceptable with respect to a set of arguments S if all of its attackers are attacked by some argument in S, and no argument in S attacks an argument in S. In a VAF, we say that an argument succeeds in *defeating* an argument it attacks if its value is ranked higher than (if the attack is symmetric) or at least as high as (if the attack is asymmetric) the value of the argument attacked (according to a particular agent's audience). Arguments in a VAF are admissible with respect to an audience A and a set of arguments S if they are admissible with respect to S in the AF that results from removing all the attacks that are unsuccessful as defeats given the audience A. In this chapter, we consider an argument to be acceptable to the agent if it is part of a maximal admissible set (a preferred extension) of the VAF evaluated according to the agent's audience.

For this chapter, we consider that an agent will find an action to be *agreeable* if they find some argument *for* that action to be acceptable. Considering the example arguments given above, if an agent prefers affordability to timeliness, which they prefer to well-being, they will find arguments A2 and A1 to be acceptable and conclude that the only agreeable action is to cycle (since this is the only action for which they have an acceptable argument). If, however, the agent prefers timeliness to well-being, which is prefered to affordability, they will find arguments A2 and A3 to be acceptable, and so will determine that driving is the only agreeable action to achieve their goal. Observe that arguments against actions are always acceptable given the instantiation of attacks derived from CQs and these are not considered by the agent in determining which actions it finds agreeable. Intuitively, this is because the CQs are concerned with evaluating presumptive proposals for performing some action. It would be possible (and we believe would not affect our experiments) to adapt the VAF generation and evaluation so as to produce the same results in terms of agreeability of actions while avoiding the (perhaps unintuitive) case where both an argument for and an argument against an action are found to be acceptable; we choose here not to adapt the model in order that our results are relatable to previous work [15, 14].

We can also see that (as in [14]) if an attack is symmetric, then an attack only succeeds in defeat if the attacked argument's value is more preferred than the value of the argument being attacked; however, if an attack is asymmetric, then an attack succeeds in defeat if the attacking argument's value is at least as preferred as the value of the argument being attacked. Asymmetric attacks occur only when an argument against an action attacks another argument for that action; in this case, if both arguments' values are equally preferred, then it is undesirable for the argument for the action to withstand the attack. If we have a symmetric attack where the values of the arguments attacking one another are equally preferred, then it must be the case that each argument is for a distinct action but promotes the same value; here, the attack does not succeed as a defeat, since it is reasonable to choose either action. We have described the mechanism that an agent uses to determine attacks between arguments for and against actions; it can then use an ordering over the values that motivate such arguments (its audience) in order to determine the acceptability of the arguments and, from this, the agreeability of actions. Next, we describe the dialogue system that agents use to jointly reason about the agreeability of actions.

3.2.2 Dialogue System

Deliberation dialogues take place between two participating agents (each with an identifier taken from the set $I = \{x, y\}$) and we assume that the dialogue participants have already agreed to participate in a deliberation dialogue in order to agree on an action to perform in order to achieve some mutual goal (this goal is the *topic* of the dialogue). At the start of the dialogue, each agent has available to it a set of arguments for and against actions to achieve the goal, which are those arguments it can construct from its private knowledge about the state of the world, the different actions that can be performed, and the values promoted or demoted by those actions. Each agent also has an audience (their personal ranking over the values).

During the course of the dialogue, agents take it in turns to make a single *dialogue move*. There are four types of dialogue move that participants may make:

- `assert` a positive argument (an argument *for* an action);
- `assert` a negative argument (an argument *against* an action);
- `agree` to an action;
- indicate that they have no arguments that they wish to assert (with a `pass`).

A dialogue terminates under two conditions: once two consecutive `pass` moves appear (in which case the dialogue is a *failure*, and no agreement has been reach), or two consecutive `agree` moves appear (in which case the dialogue is a *success*).

In order to evaluate which actions it finds agreeable at a point in the dialogue, an agent considers all the arguments it is aware of at this point and evaluates them as described in the previous section; it thus constructs a VAF consisting of the arguments it is initially aware of at the start of the dialogue and those arguments that have been asserted previously in the dialogue by the other agent, and evaluates this according to its audience. An action is *agreeable* to the agent if there is some argument for that action that it finds acceptable given this evaluation. Note that the set of actions that are agreeable to an agent may change over the course of the dialogue, due to it becoming aware of new arguments as they are asserted by the other participant.

A dialogue protocol specifies which moves are permissible for an agent x during x's turn in a deliberation dialogue with topic p as follows:

- It is permissible to `assert` an argument a iff the argument is for or against an action to achieve the topic p of the dialogue and a has not been asserted previously during the dialogue.
- It is permissible to `agree` to an action c iff either:
    - the immediately preceding move was an `agree` to the action c, or
    - the other participant $\overline{x}$ has at some point previously in the dialogue asserted a positive argument a for the action c.
- It is always permissible to `pass`.

While the dialogue protocol defines a set of moves it is permissible to make, an agent uses a particular *strategy* to decide which of the permissible moves to select. The *strategy* that the agents use is as follows.

- If it is permissible to `agree` to an action that the agent finds *agreeable*, then make such an `agree` move; otherwise
- if it is permissible to `assert` a positive argument *for* an action that the agent finds *agreeable*, then assert some such argument; otherwise
- if it is permissible to `assert` a negative argument *against* an action and the agent finds that action *not agreeable* then assert some such argument; otherwise
- make a `pass` move.

A dialogue terminates successfully if there are two consecutive `agree` moves, as both agents have managed to come to an agreement. A dialogue dialogue terminates unsuccessfully if there are two consecutive `pass` moves, as this implies that agents are unable to come to an agreement, having nothing beneficial to say to one another.

3.3 Investigating similarity

Previous work has considered whether there is a relationship between the number of unique values and actions being argued over, the number of arguments known by agents, and the likelihood agents have of reaching agreement through use of the deliberation dialogue system [15]. However, in those experiments the sets of arguments agents know at the start of the dialogue are always disjoint. It is possible, perhaps even likely, that in real world examples of agent dialogues there will be some overlap in the agents' initial argument sets. Thus, we are interested here in the question of whether the similarity of agents' initial arguments sets has an effect on the resulting dialogue.

To investigate this we perform experiments where we vary not only the number of unique values and actions being argued over and the number of arguments known, but also `sim` (a measure of the similarity of the sets of arguments known by each agent at the start of the dialogue). We thus require four parameters as follows.

1. `acts` : The number of unique actions that can be argued about.
2. `vals` : The number of unique values that can be promoted or demoted by the actions.
3. `args` : The number of arguments in the union of both agents' initial arguments.
4. `sim` : A measure of how similar the agents' sets of initial arguments are to one another.

To run experiments across the parameter space, we generate random dialogue scenarios; we initialise the two agents' argumentation frameworks (and hence the arguments that they each know at the start of the dialogue, referred to as their *initial arguments*) and their audiences. For each run of the simulation, the scenario generator is given `acts` actions, and `vals` values. It then generates all possible arguments that can be constructed from the set of actions and the set of values. For each action and value pair there are two arguments that can be produced, one argument that claims performing the action will promote the value, and the other argument that claims performing the action will demote the value. Therefore, the set of all possible arguments contains $2\times$ `acts` $\times$ `vals` arguments.

Then, random arguments are removed from the set of all possible arguments until it contains $\texttt{args}$ arguments. Note that if $\texttt{args} = 2\times\ \texttt{acts} \times \texttt{vals}$ then no arguments need to be removed. Half of the arguments remaining in the set are randomly distributed to one agent, with the other half being distributed to the other agent. The arguments that are distributed to an agent simulate the set of initial arguments that it can generate using its VAF. The set of initial arguments distributed to an agent x is denoted R^{x}.

It is clear to see at this point that R^{x_i} and R^{x_j} would be disjoint sets. However, this is not always the case in agent dialogues. Two arguing agents are likely to have some overlaps in their knowledge and hence may be able to generate and communicate the same arguments. We introduce the $\texttt{sim}$ parameter to determine how similar the sets R^{x_i} and R^{x_j} should be — the higher the value of $\texttt{sim}$ the more arguments that are shared between agents. So, once R^{x_i} and R^{x_j} have initially been determined, $(\texttt{args}/2) \times \texttt{sim}$ random arguments from each set are copied into the other set. It can be seen that after this sharing process, if $\texttt{sim} = 1$ then agents will have $\texttt{args}$ arguments each, and the arguments the agents each have will be identical. Similarly, if $\texttt{sim} = 0$ then the agents will have $\texttt{args}/2$ arguments each, and the arguments each agent has will remain disjoint (note, this is equivalent to the situation studied by Black and Bentley [15]).

The *total number of arguments* in a dialogue scenario refers to the sum of the number arguments initially known to one agent plus the number of arguments initially known to the other agent, and is calculated from the experiment parameters according to the following formula $\lceil \texttt{args} + (\texttt{args} \times \texttt{sim}) \rfloor$.

Our experiments investigate whether the similarity of agents' initial arguments has an effect on the simulated deliberation dialogues, across the following different parameter combinations.

- $\texttt{sim} \in \{0, 0.1, \ldots, 0.9, 1.0\}$,
- $\texttt{vals} \in \{2, 4, 6, 8, 10\}$,
- $\texttt{acts} \in \{2, 4, 6, 8, 10\}$,
- $\texttt{args} \in \{2, 3, \ldots, (\texttt{vals} \times \texttt{acts} \times 2)\}$.

Further, each agent has a total ordering over their values. In our experiments, this is randomly generated for each agent. Therefore, agents may have a different value ordering. This means that even if agents have exactly the same arguments they may not find the same outcome acceptable because they have different value orderings.

The randomised nature of the scenario generator and resulting simulated dialogue means that generated dialogues are not only sensitive to the input parameters, but also an element of chance. As a result, many dialogues must be simulated for each parameter combination: it is not sufficient only to run a single instance of a dialogue because two dialogues generated with the same parameter combination can still differ on the distribution of arguments among the agents, and the randomised aspect of the agents' strategy (agents select a random dialogue move when more than one is determined by the strategy). Thus, for each parameter combination, we simulate 1,000 dialogues and, for each dialogue, we record whether it ended successfully (with both agents having agreed on an action) or unsuccessfully (with agents failing to reach an agreement).

3.3.1 Relevance of framework structure and participant arguments

The structure of the argumentation frameworks of agents is determined by the critical questions and argument scheme used. This particular scheme has been developed for the purpose of practical reasoning, and has been shown to be relevant to many contexts [4]. Therefore, the argumentation framework structures we use are at least somewhat relevant to real-world scenarios.

The arguments known by the dialogue participants at the start of the dialogue are a random subset of the arguments from domain. This distribution may not be realistic. It is possible that different distributions of arguments to participants would have an effect on the dialogue's behaviour and outcome. Many other distributions other than a random distribution are possible. For example: it may be the case that if an agent knows an argument against a particular action for a specific value, then they are more likely to know further arguments against the same action for other values; or it may be the case that agents are more or less likely to know arguments for only a subset of their values.

However, it is not immediately apparent which distribution of arguments is realistic, or indeed whether it varies depending on specific agents and domains. A human study to establish which distributions are realistic could be undertaken through interviews and observations — but this is beyond the scope of our work. Therefore, in our experiments, we work with a random distribution with the acknowledgement that different distributions may be more appropriate for different problem domains.

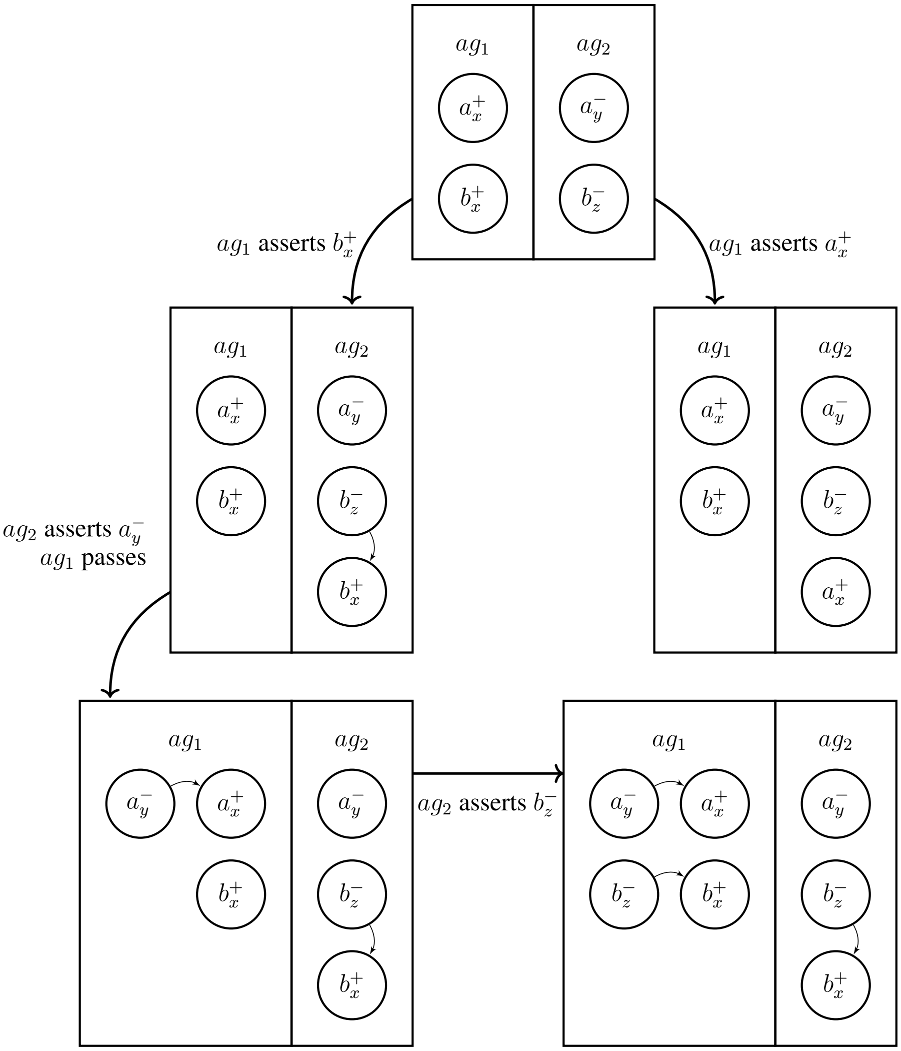


Figure 3.1: Partial flow on a deliberation dialogue between agents $ag1$ and $ag2$. Note, edges between arguments represents defeats not attacks.

3.3.2 Justification of empirical analysis

We use an empirical investigation to establish the outcomes of the deliberation dialogues rather than deriving an analytic solution because the outcome of the dialogue is hard to analyse for two reasons: (1) the outcome of the dialogue is not the same as the outcome of just taking the union of both participants' knowledge bases, and (2) the outcome is dynamically dependant on the moves selected by each agent which can vary even when using the same strategy since the strategy is non-deterministic.

The first reason this particular dialogue is hard to analyse formally is that the outcome of the dialogue cannot be established simply by analysing the agreeable actions that result in taking the union of both participants' knowledge bases. Even when there exists an action that is agreeable to both agents in the union of their knowledge bases it is still possible for the agents to fail to reach an agreement through the dialogue. Therefore, the distributed nature of the reasoning process makes it distinct from just a single agent reasoning over their own knowledge base. This property was demonstrated by Black and Atkinson [14].

The second reason the dialogue is hard to analyse formally is that the outcome of the dialogue depends on the moves selected by each agent during the dialogue. The strategy employed by agents is not fully deterministic, and so dialogues with the same initial configuration may still have different outcomes. We demonstrate this with the following example dialogue scenario, illustrated partially in Figure 3.1.

The example dialogue is between two agents, $ag1$ and $ag2$. They each know two arguments at the start of the dialogue: $ag1$ has an argument for action a promoting value x, and an argument for action b promoting value x; $ag2$ has an argument against an action a demoting value y, and an argument against an action b demoting value z. Both agents also have a preference ordering over the values as follows: $ag1$'s ordering is $z < y < x$, and $ag2$'s ordering is $y < x < z$. Initially, there are no defeats between the arguments in either of the agents knowledge bases. The first move of the dialogue is $ag1$'s.

Following the strategy, since it is not permissible to make an agree move, $ag1$ should assert a positive argument for an action that it finds agreeable. Therefore, $ag1$ can assert either one of its arguments in the first move. If it asserts the argument promoting action a with value x, then $ag2$ adds it to their knowledge base. Though $ag2$ has an argument demoting action a with value z, it is for a less preferred value as the argument it now has promoting the action ($y < x$). Therefore, $ag2$ now has an action is finds agreeable, and because $ag1$ just asserted the argument, it is permissible to make an agree move for that action. The agree move is then reciprocated by $ag1$ because it also finds action a agreeable, and the deliberation ends successfully.

Let's now analyse the other dialogue branch, and assume $ag1$'s first move is to assert the argument promoting action b with value y. $ag2$ then adds the argument to their knowledge base. However, this time, $ag2$ has an argument that defeats the new argument: the argument demoting action b with value z is preferred to the new argument ($x < z$). Since $ag2$ does not have any actions is finds agreeable, nor any positive arguments for an action, $ag2$ will assert a negative argument. There are two negative arguments $ag2$ can select. Let's assume $ag2$ selects the argument demoting action a

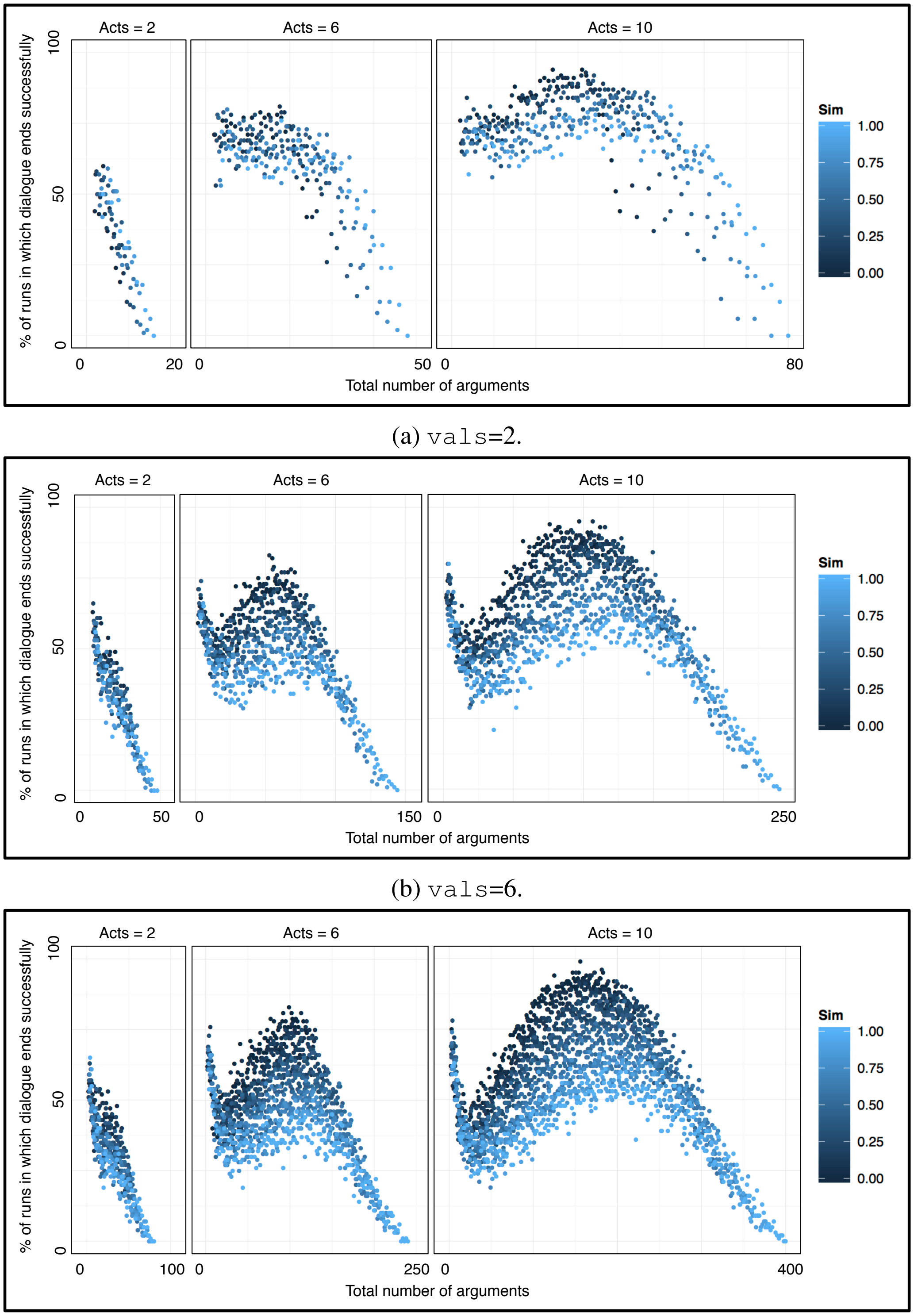
with value y. When $ag1$ adds this argument to their knowledge base, it defeats the argument it had for promoting action a with value x because it prefers value y to value x. On $ag1$'s turn, the only move it can make according to its strategy is to pass. Then, on $ag2$'s turn, it asserts its remaining negative argument. The new argument has no effect on the moves available to $ag1$, so it passes again. Finally, $ag2$ is left without any arguments to assert, so it reciprocates the pass move. The dialogue therefore ends unsuccessfully.

3.4 Results

Black and Bentley [15] also studied the likelihood of success across the parameter space studied here, but only for dialogues in which $\texttt{sim} = 0$. By limiting our parameter space to the dialogues in which $\texttt{sim} = 0$ we obtain a very close reproduction of their results: like them, we witness that successful dialogues are more likely with higher numbers of actions and values, and we can observe the relationship between the total number of arguments and the likelihood that the dialogue ends successfully (for low numbers of values and actions there is a decrease in the likelihood of dialogue success as the number of arguments increases, while for higher numbers the relationship is more complex, with likelihood initially decreasing as the number of arguments increases up to a certain point, after which the likelihood of dialogue success begins to increase).

However, by considering the different values of sim, we are able to make a number of empirical observations from which novel conclusions can be drawn. In each of the following subsections, we describe a particular aspect of our results, provide an explanation for what has been observed, and discuss the significance of the result.

Our results are shown in Figures 3.2a–3.3. Figures 3.2a–3.2c each present three graphs showing the the percentage of dialogues that end in success (y-axis), at different numbers of total arguments (x-axis), for different values of sim (the darker the shade of the plot, the lower the value of sim). The figures show the results for dialogues where $\texttt{vals} = 2$ (Figure 3.2a), $\texttt{vals} = 6$ (Figure 3.2b), and $\texttt{vals} = 10$ (Figure 3.2c). The graphs in each figure show the results for dialogues where $\texttt{acts} = 2$ (leftmost), $\texttt{vals} = 6$ (centre), and $\texttt{vals} = 10$ (rightmost). Each point represents the average of 1,000 simulated dialogues with that parameter combination. Similar results were seen across all combinations of vals and acts thus we present only a representative sample of the results here.



(a) `vals`=2.

(b) `vals`=6.

(c) `vals`=10.

Figure 3.2: Graphs to show the relationship between the total number of arguments and the percentage of dialogues that ended successfully, for different values of `sim` when `vals`=2,6,10 (1000 runs for each parameter setting).

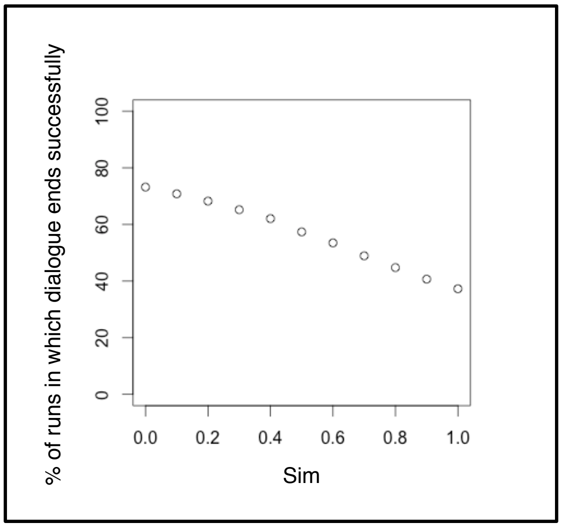


Figure 3.3: A scatterplot to show the relationship between the similarity of initial belief sets and the rate of success of the dialogue averaged over the total number of arguments (1000 runs for each parameter setting), in dialogues where `vals`=10 and `acts`=10.

3.4.1 Dialogues tend to fail with many arguments

From the results in Figures 3.2a–3.2c we can see that dialogue success is very unlikely at high levels of total arguments (every graph tails off into a 0% rate of dialogue success as the number of total arguments tends towards its maximum value for the parameter combination). We suggest that the reason for this is that if an agent believes every possible argument over a set of values and actions then it will find no action acceptable: all arguments for doing a particular action because that action promotes some value will be defeated by the negative argument that demotes that action for the same value, and hence the action will not be agreeable to the agent. In the case where agents start the dialogue with every possible argument over a set of values and actions, the agents begin the dialogue finding no actions agreeable and have no possibility of ever finding an action agreeable (since they know all arguments, no asserted argument during the dialogue will change the actions that are acceptable); this corresponds to the plot in the graphs where $\texttt{sim} = 1$, and the total number of arguments is $2 \times \texttt{acts} \times \texttt{vals}$.

This observation cannot be made without considering dialogues in which $\texttt{sim} \neq 0$ because the lower the similarity, the lower the number of total arguments, and so at low similarities, dialogues cannot have a large enough number of arguments to reveal this trend. This can be seen in Figures 3.2a–3.2c where no plots for $\texttt{sim} = 0$ exist beyond 50% of the graphs' maximum of the number of total arguments.

Beyond a certain point, the more arguments an agent knows, the greater the chance that dialogue success is impossible, and this effect becomes severe at high levels of total number of arguments. Thus, it is not the case that the complete failure of dialogues for very high number of total arguments is the fault of the dialogue system but rather is down to the likely impossibility of an agent finding any action agreeable when believing this many arguments. In real-world scenarios, agents are unlikely to have knowledge of so many arguments at once and so we consider these types of dialogue to be unrealistic.

Thus, importantly, our results show that when using the deliberation dialogue, agents will not come to an agreement when it would not be rational for them to agree to do any of the possible actions. This result was proven theoretically Black and Atkinson [14].

3.4.2 Dialogues are less successful as similarity increases

Given these initial results, we investigated whether the likelihood of success of a dialogue (measured by whether the dialogue ends in agreement or not) correlates with the similarity of the two agents' initial arguments (measured by the `sim` parameter). Looking at Figures 3.2a–3.2c, we can see how the `sim` parameter correlates with the rate of dialogue success across different numbers of values, and actions, and total numbers of arguments. Perhaps surprisingly, the general trend is that agents that have similar sets of initial arguments are less likely to reach an agreement compared to agents that have dissimilar sets of initial arguments. The result goes against the intuition that agents with similar knowledge should be able to agree more easily.

We assessed the relationship between the similarity of agents' initial arguments and the rate of success of the dialogue averaged over the total number of arguments in dialogues where the number of actions was 10 and the number of values was 10. This assessment was undertaken by calculating a Pearson product-moment correlation coefficient, which showed that there is a very strong, negative relationship between the two variables (coefficient $r = -0.96$, statistical significance $p < 0.001$), indicating that by increasing the similarity of agents' initial arguments the likelihood of the dialogue is less likely. The scatterplot in Figure 3.3 displays these results.

We explain this relationship as follows. When a dialogue is initialised with $\texttt{sim} = 1$ (i.e. agents' initial sets of beliefs are identical) any argument an agent asserts will already be known by the other agent. In these dialogues, the agents' sets of known arguments remain the same throughout the dialogue (since any asserted argument will already be known by both agents) so the actions an agent finds agreeable at the start of the dialogue will remain the same at every subsequent turn of the dialogue. If agents do not have any agreeable actions in common at the start of the dialogue, then they never will, and so the dialogue will fail. Considering the other extreme, when a dialogue is initialised with $\texttt{sim} = 0$ (i.e. agents' initial arguments are entirely disjoint), any argument an agent asserts throughout the dialogue will be novel for the other agent, potentially changing the actions it finds agreeable, and hence the actions that are agreeable to both agents. The more often an assert move changes the actions agreeable to both agents, the more likely it is that throughout the course of the dialogue there will be a point at which there is at least one action agreeable to both agents. In summary, the lower the similarity of the initial arguments, the greater the chance there will be at least one point in the dialogue at which agents mutually find at least one action agreeable, and hence the greater the chance that the dialogue will be successful.

Understanding the relationship between the similarity of the arguments known to agents at the start of the dialogue and the likelihood of the dialogue succeeding is important in understanding the situations in which deliberation dialogues are a useful method for agents trying to reach an agreement for action, and this can help to identify real-world scenarios in which this technique can usefully be applied.

3.4.3 The impact of similarity increases with the number of values

Varying sim for dialogues with a low number of values produces a relatively small effect on the likelihood of the success of the dialogue. For example, dialogues with 2 values are changed only slightly by changing sim — as can be seen in Figure 3.2a, the distances between plots for $\texttt{sim} = 1$ and $\texttt{sim} = 0$ are low, within 15%. Looking at Figure 3.2b where the dialogues have 6 values, the distances between plots for $\texttt{sim} = 1$ and $\texttt{sim} = 0$ are wider in general, and this is evidence of an increasing effect of sim at higher values. The distances are greater still for dialogues with 10 values, as seen in Figure 3.2c, where we observe a nearly 50% difference in the likelihood of success of the dialogue between dialogues where $\texttt{sim} = 1$ and $\texttt{sim} = 0$.

Generalising these results, we can say that the when agents have similar sets of initial arguments then the likelihood of dialogue success increases as the number of values that agents argue over increases. This tells us that in dialogues where a high number of values are being argued over, similarity has a strong relationship on the likelihood of dialogue success, and therefore it is especially pertinent for similarity to be considered in these scenarios.

3.4.4 Dialogues succeed at around 50% of the maximum total arguments

For dialogues in which $\texttt{vals} = 2$ or $\texttt{acts} = 2$ we observe a general decrease in the likelihood of dialogue success as the total number of arguments increases. Furthermore, for dialogues in which $\texttt{acts} = 2$ we observe a decrease in the likelihood of dialogue success as the total number of arguments increases, regardless of the number values. This relationship can be seen in the relevant graphs in Figures 3.2a–3.2c, and was also observed by Black and Bentley [15].

The relationship between the total number of arguments and the likelihood of success is more complex when we consider dialogues in which $\texttt{vals} > 2$ and $\texttt{acts} > 2$. The relationship can be described in three stages. First, in the lowest 10% of a graph's maximum total number of arguments we observe a decrease in the likelihood of dialogue success similar to that in lower numbers of values and actions. However, in the second stage, after the 10% point up to approximately 50% of a graph's maximum total number of arguments, the trend reverses and we observe an increase in the likelihood of dialogue success as the total number of arguments increase. The trend reverses again in the third stage, after 50% of a graph's maximum total number of arguments onward, where we observe a tail off towards a 0% likelihood of dialogue success. This relationship can be seen in the relevant graphs in Figures 3.2a–3.2b. This more complex relationship was not observed by Black and Bentley [15] because very high total numbers of arguments can only be reached by considering $\texttt{sim} > 0$.

Dialogues with a low sim are less affected in the initial stage of the relationship and are more greatly affected in the second stage (the trough is shallower, and the peak is higher), whereas dialogues with a high sim are more affected in the initial stage of the relationship and are less affected in the second stage (the trough is deeper, and the peak is lower).

The shape of the relationship between the total number of arguments and the likelihood of dialogue success as described here would have been extremely difficult to prove using formal methods. However, by using the experimental approach we are able to investigate performance across the entire parameter space. The observation of the shape of the relationship is useful because it allows us to predict accurately the chance a dialogue will succeed for any given parameter combination.

3.4.5 Wider applicability of results

The results presented above are limited to the specific deliberation dialogue system under investigation, where agents use the value-based argumentation framework representation of their beliefs and also follow the specified dialogue protocol when acting. It may be that the dialogue protocol or framework structure are a factor in determining the results that we have established, and that the similarity of beliefs is only a correlation. Nevertheless, the systems we investigate here are widely used in reasoning tools and agreement technologies. So while our results may be limited to these properties, they are relevant systems to study.

The argument scheme that the agents use is designed for value-based reasoning, which has been shown to be useful for practical reasoning . The scheme has been the basis of reasoning tools in many domains such as medicine [6], policy decision [111], and law [46]. So while our results do not generalise to structures outside of the chosen argumentation scheme, the structure is widely used in systems. Furthermore, deliberation dialogues are an important class of dialogue system [114]. Moreover, they have been shown to perform better in some scenarios at allowing agents to reach agreement over consensus algorithms [15].

3.5 Related work

Our experiments are related to those of Black and Bentley [15], which are based on the same argumentation model and dialogue system [14] as the work presented in this chapter. Their work was perhaps the first to use empirical methods to evaluate the benefit of using deliberation dialogues. In their experiments, they vary the number of values and actions being deliberated over, and the number of arguments available to agents at the start of the dialogue and show that the deliberation dialogue system typically outperforms consensus forming. Here, we expand the parameter space to also vary the similarity of the arguments that the agents have and show that this is an important factor in the success of a deliberation dialogue.

Kok *et al.* similarly take an empirical approach to the investigation of argument-based deliberation dialogues [64]. They focus on the expressive potential of argumentation by using a deliberation dialogue system that allows agents to communicate using elaborate arguments, assuming that agents that are able to express themselves better would be able to perform more efficiently in argument dialogues. They show that an arguing strategy offers increased effectiveness over a non-arguing strategy. In their work, agents' arguments are generated from their respective knowledge bases, but they do not consider how the efficiency of the dialogues depends on the similarity of the agents' respective knowledge bases, or the similarity of arguments that are generated from them.

In considering groups, Toniolo *et al.* investigate how argument-based deliberation dialogues can be used by a team of agents that have their own potentially conflicting goals and norms [110]. Using an empirical evaluation of their model, they find that argument dialogues are a more effective means of agent coordination than collaborative plans (using the metric of the feasibility of the resulting plan). While their work does consider agents as heterogeneous with their own goals and norms, they do not consider how the similarity of their goals and norms (and hence their arguments) affects the quality of the plans produced.

Finally, Medellin-Gasque *et al.* present a dialogue protocol for deliberation and persuasion dialogues, in which agents argue over cooperative plans [74]. Interestingly, the protocol allows for the type of the dialogue to change at a specific point, and thus allows the dialogues to be somewhat dynamic. Similar to our work, their dialogue system is based on the critical questions approach [114]. They implement 3 different agent strategies (a random strategy, and 2 strategies that place some priority over dialogue moves), which they test over a limited number of cases (20 initial states, generated from 4 different sets of information, and 5 different preference orders over values). Their results show that, for the cases and strategies tested, the quality of the outcome of the dialogue does not vary by altering the agents' strategies, but by using a priority strategy rather than a random strategy, the outcome can be reached more efficiently. Thus, agents' dialogue strategies can be an important consideration for dialogues, in at least some initial circumstances.

3.6 Conclusions and Discussion

Our results show how, in the argument-based deliberation dialogues investigated here, the similarity of agents' initial arguments correlates with the likelihood that a dialogue ends in success. We found dialogues with high similarities of initial arguments are less likely to end in agreement than dialogues with low similarities of initial arguments, because the higher the similarity of initial arguments the less potential for agents to reach a point in the dialogue at which there exists at least one action that is agreeable to both agents. Using an empirical approach, our investigation allowed a total analysis of the parameter space over a large sample size of dialogues. Our results identify scenarios in which using a deliberation dialogue is likely to lead to an agreement being reached, and scenarios when a deliberation dialogue is not likely to lead to an agreement where other agreement technologies may be more helpful in forming consensus.

In our investigation we explored the entire range of possible similarities of agents' initial arguments: from dialogues where agents started with entirely disjoint sets of initial arguments to dialogues where agents started with identical sets of initial arguments. Across this range we identified a statistically significant correlation of similarity on the likelihood of dialogue success, but, it is unclear to what extent this range typically exists in real-world scenarios. The relationship between the sets of initial arguments we randomly generate to those seen in real-world applications is also not understood (for example, dialogues that were generated with a very high number of total arguments are probably not realistic). The lack of real-world data is an identified problem in research relating to applications of argumentation.

There is a question as to whether measuring the quality of a deliberation dialogue simply on whether agents reach an agreement is the best or only measure. According to Walton and Krabbe [115], while there is a *public* goal to find an agreement that is ascribed to by both agents in a deliberation dialogue, agents also have a *private* goal to influence the agreed upon action to one that is as favourable as possible to itself. Working out a suitable metric for the success of an agent's private goal is non-trivial as it is unclear how to accurately measure the influence an agent has had on the dialogue, and it is unclear how to measure which action is an agent's most favoured (should it be the agreeable action that promotes the highest value given local beliefs of the agent, or given global beliefs of the system). There are also other factors that could be used to measure the outcome of the dialogue: efficiency and speed of the dialogue (what resources were spent during the dialogue?), soundness of the agreed upon action (is the agreed upon action the best course of action from a global perspective?), and fairness (is the outcome representative of all of the agents' preferences?). For example, Black and Bentley assign scores to dialogue outcomes, depending on whether the agreed upon action is globally agreeable to both, one, or neither agent. However, there are many other possible ways to measure the quality of a deliberation dialogue.

Walton *et al.* question whether models of deliberation dialogues are able to actually capture the richness and depth of human-like deliberation dialogues [116]. Specifically, they consider dialogues in which information available to participants of the dialogue is dynamic. This is certainly a limitation of our investigations since the knowledge the agents have remains the same throughout the duration of the dialogue. If we extended the dialogue system to simulate changing knowledge of the environment during the course of the dialogue, an interesting investigation would be to see how the similarity of the information/arguments made available to both agents would affect the dialogue (i.e., what happens if the information made available to agents becomes gradually more different or if the information becomes gradually more similar?).

Though the investigations in this chapter consider the similarity of agents' initial beliefs, they do not consider the similarity of agents' audiences (the ordering of their preferences over values). It may seem reasonable to predict that the more similar agents' preferences, the more likely they are to come to agreement. However, this hypothesis has not been tested, and we leave this for future work.

The dialogue system investigated in this chapter allows for agents to argue about their beliefs, but not about their preferences. Giving agents the ability to argue about their preferences would allow for more sophisticated dialogues, and therefore may allow agents to reach agreement more often. However, in some settings it may be preferable that agents cannot argue about their preferences since agents may not wish to get into an overly sophisticated debate: in human-oriented domains one may want to discourage complex reasoning to ensure that the dialogue is easily understandable by a human, or in time-critical domains there may not be the computational resources available to facilitate more complex forms of dialogue.

In summary, the results presented in this chapter evidence that the domain used in dialogue systems can have correlate in large and unexpected ways with the performance of the system. In this chapter we have considered the similarity of participants beliefs with the outcome of a type of deliberation dialogue. Across domains, the structural properties of argumentation framework can also vary; this idea is investigated in Section **??**. Do the structural properties of the argumentation framework influence the performance of dialogue systems? In Chapter 4, we investigate the relationship between the structural properties of argumentation systems (including dialogue systems) and their emergent semantic-level properties.

Chapter 4

Characteristics of generalised argumentation frameworks

4.1 Introduction

While progress has been made in the development of practical argument-based systems (from abstract solvers [22] to agreement technologies [77]), evaluations of such systems are limited. A significant challenge in the evaluation of developed argument-based systems is the lack of repositories of argumentation frameworks from real-world domains or applications [30]: currently, systems are typically evaluated on randomly generated frameworks, with little consideration of the *structure* of such frameworks. However, small differences in framework structures can have large, unexpected effects on the performance of some argumentation systems. Indeed, in Section 2.3.2, we reviewed work that had considered the effects of structure on the evaluation of argumentation systems, all of which found the framework structure had a significant effect on the performance of the respective system. Further, in the previous chapter, it was evidenced that the way arguments are distributed between dialogue participants in a deliberation dialogue influences whether they manage to come to an agreement.

Given that the underlying structure of argumentation have a profound impact on the performance of argumentation systems, it is important that argumentation systems are evaluated using relevant structures of framework. The identification and property analysis of relevant structures of argumentation frameworks are central problems currently facing the argumentation community: understanding the properties of these frameworks not only allows for a more grounded evaluation of argument-based systems, but can motivate the development of systems optimised for specific frameworks and structures.

Many different generalisations of classic Dung-style frameworks have been proposed [21], each of which has its own structure. Because of this difference in the structures of generalised frameworks, there is also likely to be differences in their characteristics. Against this background, in this chapter, we consider structures derived from different generalisations of argumentation framework (specifically, we consider the extended argumentation framework [76] and the collective-attack framework [83]). These structures are particularly relevant as current argument technologies are already developed to use generalised frameworks (*e.g.* [75, 82]).

In our investigation of structures of generalised argumentation frameworks, we measure three key properties: the size of the grounded and preferred extensions of the frameworks (known to affect the computational speed of argument solvers [28]); the proportion of argument subsets of the framework in which a topic argument is acceptable (known to be a factor in the *effectiveness* of dialogue strategies for persuasion and deliberation [15, 78]); and whether the addition of a new argument to the framework results in a change of acceptability of a topic argument (a type of dynamic argumentation, which is another factor in the *efficiency* of dialogue strategies [1], and may be a key property for improving the computational efficiency of a variety of other argument-based systems [67]).

The key contribution of this chapter is an experimental analysis of three different argumentation frameworks (Dung-style, extended, and collective-attack), with a consideration of relevant properties (extension size, subset acceptability, and dynamic argumentation). We begin by introducing the argumentation frameworks we investigate in this chapter. Next, we describe specific structures of these frameworks that we use in our experimental comparisons. We then detail the properties we measure, present the experiments we run, and discuss the results. We conclude with a discussion of related work.

4.2 Generalised frameworks

Though argumentation frameworks are expressive, many generalisations have been proposed which provide explicit representation of relationships other than attacks between arguments, seeking to more intuitively capture particular aspects of argumentation [21]. The large number of proposed extensions is perhaps unsurprising when considering the practical issues of developing argumentation-based systems for a diverse range of real-world problems [77].

Below, we provide the background and definitions for two popular extensions of argumentation frameworks which we focus on in this chapter: *extended argumentation frameworks* (EAFs) which allow arguments to attack attacks in order to express preferences between arguments [76], and *collective-attack frameworks* (CAFs) which allow argument sets to attack arguments in order to express more complex attack relations [83]. In this chapter, we refer to AFs as defined by Dung (Definition 1) as Dung-style argumentation frameworks (DAFs) to distinguish them from EAFs and CAFs.

CAFs and (some) EAFs can be translated into equivalent DAFs [85]. So why use the generalised form of frameworks at all, if we can represent them equivalently as DAFs? The generalised frameworks may still be valuable to use as they are typically more compact representations, requiring fewer arguments than their DAF equivalent; this can be beneficial for reasons relating to computational efficiency. Moreover, these extension may represent the underlying meaning of the framework more intuitively for a human agent.

4.2.1 Extended argumentation frameworks

EAFs allow the representation of arguments that attack attack relations [76], see Figure 4.1 for an instantiated example of an EAF. Given an argument a which attacks b, an argument c may attack the attack between a and b. In this way, an EAF may be used to capture (possibly conflicting) preference relations between arguments. For example, see Figure 4.2 in which c represents a preference for a over b, which conflicts with d representing a preference for b over a.

While it may be possible to represent preferences in Dung-style graphs without any extension, EAFs provide an intuitive and succinct way to represent these preferences. Though there are other generalised frameworks by which preference relations can be modelled in argumentation (e.g., [61]), EAFs are an especially expressive model as they represent preferences as defeasible arguments, allowing agents to argue about their preferences and, powerfully, about preferences over other preferences.

Definition 6. *An **extended argumentation framework (EAF)** is a tuple $\langle A, R, D \rangle$ s.t. A is a finite set of arguments, $R \subseteq A \times A$ is a set of attacks,*

- *$D \subseteq A \times R$ is a set of attacks on attacks, and*
- *if $(z, (x, y)), (z', (y, x)) \in D$ then $(z, z'), (z', z) \in R$.*

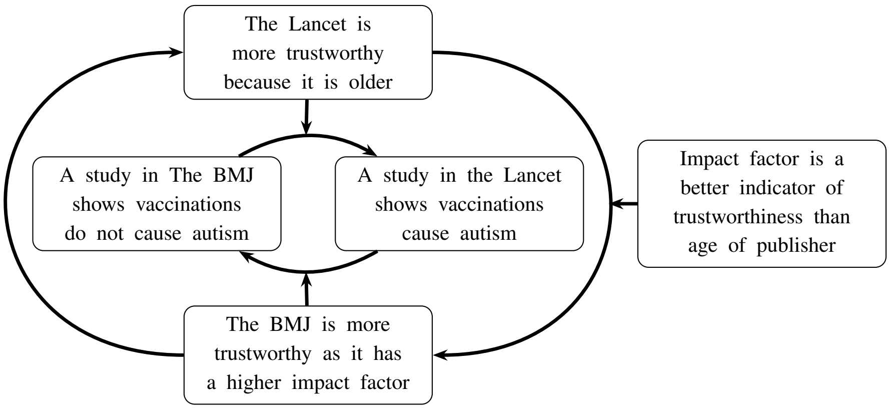

The Lancet is more trustworthy because it is older
A study in The BMJ shows vaccinations do not cause autism
A study in the Lancet shows vaccinations cause autism
Impact factor is a better indicator of trustworthiness than age of publisher
The BMJ is more trustworthy as it has a higher impact factor

Figure 4.1: An instantiated extended argumentation framework

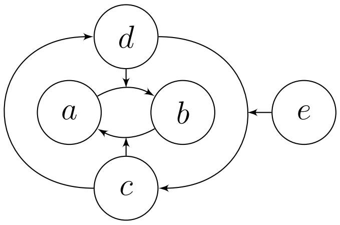


Figure 4.2: An abstract extended argumentation framework

EAF argumentation semantics are defined equivalently as for DAFs, with the following adjustments [76].

Definition 7. *Let $\langle A, R, D\rangle$ be an EAF and $S \subseteq A$.*

- *a **defeats**$_S$ b (also written as $a \rightarrow^S b$) iff $(a, b) \in R$ and $\nexists c \in S$ s.t. $(c, (a, b)) \in D$.*
- *S is **conflict-free** iff $\forall a, b, \in S$: if $(a, b) \in R$ then $(b, a) \notin R$ or $\exists c \in S$ s.t. $(c, (a, b)) \in D$.*
- *$R_S = \{x_1 \rightarrow^S y_1, \ldots, x_n \rightarrow^S y_n\}$ is a **reinstatement set** for $c \rightarrow^S b$ iff: (i) $c \rightarrow^S b \in R_S$; (ii) $\forall i \in \{1, \ldots, n\}$: $x_i \in S$, and (iii) $\forall x \in R_s$, $\forall y'$ s.t. $(y', (s, y)) \in D$: $\exists x' \rightarrow^S y' \in R_S$.*
- *$a \in A$ is **acceptable** w.r.t. S iff $\forall b$ s.t. $b \rightarrow^S a$: $\exists c \in S$ s.t. $c \rightarrow^S b$ and there is a reinstatement set for $c \rightarrow^S b$.*

Example 10. *Consider the EAF in Figure 4.2. The set of arguments acceptable under the grounded semantics is a, c, e, which is also the only preferred extension. Since e is not attacked and defeats the attack (d, c), c is not defeated by d.*

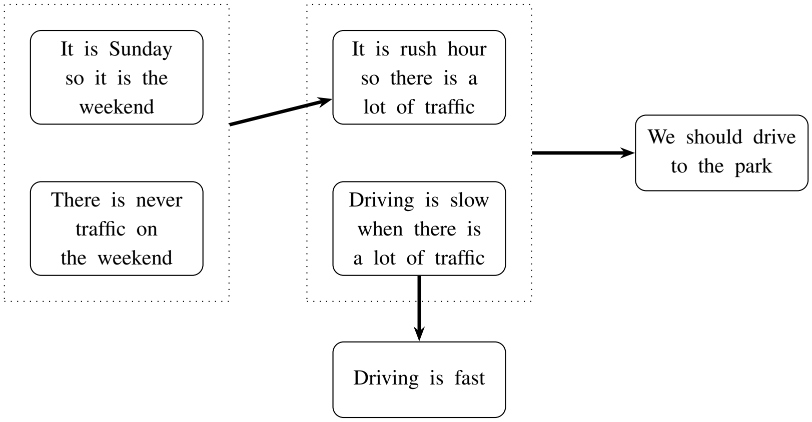

It is Sunday so it is the weekend
There is never traffic on the weekend
It is rush hour so there is a lot of traffic
Driving is slow when there is a lot of traffic
We should drive to the park
Driving is fast

Figure 4.3: An instantiated collective-attack framework.

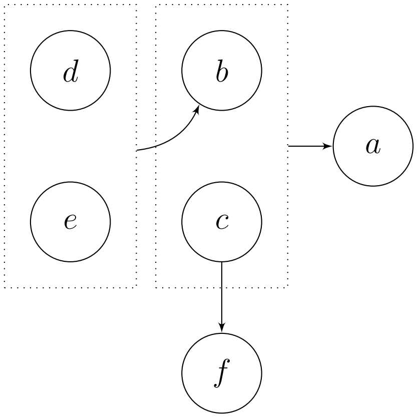


Figure 4.4: An abstract collective-attack framework.

4.2.2 Collective-attack frameworks

CAFs generalise Dung-style frameworks by permitting sets of arguments that attack an argument [83]. They can allow for a more intuitive representation of common-sense reasoning and human dialogues and have been shown to be useful in practical applications of argumentation [84]. They are particularly suited to capturing support over sub-arguments (as demonstrated in Figure 4.3), as well as accrual (allowing arguments to accumulate in an attack on an argument, in situations where there are not strong enough to do so individually). See Figure 4.4, in which there are two collective attacks: the set of arguments $\{b, c\}$ attacks the argument a, and $\{d, e\}$ attacks b.

Definition 8. *A* ***collective-attack framework (CAF)*** *is a pair $\langle A, R\rangle$ s.t. A is a finite set of arguments, and $R \subseteq (2^A \backslash \{\emptyset\}) \times A$ is a set of attacks where $(X, y) \in R$ is an attack from the set of arguments X to the argument y.*

Similarly to EAFs, CAF argumentation semantics are defined equivalently as for DAFs but with the following adjustments [83].

Definition 9. *Let $\langle A, R \rangle$ be a CAF and $S \subseteq A$.*

- *S is* **conflict-free** *iff $\nexists a \in S$ s.t. $\exists S' \subseteq S$ s.t. $(S', a) \in R$.*
- *$a \in A$ is* **acceptable** *w.r.t. S iff $\forall B \subseteq A$ s.t. $(B, a) \in R$: $\exists b \in B$, $\exists S' \subseteq S$ s.t. $(S', b) \in R$.*

Example 11. *Consider the CAF in Figure 4.4. The set of arguments acceptable under the grounded semantics is d, e, c, a, which is also the only preferred extension. b is not acceptable, because there is a collective attack $((d, e), b)$, and d and e are both acceptable since they are not attacked. a is acceptable because although there is a collective attack $((b, c), a)$, b is not acceptable, and so the collective attack is not effective.*

4.3 Classes of frameworks

Recall the previous definitions for argumentation frameworks: the principal argumentation frameworks that we refer to as Dung-style frameworks in this chapter (DAFs, Definition 1), extended argumentation frameworks (EAFs, Definition 6), and collective-attack frameworks (CAFs, Definition 8). In the following sections, we detail the experiments and results obtained from investigating the properties of these generalised frameworks. In our experiments, we randomly generate instances of each of these generalised frameworks, and so we must make decisions on their general properties (such as attack density), as well as the properties specific to the extended types (in the case of EAF, how many levels of preferences as well as the distribution of arguments across those levels; and in the case of collective-attack frameworks, how large a set of arguments can attack an argument). In this section, we present eight classes of frameworks (four for DAFs, two for EAFs, and two for CAFs) that we investigate in our experiments.

4.3.1 DAF attack density

Attack density of a DAF is the ratio of attack relations to the number of arguments. A framework with many attacks with respect to the number of arguments is *dense*, while a framework with fewer attacks is *sparse*.

Definition 10. *An* ***n-sparse DAF (n-DAF)*** *is a DAF $\langle A, R \rangle$ s.t. $n = \frac{|A|}{|R|}$, where $n \in [0, 1]$.*

We investigate 0.25-DAFs, 0.5-DAFs and 0.75-DAFs. Note that as n increases, the framework becomes more sparse. Note also that the number of attacks in the framework is linearly related to the number of arguments in the frameworks. We found in initial testing that if the number of attacks is tied instead to the number of possible attacks in the graph then small changes in sparseness value produce very sharp changes in the characteristics of that structural class of DAF; linearly relating the number of attacks to arguments allows us to explore this relationship more finely.

This is not an orthodox approach to measuring sparsity, which would relate density to the number of possible attacks instead: e.g., $o = \frac{|A|^2}{|R|}$, where o is the orthodox measure of sparsity. The measure of sparsity defined in Definition 10 can be connected to the orthodox measure of sparsity as follows: $o = n * |A|$. Our measure of sparsity is scale invariant because if you double the number of arguments in the DAF then the measure of sparsity doubles as well: a framework with 5 arguments and 20 attacks is a 0.25-sparse DAF; a framework with 10 arguments and 20 attacks is a 0.5-sparse DAF.

We also consider a class of DAFs that correspond to *minimum-spanning trees* (mst-DAFs), which are fully connected DAF in which the number of attacks is linearly related to the number of arguments ($|R| = |A| - 1$). In order to formally define a mst-DAF, we first define undirected walks in argumentation frameworks, and then define weakly-connected argumentation frameworks.

An *undirected walk* between two arguments in an argumentation framework is a list of arguments such that, between each argument in the list, there is an attack from one to the other or vice versa.

Definition 11. *An **undirected walk** in an argumentation framework $AF = \langle A, R\rangle$ between an argument $a_0 \in A$ and an argument $a_n \in A$ is a list of arguments $[a_0, a_1, ..., a_n]$ such that $\forall i \in \{0, 1, ..., n-1\}$, $(a_i, a_{i+1}) \in R$ or $(a_{i+1}, a_i) \in R$.*

Example 12. *Consider the argumentation framework in Figure 4.5a. There is an undirected walk between b and e, $[b, c, d, e]$.*

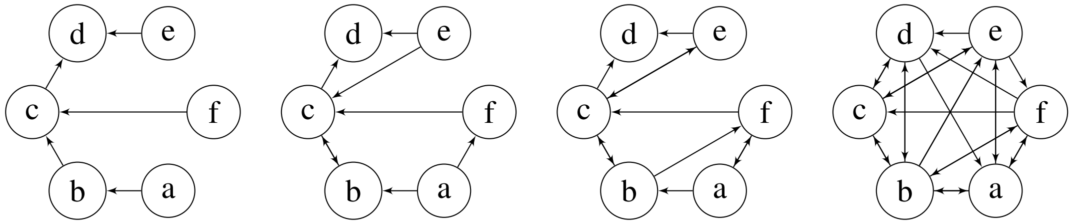
An argumentation framework is a *weakly-connected argumentation framework* if there is an undirected path from every argument in the framework to every other argument.

Definition 12. *An argumentation framework $AF = \langle A, R\rangle$ is a* weakly-connected argumentation framework *iff $\forall a, b \in A$ there is an undirected walk between a and b in AF.*

A mst-DAF is a weakly-connected argumentation framework such that if any attack is removed from the framework it would no longer be a weakly-connected argumentation framework. An example mst-DAF is shown in Figure 4.5a.

Definition 13. *An argumentation framework $AF = \langle A, R\rangle$ is a* **minimum-spanning tree DAF (mst-DAF)** *iff:*

- *AF is weakly-connected, and*
- *$\forall r \in R, AF_s = \langle A, R_s\rangle$ is not weakly-connected, where $R_s = R - \{r\}$.*



(a) mst-DAF (b) 0.75-DAF (c) 0.5-DAF (d) 0.25-DAF

Figure 4.5: DAFs of varying densities.

4.3.2 Distributed HEAFs

We especially consider here *hierarchical EAFs* (HEAFs), a particularly interesting class of EAFs that can be used to formalise practical reasoning [76]. HEAFs restrict the structure of EAFs, such that the framework is stratified into partitions. The intuition is that an argument can either attack arguments in its partition, or attack attack relations in the partition directly below its own.

Definition 14. *An EAF $\langle A, R, D\rangle$ is a* ***hierarchical extended argumentation framework (HEAF)*** *iff there exists a partition $P = [\langle\langle A_1, R_1\rangle, D_1\rangle, ..., \langle\langle A_j, R_j\rangle, D_j\rangle, ...]$ such that both:*

- *$A = \cup_{i=1}^{\infty} A_i, R = \cup_{i=1}^{\infty} R_i, D = \cup_{i=1}^{\infty} D_i$, and for $i = 1, ..., \infty$, $\langle A_i, R_i\rangle$ is a DAF, and*
- *if $(z, (x, y)) \in D_i$ then $(x, y) \in R_i, z \in A_{i+1}$.*

We refer to an argument a as being in a lower *partition than an argument b if $a \in A_p$, $b \in A_q$, and $p < q$.*

The arguments in Figure 4.2 can be partitioned into 4 levels: $\{a, b\}$, $\{c, d\}$, $\{e, f\}$, and $\{g, h\}$, where $\{a, b\}$ is the lowest partition and $\{g, h\}$ is the highest.

In some domains, particularly human dialogues, it seems reasonable to assume that the number of arguments will be higher than the number of preferences over those arguments, which will be higher than the number of preferences over preferences, *etc*. We consider two different distributions of the proportion of arguments that appear in the different HEAF partitions: *normally-distributed HEAFs* (nHEAFs) and *evenly-distributed HEAFs* (eHEAFs). In nHEAFs, arguments are distributed across the partitions with more arguments in the lower partitions compared to the higher partitions. Whereas, in eHEAFs, each partition has the same number of arguments.

For nEAFs, we use the binomial coefficient to approximate the normal distribution (continuous) over a finite number of partitions (discrete), and thus the proportions with which to assign arguments to each partition. We use the number of partitions relative to the number of arguments in the graph that allows for the best fit with the normal distribution (computed with Sturges' formula [103]). The choice of normal distribution provides the desired trend of decreasing proportions, and is somewhat common in natural domains [41].

Definition 15. *The discrete normal distribution over l partitions is given by the formula* $\mathsf{norm\_dist}(l) = [d_0, d_1, ..., d_{l-1}]$ *such that:*

- $n = 2l - 1$, *and*
- $d_k = \frac{n!}{k!(n-k)!}$.

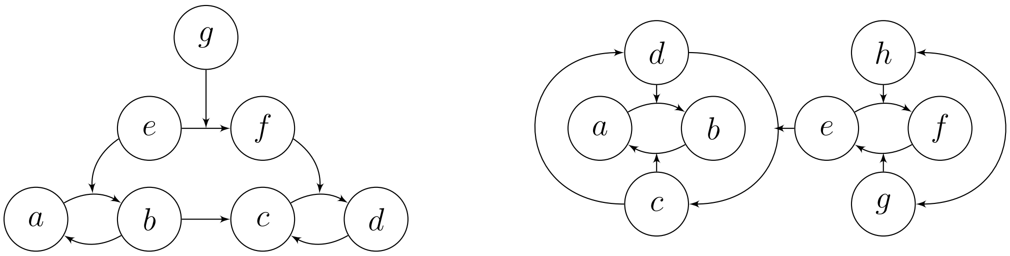
The proportional weights of the partitions are thus given by the formula $\mathsf{norm\_prop}(l) = [p_0, p_1, ..., p_{l-1}]$ *such that* $p_i = 2(d_i) \div 2^n$.

We can then use this definition of a normal distribution over partitions to define normally-distributed HEAFs. An example nHEAF is shown in Figure 4.6a.

Definition 16. *A* **normally-distributed HEAF (nEAF)** *is a HEAF* $\langle A, R, D \rangle$ *with a partition* $P = [\langle\langle A_1, R_1\rangle, D_1\rangle, ..., \langle\langle A_m, R_m\rangle, D_m\rangle]$ *such that:*

- $A = \cup_{i=1}^{m} A_i, R = \cup_{i=1}^{m} R_i, D = \cup_{i=1}^{m} D_i$, *and for* $i = 1, ..., m$, $\langle A_i, R_i \rangle$ *is a DAF,*
- *if* $(z, \langle x, y\rangle) \in D_i$ *then* $(x, y) \in R_i, z \in A_{i+1}$,
- $m = \lfloor \log_2 |A| \rfloor + 1$ *(Sturges' formula), and*
- $|A_j| = \lfloor (p_{l-j} \times |A|) + 1 \rfloor$ *where* $\mathsf{norm\_prop}(m) = [p_0, p_1, ..., p_{l-1}]$.

We also consider *evenly-distributed HEAFs* (eEAFs), in which each level of the partition has an equal number of arguments. An example eHEAF is shown in Figure 4.6b. We consider eEAFs to be an interesting corner-case to investigate. Again, we



(a) Normally-distributed EAF (nEAF)

(b) Evenly-distributed EAF (eEAF)

Figure 4.6: The different distributions of HEAF.

use Sturges' formula to compute an appropriate number of partitions for the number of argument.

Definition 17. *An **evenly-distributed HEAF (eEAF)** is a HEAF $\langle A, R, D\rangle$ with a partition $P = [\langle\langle A_1, R_1\rangle, D_1\rangle, ..., \langle\langle A_m, R_m\rangle, D_m\rangle]$ such that:*

- $A = \cup_{i=1}^{m} A_i, R = \cup_{i=1}^{m} R_i, D = \cup_{i=1}^{m} D_i$*, and for* $i = 1, ..., m$*,* $\langle A_i, R_i\rangle$ *is a DAF.*
- *If* $(z, (x, y)) \in D_i$ *then* $(x, y) \in R_i, z \in A_{i+1}$*,*
- $m = \lfloor \log_2 |A| \rfloor + 1$*, and*
- *For* $i = 0, ..., m$*,* $|A_i| = \lceil (|A| \div m \pm 1) \rceil$.

4.3.3 Capped CAFs

We consider two structures of CAF: those in which the size of any collective-attack set is no greater than (*capped at*) 3 and CAFs in which there is no restriction on the size of collective-attacks sets. We refer to capped frameworks as *cCAFS*, and those which are uncapped as *uCAFs*.

Definition 18. *A **capped collective-attack framework (cCAF)** is a CAF* $\langle A, R\rangle$ *s.t.* $\forall (S, a) \in R : |S| \leq 3$.

Note, in the rest of this chapter, to emphasise the distinction with capped collective-attack frameworks, we refer to collective-attack frameworks as **uncapped collective-attack frameworks, (uCAFs)**.

4.4 Investigating characteristics

In this section we describe the experiments we ran on the classes specified in the previous section (0.25-DAF, 0.5-DAF, 0.75-DAF, mst-DAF, eEAF, nEAF, cCAF, uCAF).

We detail the three properties that we investigate (extension size, proportion of subsets in which a topic is acceptable, and dynamic argumentation) and their relevance to argument-based systems, and discuss the results for each class.

Our experiments were implemented in Java, partly using the Tweety library [108]. Experiments were run on an Intel i5 3.20GHz CPU, with 4GB RAM.

4.4.1 Framework generation for experimental setup

Before presenting the experiments, we first clarify the details of the experimental setup, specifically the generation process of the argumentation framework structures.

- **DAFs**

  The mst-DAFs are generated as random, directed, unweighted, spanning trees [112]. To generate denser DAFs, an mst-DAF is generated first, and then attacks are added uniformly at random from the remaining set of possible attacks, until the desired density is met.

- **EAFs**

  EAFs are generated as follows. First, the arguments are distributed across the defined number of partitions, according to the defined distribution (normal or uniform). Second, for each partition's arguments, a 0.75-DAF is generated as above. Third, for each argument in a partition $p > 1$, an attack is generated to an attack in partition $p - 1$ that is selected uniformly at random.

- **CAFs**

  Both uCAFs and cCAFs are first generated as mst-DAFs as above. Then, the number of collective attacks to be added, ca is determined by a uniformly random number in the range $[1, n]$, where n is the number of arguments in the framework. The ca sets of arguments for each collective attack are then generated.

  For uCAFs, these sets can be of a maximum size $n - 1$, so for each set to be generated, a uniformly random number in the range $[2, n - 1]$ is used as its size. For cCAFs, the sets can be of a maximum size of 3, so the sets can be of size 2 or 3 with equal probability. A random subset of arguments of the established size is then selected. Note, that the subsets of arguments can overlap, or even be identical.

Finally, for each generated subset of arguments, an attack is generated from the subset to a randomly selected argument that is in the framework but not in the argument subset.

4.4.2 Size of extension

The First International Competition on Computational Models of Argumentation [28], in which argument solvers attempt to complete a set of tasks related to computational argumentation as efficiently as possible (such as computing an extension, or determining whether a particular argument is acceptable) used three different benchmark sets of frameworks to evaluate the solvers. Two of the benchmark sets were based on the size of the extensions of the frameworks: frameworks with large grounded extensions and frameworks with a large number of stable extensions. The results showed that most solvers were slower when tasked with frameworks with a large number of stable extensions compared to those frameworks with a large grounded extension. This indicates that the size of the extensions of a framework is an important consideration when employing an argument solver for certain tasks. It is therefore of interest which structures of framework have large extensions.

We investigate how the average size of both the grounded and preferred sceptical extensions differs between our chosen framework classes. We generate 1,000 instances of each framework class for sizes of 12, 19, 24, 31, and 36 arguments and measure the size of the extensions. The irregular intervals between the number of arguments are used so that both even and odd sizes of argument frameworks are investigated.

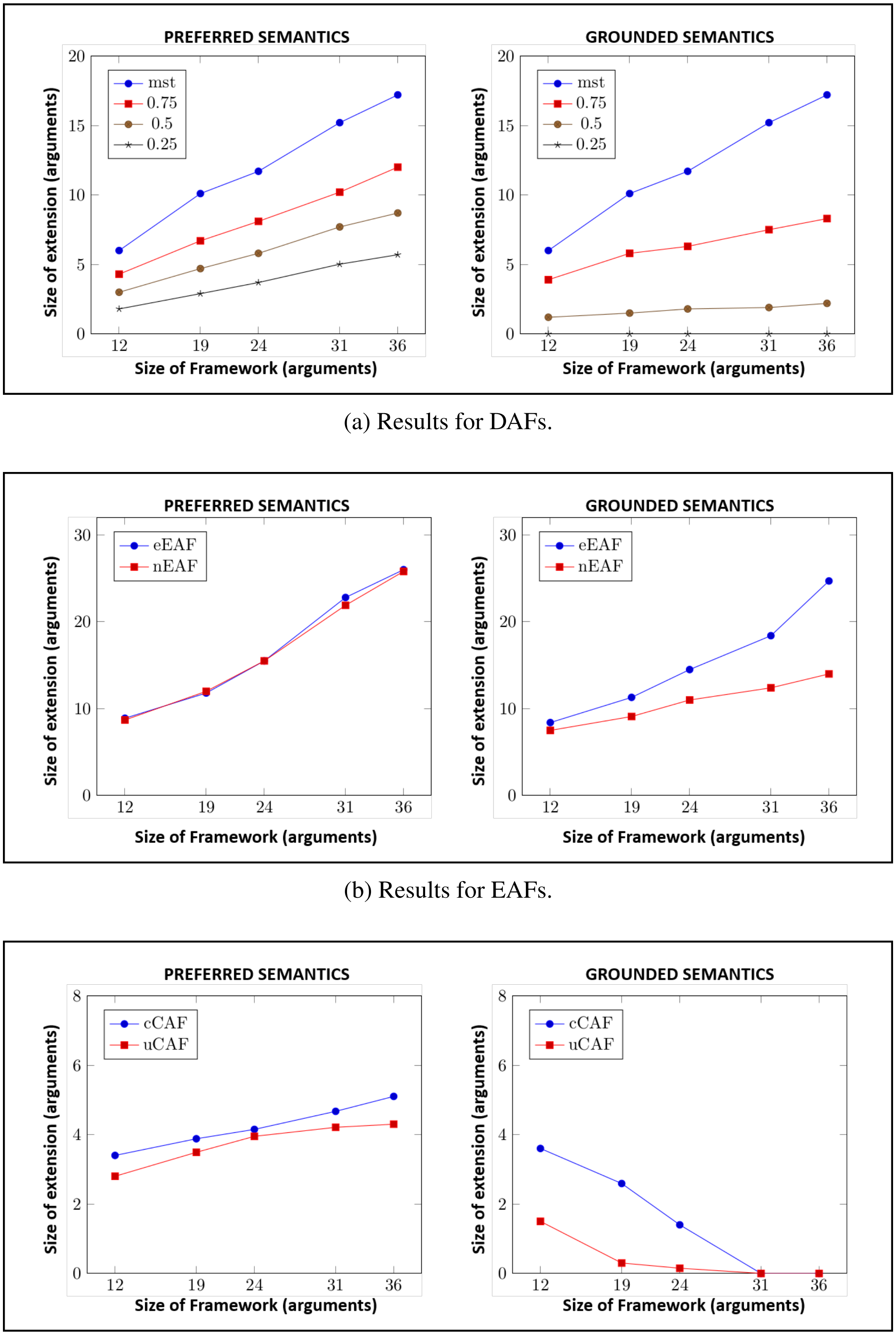
In Figure 4.7, we show the size of the grounded and preferred sceptical extensions for each class of framework. We use a series of line graphs to visualise the results, to allow comparison of the framework classes across the parameter space of framework size. For DAFs, we observe a trend for both semantics that the more dense the DAF, the smaller the size of the extension. We also observe that the larger the framework, the larger the extension will be on average, even as a proportion of total arguments.

We find that eEAFs are more likely to have a larger grounded extension than nEAFs, but have similar sized preferred sceptical extensions. We reason that in EAFs, the more preference arguments there are in a framework, the more likely attack relations in the partition *below* will be defeated. This effectively lowers the attack density in lower partitions. So in the frameworks with a higher proportion of preference relations (eEAFs) there will be a lower overall attack density. As we observe in DAFs, the lower the attack density of the framework, the larger the extension — this is reflected in the results for the grounded extension.

Interestingly, CAF frameworks reverse the trend when using the grounded semantics: the larger a uCAF/cCAF framework, the smaller the average grounded extension is. This surprising result can be explained by the intuition that as you increase the number of arguments in a CAF, this increases the *proportion* of collective attacks, and thus the more arguments that are part of a collective attack relation, leading to a higher number of attack cycles (the more arguments in a set S that collectively attack an argument a, the higher the chance that a will attack at least one argument in S, causing a cycle), and the more attack cycles in a framework the smaller the grounded extension is likely to be. This is supported by the fact that we observe that uCAFs have a smaller grounded extension on average than cCAFs, which, we conclude, is due to more arguments being part of a collective attack relation in uCAFs (as there is no cap on the number of arguments in the attack relation). When using the preferred semantics, cycles are less of a factor in the size of the extension (since arguments in a cycle may still be justified), and so we observe that the size of the preferred sceptical extension increases as the size of the framework increases.

To further investigate the relationship between the size of extensions in CAFs and the number of cycles, we examine the effect of the proportion of the number of even-length cycles to the number of odd-length cycles. We use CAFs of size 12 from the previous experiments (the grounded extensions of larger CAFs is too small). The range of proportions plotted are from one even-length cycle for every odd-length cycle, up to four even-length cycles for every odd-length cycle, in steps of 0.5. Proportions beyond this range did not exist within the population in large enough numbers for sufficient analysis.

The results of the further investigations are shown in the plots in Figure 4.8, where a framework with a proportion of p even to odd cycles is a framework is p times as many even cycles as odd cycles. The Pearson correlation coefficients are shown in Table 4.1. We observe statistically significant, strong, positive correlations for all plots. This demonstrates that there is a strong relationship between the ratio of even to odd cycles in a CAF and the size of the grounded and preferred extensions. However, the apparent effect of even/odd-length cycle proportions on extension size is larger under the grounded semantics than the preferred semantics. Under the preferred semantics, the range in the average size of the extension is less.



(a) Results for DAFs.

(b) Results for EAFs.

(c) Results for CAFs.

Figure 4.7: Graphs showing the size of extensions in framework structures.

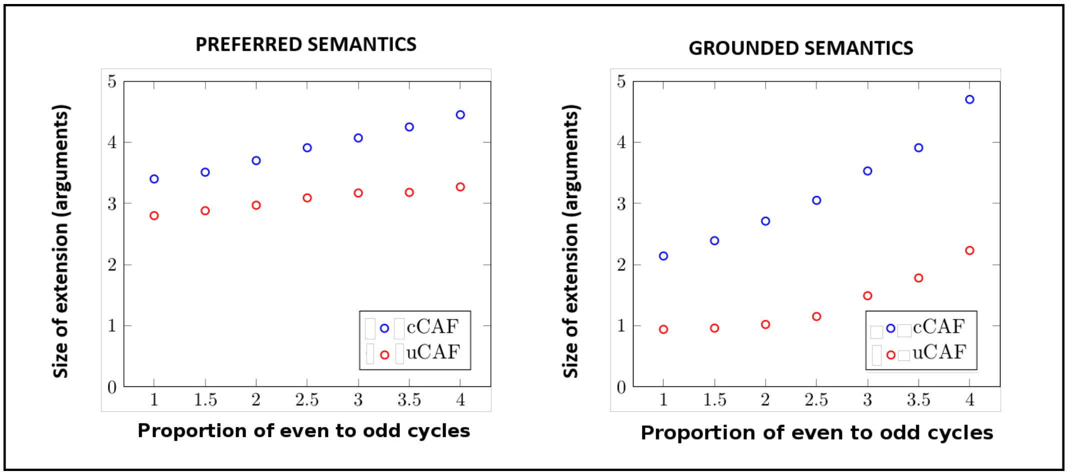


Figure 4.8: Scatterplots to show the relationship between the extension size and the proportion of even to odd cycles in CAFs.

Semantics	**CAF**	R **value**
Preferred	cCAF	0.988
	uCAF	0.971
Grounded	cCAF	0.985
	uCAF	0.939

Table 4.1: Pearson correlation coefficients (R) for even/odd cycle proportions in CAFs and size of extension (all statistically significant with $p < 0.05$).

4.4.3 Subsets in which topic is acceptable

A topic argument t of a framework will be acceptable in some subgraphs of the framework, but not in others. A topic argument t will be acceptable in at most 50% of the subsets, since it will not exist in half of the subsets of the power set (an argument is deemed unacceptable in a framework it is not a part of). We refer to the proportion of subsets in which the topic argument is acceptable as *SA*. This property of frameworks has been found to be an important factor in determining a strategy in persuasion dialogues [15]: in a domain where SA is lower it is more difficult to persuade an agent that t is acceptable.

We investigate whether average SA differs between the selected framework classes. Our implementation is naive, exhaustively checking whether the topic is acceptable in every set in the power set. The time for these experiments is very high due to the exponential growth in the number of sets in the power set. To feasibly compute the results we use the grounded semantics (which are faster to compute), and limit the framework size to 12 arguments. We generate 1,000 instances of each framework class with 12 arguments, each time randomly selecting a topic argument.

The results are presented in Figure 4.9. We use a box plot to visualise the data to allow for comparisons of the different framework structure populations across the continuous dependant variable of SA. In the different classes of DAF, we observe a clear trend that the more dense a framework class, the lower SA is for that class. This follows the trend observed in Figure 4.7, where the more dense a DAF, the smaller the grounded extension. Similarly, uCAF frameworks typically have a smaller grounded extension than cCAF frameworks, and this trend is repeated for SA. For nEAF and eEAF frameworks of 12 arguments, there is little difference between the size of grounded extensions, and this trend is again shown for SA, where eEAF and nEAF do not appear to have different SA. When using the grounded semantics it appears that the size of the extension and SA are fundamentally linked.

We use a series of t-tests to establish whether the differences observed in the framework classes are statistically significant. Each class of framework is compared with every other class of framework, giving 56 separate t-tests. The assumptions of the t-tests are met as follows.

- Each class is approximately normally-distributed (confirmed by using the Kolmogorov–Smirnov test).
- The sample sizes of each class is equal (1,000 instances each).
- The classes are independent (instances being generated independently).

We find that the classes have significantly different SA (apart from nEAF and eEAF which are distinct from other classes but not from each other) and thus that each class is a distinct population ($p < 0.05$ for each class); this implies that the framework class is a significant factor in determining SA. The largest difference between two classes is between *mst-DAF* and *nEAF* (36.06 percentage points between means).

4.4.4 Dynamic argumentation

Argumentation is an inherently dynamic process, with arguments and attack relations changing as new knowledge becomes available: for example, an individual agent exploring their environment to gain novel information, or a group of agents communicating new arguments to one another in a dialogue. The dynamic nature of argumentation can potentially be exploited for computational efficiency [67] as well as for strategic advantage [27].

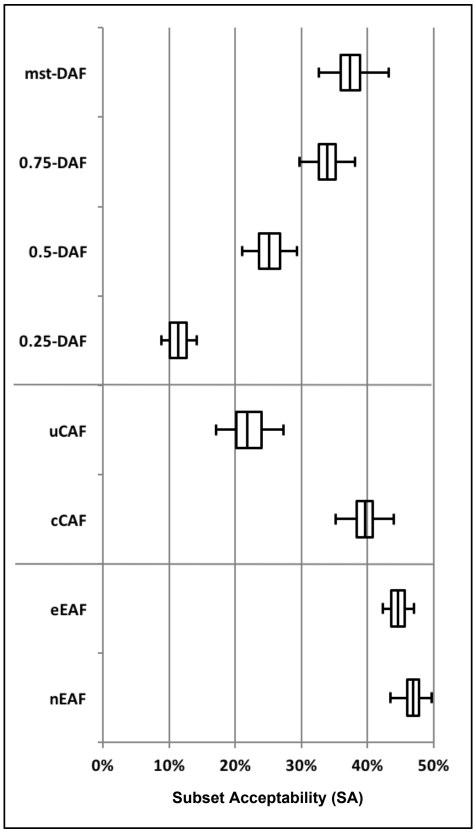


Figure 4.9: Graphs showing subset acceptability in framework structures.

Amgoud and Vesic [1] consider whether the addition of a new argument to a framework changes the acceptability of a specific argument (termed the *topic argument*). If the addition of a new argument does not cause a change in the topic argument's acceptability we say the framework is *resistant*, otherwise is it *susceptible*. We investigate whether there is a difference in the resistance of the different framework classes. We generate at least 1,000 instances of each framework class with 12, 24, and 36 arguments, selecting a topic argument at random, and testing whether the acceptability of the topic changes with the removal of a random argument under the preferred sceptical semantics.

Figure 4.10 shows the results from experiments on dynamic argumentation, displaying the resistance for each framework class. We use a stacked bar chart to visualise the data to allow for comparisons of the different framework structure populations across the binary variable of resistance/susceptibility. For all classes we observe that the more arguments in the framework, the less likely it is that adding a new argument will have an effect on the acceptability of the topic argument. The intuition behind this result is as follows: the more arguments in a framework, the more likely it is that the argument is topographically further away from the topic, and therefore the less likely the added argument will change the acceptability of the topic (this relationship is explored further in Chapter 5, where it is used as a heuristic to inform an argument-dialogue strategy).

In a cCAF, a new argument can alter the acceptability of arguments both through introducing new argument-argument attacks as well as new collective attacks. This is also true in uCAFs, though they have a greater chance of introducing collective attacks: since the size of a collective attack is uncapped, each argument is in more collective attack relations on average. Thus, when we add a new argument to a uCAF it is likely to result in more changes in the acceptability of arguments, and this is a reason why we observe that cCAFs are more resistant.

We see that eEAFs are more resistant than nEAFs, indicating that the higher the proportion of preference arguments to arguments, the more resistant the EAF will be. This is because an argument cannot alter the acceptability of an argument in a partition higher than its own partition since all attack relations are either to arguments in the same partition or to arguments in the partition directly below. Therefore, if the topic argument is in a higher partition than the added argument, the framework is guaranteed to be resistant. In eEAF it is more likely that the topic will be in a higher partition (since it is randonly selected and there are more arguments in higher partitions than in a nEAF), and thus the less likely it is that an added argument will have any effect on the topic's acceptability.

4.5 Case studies

In this section we present two case study frameworks, obtained from argumentation tools deployed on real-world data. The case studies provide motivation for the relevance of the classes of framework structure we investigate (showing the results of our experiments map to results of the experiments run on real-world frameworks), and also allows us to demonstrate how our results can inform argument technologies.

The tests for resistance and subset acceptability are averages over varying both the framework and the argument topic. For our case studies we are unable to vary the framework, since we have only a single framework for each case study; instead, we vary the argument topic only.

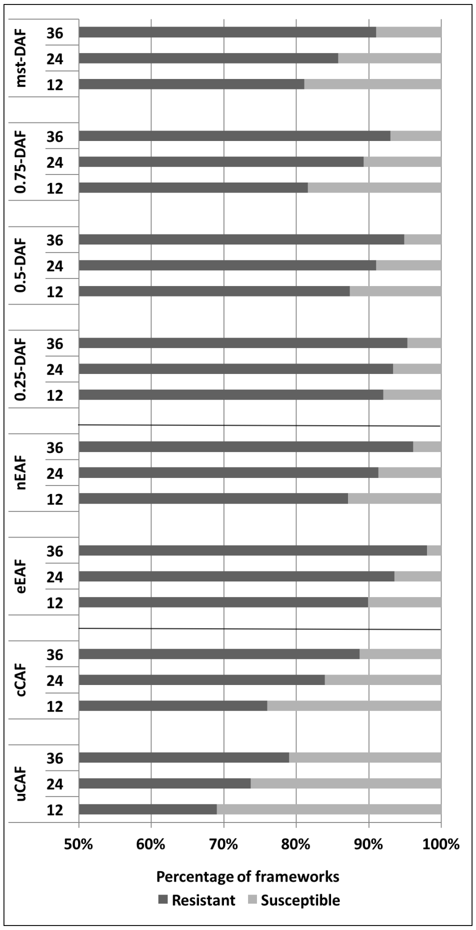


Figure 4.10: Graphs showing resistance of framework structures.

4.5.1 Trial aggregation

As evidence-based decision-making becomes increasingly important, clinical trials can provide an important source of information to inform healthcare professionals. Hunter and Williams propose an argument-based approach for aggregating the positive and negative effects of potential treatments, by representing each study as an argument in a Dung-style framework [60]. The recommended treatment options obtained from this approach have been shown to align with published clinical guidelines, demonstrating the usefulness of the approach. The approach performs a type of meta-analysis on a range of clinical literature, producing a Dung-style argumentation framework (very sparse; almost a mst-DAF in structure), on which reasoning about possible treatment options is done. We use such a framework as our first case study.

The results of our experiments on this framework are shown in Table 4.2. For each experiment on this framework, we find the are similar to the results obtained from mst-DAF presented earlier in this chapter, with the size of extensions, SA, and resistance being within the expected ranges of mst-DAFs. This evidences the relevance of the structures we investigate. We examine the resistance of this particular framework, to demonstrate how our results may be used to inform specific domains.

The resistance of the framework from the trial aggregation case study is exceptionally high (*97.2%*). This indicates that new arguments added in the future, in this case by the addition of new clinical studies, are unlikely to change the acceptability of other arguments in the framework. This implies that new studies are unlikely to have an affect on the recommended treatment, meaning there can be confidence in the current recommendation. If a framework produced by the trial aggregation approach had a low resistance, new studies would be likely to change the recommended treatment, and this would imply that the recommendation is not yet reliable.

4.5.2 Statistical model selection

Clinicians without statistical training often need support to correctly analyse and reason about their data. Sassoon *et al.* propose a tool that uses argumentation to aid in the process of deciding which statistical model is most suited to a users' data and preferences [98]. The requirements and preferences of the user, as well as preferences from their specific context domain, are captured in an EAF, which can then inform the user of the most suitable model to use. We use a framework produced by using this tool with real-world data from a study involving clinicians (originally presented in [98]) as our second case study. The framework is an example of an eEAF, being an EAF with the same number of arguments at each level of the hierarchy.

The results of our experiments on this framework are shown in Table 4.2. For this case study we find that for each experiment on the framework, the results again correlate with the corresponding generalised framework class, this time eEAF. Perhaps the most interesting result from this case study is the high SA of the framework (*46.5%*). Empirical investigations have demonstrated that the higher SA, the easier it is for a persuader to convince a persuade of a particular argument through dialogue [15, 78], and so we would thus expect the persuasion of a user to use a particular statistical model to be successful in the majority of cases.

Table 4.2: Results for case study frameworks*

Case Study	Args	Gr	Pr	SA	Res
Trial aggregation	34	9	9	41.9	97.2
Model selection	13	7	7	46.5	89.1

**Args*: number of arguments in the framework. *Gr* and *Pr*: size of the grounded and preferred sceptical semantics respectively. *SA*: percentage of subsets that determine topic to be acceptable. *Res* is the resistance of the framework.

4.6 Conclusions and Discussion

In this chapter we have presented the results from measuring the properties of different generalised frameworks (Dung frameworks, collective-attack frameworks, and hierarchical extended argumentation frameworks), and different structures of those frameworks (relating to attack density, collective attack size, and preference distribution). We have investigated properties of the frameworks that are especially pertinent to the performance of argument-based systems: the size of extensions is a factor in the efficiency of solvers, the resistance and SA of a framework are important properties for argument-based dialogues as well as for strategic argumentation [15, 1], and dynamic argumentation may have applications in the performance of argument solvers [67].

We have shown that the class of framework, and indeed the structure of the chosen framework, has a significant effect on all of the investigated properties. Identifying the characteristics of frameworks derived from argument technologies (such as different generalised frameworks, as explored in this chapter) is important when considering how to evaluate an argument-based system. Selecting relevant framework structures leads to a grounded evaluation of the system, ensuring the system's performance is measured on a realistic domain. Furthermore, it allows systems to be optimised for specific domains in which a general system may be less efficient. For example, solvers can be developed to be faster for particular classes of framework, or a dialogue strategy can be effective for particular knowledge domains.

A complementary approach to the one taken in this chapter is to evaluate systems based on examples of human argumentation (such as recent work by Rosenfield and Kraus [95]). Argument mining offers the possibility of obtaining large datasets of frameworks from real-world human-based argumentation, and which can be applied to a vast array of domains (*e.g.* from biomedical research literature [47]), providing a range of framework structures related to human-reasoning. However, current corpora of human arguments are limited in size and availability to allow for large scale empirical investigations (especially for EAF and CAF style frameworks). A direction for future work is to investigate frameworks obtained from human reasoning, possibly through techniques such as argument mining, and to investigate their characteristics.

Chapter 5

A heuristic strategy for persuasion dialogues

Argument-based dialogues are a useful mechanism for agent co-ordination, particularly in the domains of human-machine interaction and agreement technologies [77]. In this chapter, we focus on a simple type of persuasion dialogue (where one agent presents arguments to another with the aim of convincing it to accept some argument that is the topic of the dialogue) and consider the problem of how the persuader can determine which arguments to present during the dialogue, *i.e.,* what dialogue *strategy* it should employ.

The development of methods for generating agent dialogue strategies is an active area of research [107]. So far, work on this problem has shown that computing an optimal strategy for one-to-one persuasion dialogues is computationally expensive, and becomes intractable as the number of arguments in the dialogue domain increases. Black *et al.* [15] consider the a simple persuasion dialogue setting similar to the one that we focus on in this chapter, modelling it as a planning problem so that a planner can be used to generate an optimal strategy for the persuaded. The planning approach was later adapted by Black *et al.* to a richer model of argument dialogue [16]. Hadoux *et al.* [50] and Rienstra *et al.* [92] also each support richer models of argument dialogue, generating optimal strategies using Mixed Observability Markov Decision Problems (MOMDPs) and a variant of the minimax algorithm respectively. While all of these approaches [15, 50, 92] determine an optimal strategy for the persuader, none have been shown to scale to domains with more than 13 arguments.

The key contribution of this chapter is a heuristic strategy for persuasion that can easily scale to domains with 50 arguments (with computation time of less than 1 second). Although this heuristic strategy is not optimal, it gives a reasonable chance of successful persuasion and significantly outperforms a strategy that randomly selects arguments. Our heuristic strategy does not require the persuading agent to have any knowledge of the persuadee, relying only on arguments the persuader knows may exist in the domain. The heuristic uses a measure of topographical distance to the topic argument to estimate the likelihood that any argument would (if asserted) affect the persuadee's perception of the topic's acceptability.

We evaluate our strategy in a simple persuasion setting, where one agent, the *persuader*, asserts arguments with the aim of convincing the other agent, *the responder*, to accept the topic of the dialogue, while the responder replies truthfully at each dialogue step to indicate whether it finds the topic to be acceptable. Since our heuristic strategy only uses knowledge of the arguments that might exist in the domain, it can also be applied in more complex persuasion settings (*e.g.,* one with multiple participants, or one in which each agent asserts arguments with the aim of having its preferred argument accepted). As we discuss later in Section 5.7, we believe that the results we present regarding the performance of the heuristic strategy in the simple persuasion setting are indicative of the performance we might expect to see in more complex dialogue settings.

The development of strategies such as ours that scale to large numbers of arguments is particularly important if we are to support a full range of dialogues, such as those in which more than two agents are engaged in the communication [32]. Such non-trivial dialogue scenarios have increasing numbers of arguments as the number of participating agents increases (typically, each agent brings arguments unknown to others in the dialogue) so adapting current methods to compute optimal strategies for dialogues with more than two parties would likely be impractical. A real-world example of such a multi-party domain is Decide Madrid[1], an online forum in which citizens can participate in debates in order to make meaningful decisions about local government policy. Debates on this site have many interacting users, and commonly have in excess of 50 arguments.

This chapter is structured as follows. Section 2 provides the preliminary background on argumentation and argument dialogues, in particular the two-player simple persuasion dialogue we use as a test-bed for our strategy. Section 3 introduces the heuristic used to estimate the likelihood of each argument to persuade the responder, and in Sec-

[1] `decide.madrid.es`

tion 4 the strategy is formally defined. Section 5 details the experimental set-up, and Section 6 presents the results. Section 7 concludes with a discussion.

5.1 Argumentation and simple persuasion dialogues

Recall that, given an argument framework, we can determine which *extensions* (sets of arguments) are rational for an agent to consider acceptable. While different extensions are based on different intuitions, a desirable property for a set of acceptable arguments is often that of *admissibility*. An argument is admissible with respect to a set of arguments S if all of its attackers are attacked by some argument in S, and no argument in S attacks an argument in S. For the rest of this chapter, we consider an argument to be *acceptable* to an agent (w.r.t. an argumentation framework) if it is part of all maximal admissible sets. These criteria for acceptability are known as the *preferred sceptical semantics* (as in [33]).

Definition 19. *We define a function, $\pi(AF)$, to return the set of acceptable arguments under the preferred sceptical semantics of the given argumentation framework AF.*

To investigate the effectiveness of the heuristic strategy we apply it to a persuasion dialogue (adapted from [15]) that has two participating agents: a *persuader* and a *responder*. The persuader's goal is to convince the responder of the dialogue topic (an argument). The responder replies truthfully as to whether it finds the topic acceptable given its (private) beliefs and the arguments asserted by the persuader. Agents engage in a dialogue under an argument framework — the *global knowledge* (all possible arguments in the domain, and the attacks between them) — from which their own personal knowledge is a subset.

Definition 20. *A **simple persuasion dialogue scenario**, under global knowledge $AF_G = \langle A_G, R_G \rangle$, is a tuple $\langle AF_P, AF_R, t \rangle$, such that:*

- *$AF_P = \langle A_P, R_P \rangle$, where $A_P \subseteq A_G$ and $R_P = R_G \cap (A_P \times A_P)$, is the persuader's initial knowledge base,*
- *$AF_R = \langle A_R, R_R \rangle$, where $A_R \subseteq A_G$ and $R_R = R_G \cap (A_R \times A_R)$, is the responder's initial knowledge base, and*
- *$t \in A_P$, is the dialogue topic.*

During the dialogue, the persuader and responder take turns to make utterances to one another; the persuader may assert an argument or choose to terminate the dialogue, while the responder makes a *yes* or *no* move, indicating whether they find the topic acceptable. A *well-formed simple persuasion dialogue* is one in which the persuader only asserts arguments from their knowledge base, the responder replies truthfully by indicating whether they currently find the topic argument acceptable, and that terminates once either the responder is convinced or the persuader chooses to give up.

Definition 21. *A **well-formed simple persuasion dialogue** of a simple persuasion dialogue scenario $\langle AF_P, AF_R, t\rangle$ under global knowledge $\langle A_G, R_G\rangle$, is a sequence of moves $[M_0^P, M_0^R, ..., M_n^P, M_n^R]$, such that:*

- $\forall i$ *such that* $0 < i < n$, $M_i^P \in A_P$,
- $M_n^P \in A_P \cup \{\texttt{terminate}\}$,
- $\forall i$ *such that* $0 < i < n$, $M_i^R = \texttt{no}$ *and* $t \notin \pi(\langle A_R \cup \{M_0^P, ..., M_i^P\}, R_G\rangle)$,
- $M_n^R \in \{\texttt{yes}, \texttt{no}\}$, *and*
- $M_n^R = \texttt{yes}$ *iff* $t \in \pi(\langle A_H \cup \{M_0^P, ..., M_n^P\}, R_G\rangle)$.

*A dialogue is **terminated** iff either* $M_n^P = \texttt{terminate}$ *or* $M_n^R = \texttt{yes}$. *A terminated dialogue is said to be **successful** iff* $M_n^R = \texttt{yes}$*, and **unsuccessful** otherwise.*

Over the course of a well-formed simple persuasion dialogue, the responder has no strategic concerns, as it must reply honestly if it finds the topic acceptable. However, each turn of the persuader requires a decision as to whether an argument should be asserted, and if so, which arguments in its knowledge base should be asserted. Previous work [15] has applied automated planning techniques to find an optimal strategy for the persuader to apply in this simple dialogue setting, but this does not scale well beyond 8 domain arguments. In Section 5.3 we present a heuristic strategy, and show that this can easily scale to domains with up to 50 arguments. First, however, we give the intuition on which this heuristic strategy relies.

5.2 Evaluating the influence of arguments

We consider the *local* topological properties of argument graphs to estimate how beneficial an argument would be if asserted. The estimate is based on the intuition that arguments topologically closer to the topic are more likely to affect its acceptability. We estimate the likelihood that an argument affects the acceptability of the topic and determine whether the argument defends or attacks (perhaps indirectly) the topic. Note that argument acceptability not only depends on the attackers of the argument, but on the

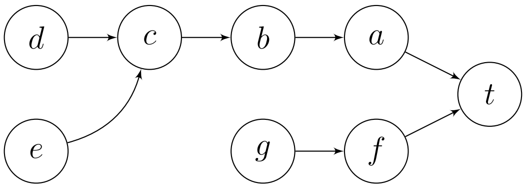


Figure 5.1: An example argumentation framework.

acceptability of the attackers. Thus, we are interested in *argument paths* terminating in the topic argument.

Definition 22. *An **argument path**, in an argument graph $AF = \langle A, R \rangle$ with topic t, is a list of arguments $p = [a_0, a_1, ..., a_k]$, such that:*

- *$a_0 = t$,*
- *$\forall i$ such that $1 \leq i < k$, $\langle a_{i+1}, a_i \rangle \in R$,*
- *$\forall i, j$ such that $0 \leq i, j \leq k$, $a_i = a_j$ iff $i = j$ (arguments are distinct).*

*The **depth** of an argument a in an argument path $p = [a_0, a_1, ..., a_i]$ is given by the function:* depth*(a, p) = x where $a = a_x$.*

Example 13. *Consider the example argumentation framework in Figure 5.1 with the topic being t. Valid argument paths include $[t, f, g]$, $[t, a, b]$, and $[t, a, b, c]$; sequences of arguments that are not argument paths include $[a, b, c]$ (the first argument is not the topic), and $[t, a, f]$ (there is no such path is in the argumentation framework).*

The distance of an argument from the topic argument provides an estimate of how likely it is that asserting the argument will affect the acceptability of the topic. The intuition behind this is as follows: for an argument to affect the topic through a particular argument path, all preceding arguments on that path must be present; furthermore, any arguments that precede the argument in question and support the topic cannot be defeated by an acceptable argument from another path. The more arguments that precede the argument on a particular path, the more chance that one of these conditions may not hold, thus the more likely it is that the argument will not affect the topic through that path.

Example 14. *Consider the example argumentation framework in Figure 5.1. The persuader wishes to convince the responder (whose arguments are unknown) that the topic t is acceptable. Consider that the persuader chooses to assert the argument g; in order for this to have a chance of changing the responder's perception of the acceptability of the topic, the responder must know f. Consider instead that the persuader chooses to* *assert the argument d (which is twice as far away from the topic as g); for this to have a chance of changing the responder's perception of the acceptability of t, not only must the responder know a, b and c, but it must also be that the responder cannot know e.*

To obtain an estimate of how likely each argument is to affect the acceptability of the topic, we must consider all argument paths in the argument graph that start with the topic. The importance of an argument on an argument path decreases to insignificantly small amounts as it gets further from the topic, so we consider only argument paths up to a specified depth.

Definition 23. *The* ***complete set of argument paths*** *with depth d of an argumentation framework AF and topic argument t, is a set of argument paths $C^d_{AF,t}$ where:*
$C^d_{AF,t} = \{[t, a_1, ...a_x] \mid [t, a_1, ...a_x]$ *is an argument path in* $AF,\ x \leq d,$ *and*
$\nexists[t, a_1, ...a_x, ..., a_y]$ *s.t.* $[t, a_1, ...a_x, ..., a_y]$ *is an argument path in* AF *and* $x < y \leq d\}$.

An argument at an even depth in a path will be a *supporting argument* of the topic, and its presence in an agent's knowledge *increases* the likelihood that it finds the topic acceptable (the argument is either the topic argument itself, or an argument that attacks an opposing argument). Similarly, an argument at an odd depth will be an *opposing argument*, and its presence *decreases* the likelihood that it finds the topic to be acceptable (the argument is an attacker of a supporting argument). With respect to a particular argument path, the magnitude of an argument's *value* is an estimation of the likelihood that the argument will affect the acceptability of the topic, and the sign indicates whether it is likely to make the topic acceptable (positive sign) or unacceptable (negative sign). Note that an argument can be both supporting and opposing of the topic in different argument paths of the same AF.

Definition 24. *The* ***value*** *of an argument a with depth $d = \mathsf{depth}(a, p)$ w.r.t. an argument path $p = [a_0, a_1, ..., a_i]$ is given by the function:*

$$\mathsf{value}(a, p) = \begin{cases} 0 & \text{if } a \notin \{a_0, ..., a_i\} \\ 1/2^d & \text{if } a \in \{a_0, ..., a_i\} \text{ and } d \mod 2 = 0 \\ -1/2^d & \text{if } a \in \{a_0, ..., a_i\} \text{ and } d \mod 2 = 1 \end{cases}$$

Example 15. *Consider the AF in Figure 5.1 with the topic being t. The value of c, with respect to the path $[t, a, b, c, d]$, is $-\frac{1}{2^3} = -\frac{1}{8}$. The value of d with respect to the same argument path is $\frac{1}{2^4} = \frac{1}{16}$, which is both smaller in magnitude than the value of c (as it is further from the topic) as well as positive (as it is defending the topic rather than attacking).*

To get an accurate estimation of whether the presence of an argument in an agent's knowledge base is likely to make them find the topic acceptable, and thus predict how beneficial it is for the persuader to assert that argument in a persuasion dialogue, the value of the argument in all argument paths needs to be considered. To determine the *estimated utility* of an argument, which represents the argument value with respect to the complete set of argument paths, we sum the values of that argument with respect to each argument path to the topic.

Definition 25. *The* ***estimated utility*** *of an argument a in an argumentation framework AF with topic t to a depth d, is a real number given by the function* eu *such that:*

$$\mathsf{eu}(a, C^d_{AF,t}) = \sum_{p \in C^d_{AF,t}} \mathsf{value}(a, p).$$

Example 16. *Consider the AF in Figure 5.1 with the topic being t. The estimated utility of c is the sum of two values in two argument paths. The two paths are $[t, a, b, c, d]$ and $[t, a, b, c, e]$, in which c has a value of $-\frac{1}{8}$ in both. This gives a total estimated utility for c as $-\frac{1}{8} - \frac{1}{8} = -\frac{1}{4}$.*

5.3 Heuristic strategy

A persuader using the heuristic strategy will not give up trying to convince the responder until it has run out of arguments to assert (known as an *exhaustive persuader* [18]). It uses estimated utility to determine which argument to assert, choosing one not yet asserted.

Definition 26. *Consider a persuader with a knowledge base $AF_P = \langle A_P, R_P \rangle$ participating in a dialogue $D = [M^P_0, M^R_0, ..., M^P_n, M^R_n]$, under a global knowledge $AF_G = \langle A_G, R_G \rangle$. The* **heuristic strategy** *for a depth d is given by the function* $\mathsf{hStrategy}_d$ *such that:*

- *if* $A_P \setminus \{M^P_0, ..., M^P_n\} = \emptyset$ *then* $\mathsf{hStrategy}_d(D) = \mathtt{terminate}$*, otherwise*
- $\mathsf{hStrategy}_d(D) = M$ *where* $M \in \{A \in A_P - \{M^P_0, ..., M^P_n\} \mid \forall B \in A_P - \{M^P_0, ..., M^P_n\}, \mathsf{eu}(A, C^d_{AF_G,t}) \geq \mathsf{eu}(B, C^d_{AF_G,t})\}$

Note that a persuader using the heuristic strategy can only assert arguments from their knowledge base, but uses global knowledge to determine which argument to assert. Similar to the virtual argument approach taken by Rienstra *et al.* [92], we assume that the persuader can only assert arguments they are aware of, but the persuader is aware

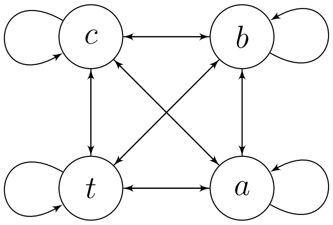


Figure 5.2: A framework with four arguments and the maximum number of attacks.

of the potential existence of all arguments in the domain, even those that they cannot themselves assert.

5.3.1 Complexity of the heuristic strategy

The complexity of generating the heuristic strategy is dependant on the complexity of two subcomputations: calculating the estimated utility for arguments in the framework up to the specified depth, and then sorting the arguments according to their estimated utility. The individual complexity of these computations are discussed below.

The complexity of calculating the estimated utility of the arguments in the heuristic strategy to a depth is dependant on the size of the complete set of argument paths (Definition 23); this is because the estimated utility of an argument is calculated from the position of the argument in *every* argument path it is in.

An argumentation framework with the maximum number of attacks (see Figure 5.2) is a framework with the maximum number of argument paths. There are $(n-1)!$ possible argument paths in such an argumentation framework (where n is the number of arguments in the framework). This can be seen by considering that there are $n-1$ possible arguments immediately following the topic, each of which have $n-2$ possible arguments immediately following them (since the argument immediately following the topic cannot be repeated because arguments in a path must be unique), and so on up to the final argument in each path. A visualisation of this is shown in Figure 5.3.

In the heuristic strategy, we only consider paths to a specified depth d. Therefore, where the depth is less than the number of arguments in the framework, the size of the the complete set of argument paths is reduced to $(n-1)! - (n-d)!$.

This means computing the estimated utility of the arguments in the framework up to the specified depth is $\mathcal{O}(n! - (n-d)!)$, which appears to be costly. However, in practice, frameworks are typically very sparse [121], and so the number of argument

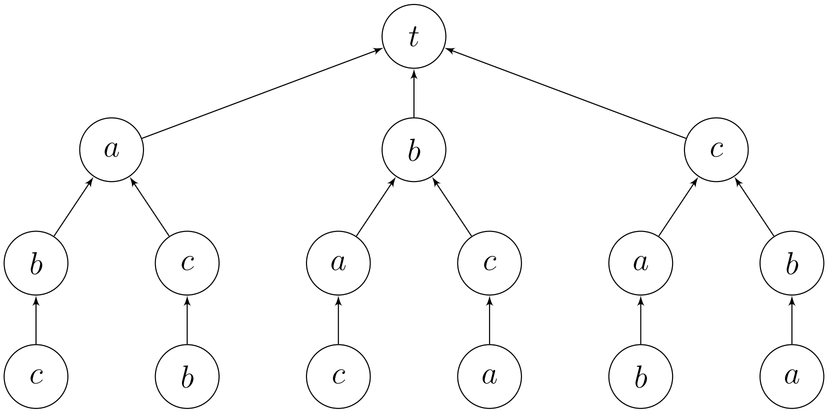


Figure 5.3: A visualisation of the complete set of argument paths for the framework in Figure 5.2. Note that there are four arguments in the framework, resulting in $(4-1)! = 6$ argument paths.

paths is reduced considerably. Also, we find that only a relatively small depth is required for effective strategies, which further reduces the practical complexity.

Once the estimated utilities of the arguments have been computed, they need to be sorted for the heuristic strategy. In the worst-case, all arguments in the framework will have an estimated utility, and will need sorting. This sub-computation therefore has the complexity of comparison sort, which is $\mathcal{O}(n \log n)$ [63] where n is the number of elements to be sorted (e.g. the number of arguments in the framework).

It may be possible to improve this complexity by combining the two subcomputations. The estimated utility is dependant on the position of the argument in the computed paths, and since the positions in the paths are already sorted in ordered structures, it may be that there is some overlap in the computation being done. However, in our implementation, we do not employ such a sophisticated algorithm, and instead deal with the two sub-computations separately. Nevertheless, as is shown in the following evaluation, the practical time taken to compute the strategy is sufficiently fast for our purposes, and is significantly faster than existing solutions that find the optimal strategy.

5.4 Experimental setup and implementation

To evaluate our heuristic strategy we generate random simple persuasion dialogue scenarios, in which the persuader selects which arguments to assert. As a benchmark for evaluation, we use a random strategy and a brute force strategy. The random strategy will assert one of its unasserted arguments at random until the responder is persuaded or there are no unasserted arguments. Other proposed approaches to generating a strategy for a persuasion dialogue use different models for the dialogue, and so a direct comparison to these approaches is not possible. A brute force strategy searches through assertions until either it has searched all assertions or it has found a series of assertions that convinces the responder; this is not a practical strategy as the persuader would not be able to determine how good a series of assertions is without asserting it, however this acts as a upper bound for comparison.

To generate a random simple persuasion dialogue scenario, an argument graph representing the global knowledge must be selected. In our experiments, we randomly generate two types of argument graph: tree-like graphs (Definition 29, an example is shown in Figure 5.4) and grids (Definition 30, an example is shown in Figure 5.5). This allows us to generate a large number of dialogue scenarios on which to run experiments.

In order to formally define a tree-like arguments frameworks, we first define directed walks in argumentation frameworks, and then define rooted-tree argumentation frameworks.

A *directed walk* between two arguments in an argumentation framework is a list of arguments such that, between each argument in the list, there is an attack from one to the other. Note that directed walks allow for repeated arguments, unlike arguments paths in which each argument must be distinct.

Definition 27. *A **directed walk** in an argumentation framework $AF = \langle A, R \rangle$ between an argument $a_0 \in A$ and an argument $a_n \in A$ is a list of arguments $[a_0, a_1, ..., a_n]$ such that $\forall i \in \{0, 1, ..., n-1\}$, $(a_i, a_{i+1}) \in R$.*

An argumentation framework is a *rooted-tree argumentation framework* with root r if there is an directed walk from every argument in the framework to r, and the framework is minimally connected in that removing any attack would mean the framework is not fully connected anymore. The framework in Figure 5.1 is a rooted-tree argumentation framework, where t is the root.

Definition 28. *An argumentation framework $AF = \langle A, R \rangle$ is a **rooted-tree argumentation framework** with root $r \in A$ iff $\forall a \in A$ there is a directed walk between a and r in AF, and $|A| = |R| + 1$.*

A tree-like argumentation framework is a rooted-tree argumentation framework with some additional random attacks added in. The additional attacks introduce the possibility of cycles existing in the framework, introducing arguments that both attack and

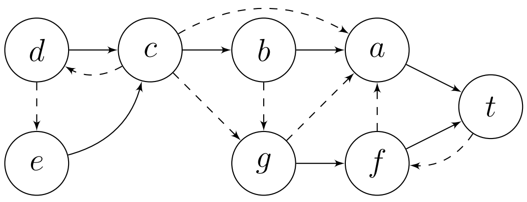


Figure 5.4: An example tree-like argumentation framework, based on the rooted-tree argumentation in Figure 5.1, where dashed attacks are the additionally added attacks.

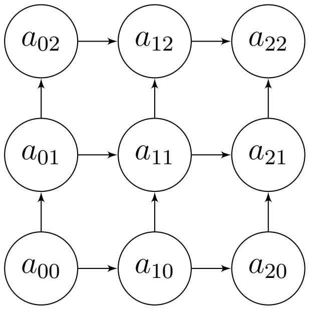


Figure 5.5: An example grid of size 3.

defend the topic, making the task of persuasion more complicated. An example tree-like argumentation framework is shown in Figure 5.4.

Definition 29. *An argumentation framework $AF = \langle A, R_0 \cup R_1 \rangle$ is a* **tree-like argumentation framework** *iff:*

- *$\langle A, R_0 \rangle$ is a rooted-tree argumentation framework with root $r \in A$,*
- *$|A| = |R_0| + 1$,*
- *$R_1 = A$, and*
- *$R_0 \cap R_1 = \emptyset$.*

We now define grids, the second type of framework structure used in our evaluation.

Definition 30. *An argumentation framework $AF = \langle A, R \rangle$ is a **grid** of size n with topic t iff:*

- *$t \in A$,*
- *$A = \{a_{i,j} : 0 < i, j < n\}$,*
- *$R = \{(a_{i,j}, a_{i+1,j}) : 0 < i < n-1, 0 < j < n\} \cup \{(a_{i,j}, a_{i,j+1}) : 0 < i < n, 0 < j < n-1\}$.*

Once the argumentation framework that represents the global knowledge has been generated, arguments are evenly distributed into the persuader's and responder's knowledge bases at random (with 50% of arguments in the persuader's and the other 50% in the responder's), only ensuring the topic argument of the dialogue is initially known by the persuader, but not by the responder. For our experiments the heuristic strategy considers argument paths up to depth 5; initial testing showed this allowed for a strong success rate while remaining fast to compute.

The implementation for the generation and testing of simple persuasion dialogues was written in Java, and run on a standard PC (1.86 GHz dual-core processor, 2GB RAM). We used libraries from Tweety [108] to determine whether the argument topic was acceptable under the preferred sceptical semantics for a given argument graph.

5.4.1 Experimental assumptions

In our experiments, we have make three key assumptions regarding the dialogue set up, which are set out in the previous sections. In this section, we examine these assumptions.

Virtual arguments

The heuristic strategy exploits knowledge that the persuader has of the dialogue domain; we make the assumption that the persuader knows about the existence of all the arguments in the global knowledge as virtual arguments, and uses this knowledge when evaluating which arguments should be asserted next. While this is a restrictive assumption, for a persuader to be effective it must have at least some knowledge either of the domain arguments, or of the persuadee's arguments. In comparison to the approach presented in this chapter, other mechanisms for generating dialogue strategies are similarly restrictive in that they assume the persuader has a model of the persuadee's arguments [15, 92] or of its expected behaviour [50]. However, in some domains it may be unrealistic to assume that the persuader has prior knowledge of the responder. Instead, virtual arguments act as a minimal form of opponent modelling: the persuader has a general sense of arguments that may be known by the responder, but without specific knowledge of the precise nature of these arguments [92].

We predict the heuristic strategy's success rate could be improved by incorporating knowledge of the arguments that are known by the responder into the utility calculation for arguments at a slight cost to computation time, in domains where such knowledge is available. This is discussed further in Section 5.6.2.

Framework structures

We use two types of framework structures in our evaluation: tree-like structures and grid-like structures. We use these structures of frameworks for our evaluation because they are fully-connected, sparse, and have few cycles in them. These properties are based loosely on argument frameworks transcribed from BBC Radio 4's Moral Maze program, in which experts aim to persuade a panel of an opinion [66], and so are somewhat relevant to real-world persuasion. We do not use the frameworks from the corpus itself, as there are too few instances to allow for a detailed evaluation.

We could have instead generated frameworks that held properties that were especially challenging for the strategy generation instead of generating semi-realistic argumentation frameworks. However, we use subset acceptability as a proxy for how challenging the dialogue is, and this demonstrates the limits of the heuristic strategy.

Distribution of arguments

In Chapter 3 we found that the similarity of participants' initial arguments at the start of a dialogue can have an impact on the outcome of the dialogue. Therefore, in these experiments we ensure that this parameter is kept constant. When generating the persuader's and responder's knowledge bases, we ensure that the sets of arguments that they know are disjoint.

For the simple persuasion scenarios we investigate here, and the behaviour of the proposed heuristic strategy, having the persuader and responder have arguments in common would not affect the computation speed of the heuristic strategy (as all arguments are still valued regardless of who they are known by). The outcome for the dialogue may still be affected in the same way as the deliberation dialogues in Chapter 3. However, this would be due to the properties of the scenario, rather than a shortcoming of the heuristic strategy. Therefore, we do not investigate any other distributions of arguments between participants.

5.5 Evaluation of the heuristic strategy

5.5.1 The heuristic strategy has a high success rate

It is desirable for a dialogue strategy to have a high success rate in achieving an agent's dialogue goals no matter what the agents know. For simple persuasion dialogues, this means that the persuader's strategy should have a high probability of persuading the responder of the topic argument. The heuristic, random, and brute force strategies were run on dialogues with domains of tree-like argumentation frameworks that had 8 arguments, with different proportions of argument subsets making the topic acceptable (this is the same property as the SA measurement from Section 4.4.3). The proportion of argument subsets that make the topic acceptable have been shown to be a strong indicator of how difficult a particular persuasion dialogue will be for the persuader [15]: if there are few subsets of the framework in which the topic is acceptable, it is likely to be harder for the persuader to manipulate the persuadee's framework into one in which the topic is acceptable. The probability of persuader success for the strategies was determined by running many simulations of dialogues, each with a different randomly generated argumentation framework, and recording the percentage of argument subsets that make the topic acceptable in the argumentation framework, as well as whether the persuader is successful when using the heuristic, brute force, or random strategy. The results are shown in Figure 5.6.

We observe a similar trend for all strategies: as the proportion of argument subsets of the global knowledge that make the topic acceptable increases, so does the likelihood that the strategy is successful. At proportions of subsets making the topic acceptable it is likely that, given the arguments known by the responder at the start of the dialogue, it may be impossible for the persuader to be successful. At the other extreme of 50% of subsets making the topic acceptable (in the topic is acceptable in all subsets that contain it) the persuader only has to assert the topic argument at some point in the dialogue in order to convince the responder: since both the heuristic strategy and random strategy are exhaustive, asserting all their arguments until the responder is convinced, it is guaranteed that the persuader will eventually be successful in these scenarios.

The results show that the heuristic strategy is, on average, significantly more likely to be successful than the random strategy. In some scenarios, the heuristic strategy is over three times more likely to be successful than the random strategy. In comparison, the brute force strategy, which is the optimal, is better than the heuristic strategy. However, the difference between the brute force strategy and heuristic strategy is noticeably less than the difference between the heuristic strategy and the random strategy.

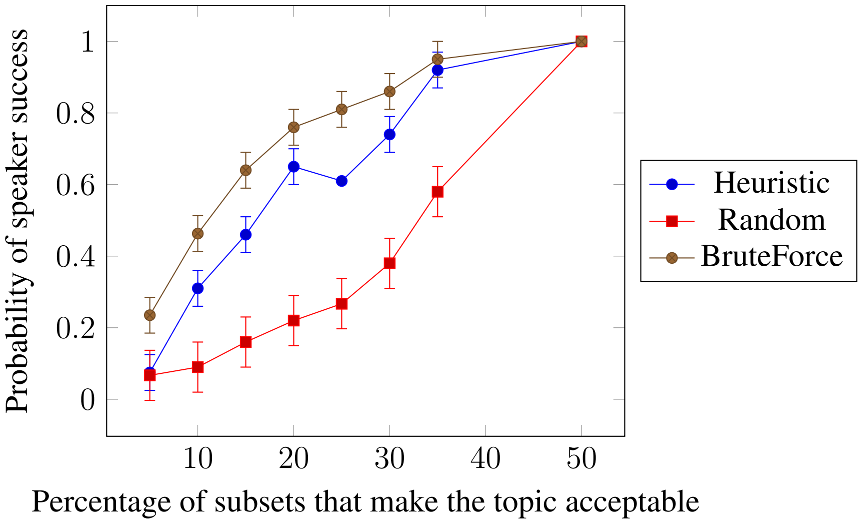


Figure 5.6: Percentage success rate of heuristic, random, and brute force strategies.

Table 5.1: Time to compute heuristic strategy (seconds). *Args* is the number of arguments in the domain.

Args	10	20	30	40	50
Time	<0.1	0.21	0.37	0.56	0.77

5.5.2 The heuristic strategy is fast to compute

To determine the computational cost of generating the heuristic dialogue strategy, we measure the time taken to compute the heuristic strategy in a randomly generated dialogue scenario. We generated tree-like argumentation frameworks of increasing sizes $\{10, 20, 30, 40, 50\}$. The results are shown in Table 5.1, giving the average time for 1,000 random dialogue scenarios. For domains with fewer than 10 arguments the generation of the strategy took less than 0.1 seconds. At 11 arguments, the increase in time is noticeable, allowing computation of the heuristic strategy in less than a second for as many as 50 arguments in the domain. The results demonstrate that the heuristic strategy is efficiently scalable with large numbers of arguments.

5.5.3 The heuristic strategy succeeds with many arguments

As can be seen from the results in Figure 5.6, the chance of successfully convincing the responder depends heavily on the particular argument graph that determines the global knowledge. The more subsets of arguments from the global knowledge that determine

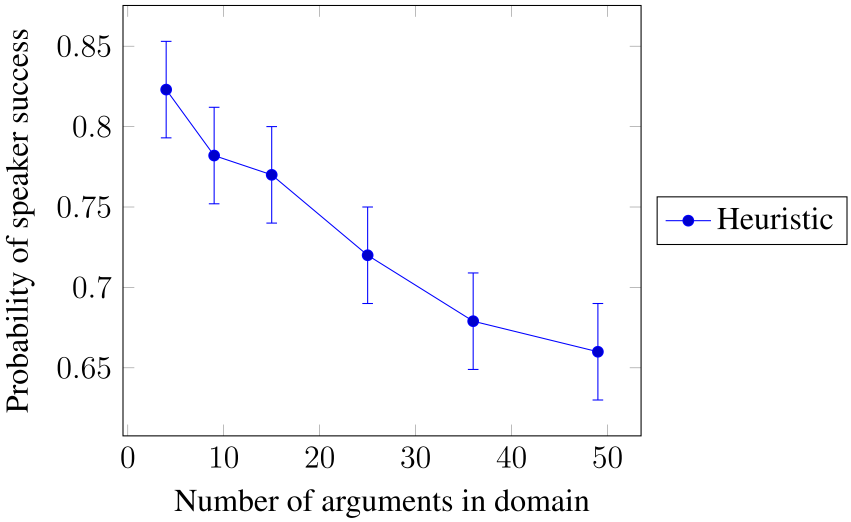


Figure 5.7: Success of the heuristic strategy with increasing numbers of arguments.

the topic to be acceptable, the more chance of reaching a point in the dialogue where such a set of arguments is available to the responder, causing it to terminate the dialogue successfully. To investigate how the performance of the heuristic strategy scales with the number of arguments we needed to generate global knowledge argument graphs in such a way that the proportion of argument subsets that determine the topic to be acceptable remains near constant as the size of the graphs increases. Thus, here we used partial grids, which allowed us to keep the average percentage of subsets of the global knowledge that make the topic acceptable within the range 25%–35% for all argument graphs we experimented with. We observe in Figure 5.7 that there is a slight decrease in the success rate of the heuristic strategy as the number of arguments increases because, as the argument graph grows, so does its complexity, and these complexities are ignored by the heuristic strategy. The decrease in success can be considered a necessary sacrifice for a computationally tractable strategy.

5.6 Suitability of heuristic strategy for other dialogues

In the previous section we demonstrated the effectiveness and efficiency of the heuristic strategy for the simple persuasion scenario presented earlier (Definition 20). However, we do not believe the heuristic is limited to just this simple scenario, but rather that it can be applied to a range of sophisticated persuasion dialogue types. In this section, we discuss how the heuristic is still likely to be useful for more sophisticated persuasion dialogues with more sophisticated persuadees, multiple perusadees, and opponent models.

5.6.1 Persuadee may assert arguments

In the simple persuasion dialogue, the responder does not themselves assert any arguments, instead it only states each round whether it has been successfully persuaded. In many persuasion dialogue models both participants are able to assert arguments as part of the dialogue; this feature opens up the possibility of the persuadee asserting arguments that were not previously known by the persuader, in the case where the persuader does not have knowledge of the global argumentation framework.

In such dialogues, the heuristic can still provide an estimate of which order arguments should be asserted, and would behave in the same way unless the persuadee does assert an argument that was not previously known by the persuader. In the case where the persuadee asserts an argument not previously known by the persuader there are two options for the heuristic strategy. The first option is to ignore any new arguments, and continue regardless: this would not have any impact on the effectiveness or efficiency of the heuristic strategy because the behaviour of the persuader is unchanged, and thus the outcome of the dialogue would be the same.

The second option is to encorporate any new arguments asserted by the persuadee within the persuader's framework, and then recompute the heuristic strategy with the new information: this makes use of the additional knowledge the persuader has in such dialogues, and is like to make the strategy more effective. It is likely that with additional knowledge, the heuristic would be more accurate, and therefore would more accurately estimate which arguments are the most effective to assert. However, the strategy would also be less efficient, since the heuristic has to be recomputed every time the persuadee puts forward a previously unknown argument.

5.6.2 Opponent models

Some dialogue models assume that the persuader has an opponent model of the the persuadee. This gives the persuader an advantage in that it has knowledge of which arguments are known by the persuadee, and therefore can determine more effectively which arguments can be put forward to be convincing: the persuader knows which arguments need to be countered, and which arguments the persuadee already has a counter-argument to.

Again, the heuristic strategy could be applied to such a dialogue by simply ignoring the additional information in the opponent model; this would have no impact on the effectiveness or efficiency of the strategy as the persuder would behave in the esame way and so the outcome of the dialogue would be the same. However, the heuristic could be adjusted to make use of such opponent models, and therefore improve the effectiveness at some additional computational cost; this would combine the heuristic's weighting of arguments with the opponent model's weighting of which arguments are known by the persuadee. For example, arguments that the persuader knows the persuadee knows could be weighted have a higher impact within the calculation of the heuristic.

5.6.3 Multiple persuadees

The simple persuasion dialogue has only two participants. Some other dialogue models allow for multiple persuadees, in which the goal of the persuader is to persuade as many persaudees as possible. Adding additional persuadees does not increase the time taken to compute the heuristic strategy because the heuristic is computed on the argumentation framework of the persuader, instead on each individual persuadee framework. We expect the effectiveness of the heuristic strategy to be unchanged as the multi-persuadee dialogues would behave in the same way as multiple single-persuadee dialogues.

5.7 Conclusions and Discussion

In this chapter we have presented and evaluated a heuristic strategy that can be used in persuasion dialogues. Our results show that this heuristic strategy is fast to compute, even for domains with a large number of arguments, which had not been shown to be the case for existing approaches that generate optimal strategies [15, 50, 92].

The heuristic strategy was evaluated by applying it to simple persuasion dialogues, in which the responder acts truthfully, and only in response to the persuader. The scenario we investigate has some application to real-world scenarios: Consider the example of an agent trying to persuade an administrator to grant them privileged security permissions: the agent can assert arguments in order to convince the administrator that it should be granted, but the administrator does not have the resources to respond to all requests with any more than a notification of acceptance or rejection. In future work, we intend to investigate the performance of the heuristic strategy in more complex scenarios, specifically persuasion dialogues involving more than two participants, each of which may have their own set of beliefs. We expect that existing approaches for determining optimal strategies [15, 50, 92] would be even more computationally expensive here. This is partly due to the fact these approaches use probabilistic information about the opponent to determine the strategy, and with additional opponents added the number of possible states in set of all opponent models grows exponentially.

Argument strategies that use heuristic information have also been investigated in different types of dialogue. Kontranis *et al.* evaluate a set of heuristic-style strategies that agents use in a dialogue-type scenario, in which participants vote on the attacks between globally known arguments, with the goal to reach a consensus [65]. In comparison, the heuristic strategy we present is based on a typical dialogue game in which agents assert arguments, rather than the focus of communication being on attack relations. Wardeh *et al.* investigate PADUA, a dialogue protocol allowing agents to classify objects based on evidence from previous examples of object classification [117]. Depending on whether the opponent is agreeable or not, the persuader can select the appropriate heuristic strategy in order to increase their success rate in deciding upon their desired classification. However, Wardeh *et al.* do not investigate the scalability or performance of their proposed strategies.

Oren *et al.* [84] also present a heuristic for determining a strategy in a more general form of agent dialogue than the simple persuasion dialogue considered in this chapter. In their work, Oren *et al.* consider instantiated arguments, constructed as a series of literals in support of a conclusion literal, as opposed to the abstract arguments we consider here. Oren *et al.*'s heuristic guides the dialogue participant in selecting arguments that when asserted would minimise the information that is exposed to other participants (as measured by the number of literals revealed) whereas our heuristic estimates which arguments would be most beneficial to assert in order to achieve the participant's goal. However, we do not considered any associated cost with asserting arguments in this chapter. In the next chapter, we propose a different approach to generating a strategy, and we will considered a cost to asserting arguments.

Chapter 6

Deriving persuasion strategies using search-based model engineering

6.1 Introduction

Persuasion is the task of inducing the acceptance of a belief in other agents. In the previous chapter, we consider a particular type of persuasion dialogue in which there is one persuader and one persuadee. This chapter focuses on a one-to-many persuasion setting, where a single persuader broadcasts arguments to a multi-party audience with the aim of convincing them of some goal argument. Since each individual audience member reasons with its own set of personal knowledge (which we assume is known to the persuader) any particular set of persuader arguments may be convincing to some audience members but not others, and so the persuader must carefully select which arguments it should assert in order to maximise the number of audience members it convinces. This is a challenging problem because of the number of potential solutions and the number of audience members to evaluate against: to exhaustively explore the solution space, for each subset of the persuader's arguments one must consider each audience member and determine whether it would be convinced by those arguments.

A political speech is an example of many-to-one persuasion, in which the politician attempts to persuade the public (comprised of many individual agents) that their party is the one to vote for at the next election. In such a dialogue, the politician wants to maximise the number of agents that are convinced.

Much of the recent work looking at strategic argumentation settings has focused on one-to-one persuasion, e.g., [16, 50, 95, 51, 56, 92]. A notable exception is the work of

Hunter and Thimm [59], who also consider how to determine which set of arguments to present to an audience, using probabilistic argumentation to capture uncertainty about the audience members' beliefs. In contrast to our approach, they do not allow for a range of audience members each with different beliefs. Furthermore, their approach has been shown to apply to settings with up to 7 arguments, while we show that our approach scales to more than 200 arguments. In earlier work [54, 55], Hunter looks at how one can select arguments that will resonate with a particular audience, but this similarly assumes a typical audience member, while our approach allows representation of distinct audience members. Bench-Capon *et al.* present a framework that can be used to describe audiences comprised of members with different values [11], but do not address the strategic considerations of the persuader in such a domain.

To efficiently determine the arguments the persuader should assert, i.e., its *strategy*, we apply techniques from *search-based model engineering* (SBME) [20]. By representing the persuasion setting as a meta-model (a schema describing the structure of valid solutions), we can apply evolutionary search to a find a near-optimal strategy for the persuader that maximises the number of convinced audience members. We ran experiments over a range of settings, varying both the size of the problem and the structure of argumentation framework representing the underlying knowledge available to the persuader and audience members, and show that our approach:

C1 produces strategies that are effective in convincing members of the audience;

C2 finds strategies efficiently, in that it scales well with increasing numbers of arguments in the domain, and increasing numbers of audience members; and

C3 can efficiently find strategies that satisfy multiple objectives (in particular, maximising convinced audience members while minimising arguments asserted).

This work is the first to apply evolutionary search to strategic argumentation. McBurney and Parsons [70] propose the possibility of applying an evolutionary algorithm to automate a chance discovery dialogue, where individuals exchange knowledge with the aim of discovering unknown risks and opportunities, but, while they outline their proposed approach, it has not been specified in detail. While the focus in our approach is on a one-to-many persuasion setting, where a persuader uses its knowledge of the audience members to select a set of arguments to assert, the approach we present is sufficiently flexible to capture a range of argument dialogue settings, and we discuss in Section 6.6 our plans to extend this work to account for uncertainty in the persuader's knowledge of its audience and to allow dialogues in which each party may present arguments.

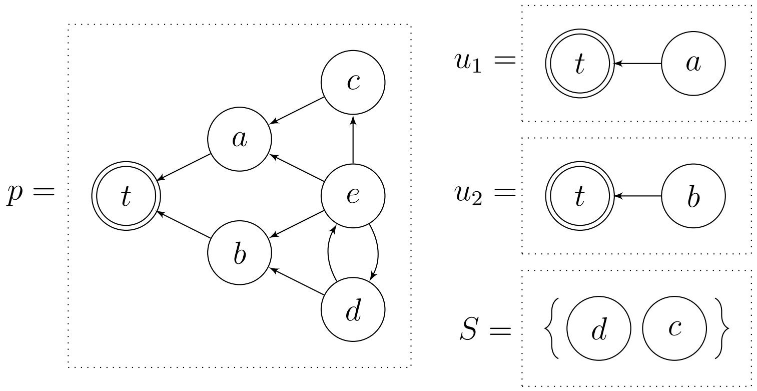


Figure 6.1: A multi-audience persuasion game, with persuader p, persuadees u_1, u_2, and strategy S.

Further, in their outline, McBurney and Parsons [70] describe a more traditional approach to using evolutionary algorithms, in which candidate solutions are represented as binary strings. In contrast, our approach represents candidate solutions as high-level models instead of binary strings. The approach of representing candidate solutions as high-level models has two advantages. First, the candidate solutions are more understandable by a human user and so the process is more transparent as it does not deal with abstract representations such as binary strings. Second, the computational cost of the transformation from a model to a solution is potentially less than that of the transformation between a binary string and a solution as the mapping from a binary string to a solution is likely to be a more complex process.

This chapter is set out as follows. In Section 6.2 we formally define our multi-audience persuasion setting and introduce search-based model engineering in Section 6.3. Section 6.4 explains how we represent multi-audience persuasion as a meta-model and use this to search for persuader strategies. We evaluate our approach in Section 6.5 and finish with discussion in Section 6.6.

6.2 Multi-Audience Persuasion Games (MAPGs)

We consider a *multi-audience persuasion game* (MAPG), in which a persuader seeks to convince a set of persuadees, known as the audience, that a particular topic argument is justified. The persuader's knowledge is represented by an AF, from which each persuadee's knowledge is a subset. The audience captures each persuadee's knowledge, thus we assume that the persuader has certain knowledge of the audience members; we discuss in Section 6.6 how our approach can be adapted to allow for uncertain knowledge of the persuadees. The persuader's strategy is a subset of the persuader's AF, which are the arguments the persuader will assert to the audience. We assume *without loss of generality* that persuadees each know the topic argument before the persuader presents their arguments.

Definition 31. *A **multi-audience persuasion game** is a tuple $g = \langle p, t, U, S \rangle$, such that:*

- *$p = \langle A_p, R_p \rangle$ is the argumentation framework belonging to the **persuader**.*
- *$t \in A_p$ is the **topic**, the argument the persuader tries to convince the audiences of.*
- *$U = \{u_1, ..., u_n\}$ is the **audience**, where $u_i = \langle A_i, R_i \rangle$ is the argumentation framework belonging to persuadee i, s.t. $A_i \subseteq A_p$, $R_i \subseteq R_p$, and $t \in A_i$.*
- *$S \subseteq A_p$ is the persuader's **strategy**.*

An example MAPG is shown in Figure 2. Note, the persuader's strategy is asserted to all persuadees at once; the persuader cannot choose to assert an argument to only a subset of persuadees. In this chapter, we consider that a persuadee is convinced if, under their framework combined with the strategy, the topic is justified under the preferred credulous semantics (which are well-suited to practical reasoning about what to do [86]). However, the approach detailed in this chapter could easily be adapted to other semantics, or indeed any arbitrary function that maps an argumentation framework to a set of justified arguments.

Definition 32. *We denote the **justified arguments under the preferred credulous semantics** as $\sigma(AF) = \{a \mid \exists S \subseteq A$ s.t. S is maximally admissible and $a \in S\}$*

Definition 33. *In a multi-audience persuasion game $g = \langle p, U, t, S \rangle$ with the persuader's framework $p = \langle A_p, R_p \rangle$, a persuadee i with AF $\langle A_i, R_i \rangle \in U$ is **initially convinced** in g iff $t \in \sigma(\langle A_i, R_i \rangle)$. The function $\gamma(g, u_i) \to [0, 1]$ returns 1 iff u_i is initially convinced in g, 0 otherwise. Similarly, i is **convinced** in g iff $t \in \sigma(\langle A_i \cup S, R_p \cap (R_i \cup (A_i \cup S)^2) \rangle)$. The function $\hat{\gamma}(g, u_i) \to [0, 1]$ returns 1 iff u_i is convinced in g, 0 otherwise.*

A persuader is typically interested in convincing as many persuadees as they can. We measure *effectiveness* of a strategy as the increase in the number of convinced persuadees from those that are initially convinced. By asserting arguments, the persuader may dissuade audience members of the topic; a persuader that dissaudes more audience members than they persuade will have a negative effectiveness. As well as trying to convince as many persuadees as possible, the persuader may also wish to minimise some *cost* associated with asserting a strategy. For this work, we assume the cost of a strategy is the proportion of the persuader's arguments put forward in the strategy. The persuader wants to minimise the number of arguments they present, since more arguments may lead to audience disengagement [58]. We refer to this cost as the *efficiency* of a strategy. We assume each persuadee is as *valuable* as each other, and so all persuadees are of the same importance.

Definition 34. *The* ***effectiveness*** *of the strategy in a multi-audience persuasion game* $g = \langle p, U, t, S \rangle$*, denoted* $\epsilon(g)$*, is:* $\sum_{u \in U} \hat{\gamma}(g, u) - \sum_{u \in U} \gamma(g, u)$*.*

Definition 35. *The* ***efficiency*** *of the strategy in a multi-audience persuasion game* $g = \langle p, U, t, S \rangle$ *with persuader's framework* $p = \langle A_p, R_p \rangle$*, denoted* $\kappa(g)$*, is:* $\frac{|A_p| - |S|}{|A_p|}$*.*

Example 17. *Consider the example multi-audience persuasion game in Figure 2. The* effectiveness *of the strategy is* 2*, as both persuadees will find the topic acceptable once the arguments in* S *are added to their respective frameworks. The* efficiency *of the strategy is* $\frac{6-2}{6} = \frac{4}{6}$ *as two argument are asserted in the strategy. Note, that had the persuader asserted the argument* e *as their strategy instead then the effectiveness would remain the same but the efficiency would be improved to* $\frac{6-1}{6} = \frac{5}{6}$*.*

We use evolutionary search to find an effective and efficient strategy of a multi-audience persuasion game. We implement the problem using SBME, which provides a natural and efficient encoding.

6.3 Search-Based Model Engineering (SBME)

Search-based methods have long been used to solve optimisation problems [35]. Here, we give an overview of search-based methods, before examining SBME in more detail.

6.3.1 Meta-heuristic search.

Many optimisation problems can be solved by dedicated algorithms or using specialised heuristics. However, as problems become more complex, it often becomes more efficient to find (near-)optimal solutions using meta-heuristic search techniques. These techniques start from one (or a population of) randomly generated *feasible candidate solutions* (i.e., solutions that satisfy all relevant constraints) and incrementally change these to explore the solution search space. The quality of any candidate solution is indicated by one or more *objective functions*—functions that take a solution and provide a numeric value indicating relative quality. A meta-heuristic algorithm then evolves the population of candidate solutions by:

1. creating a set of new candidate solutions derived from the existing solutions;
2. ranking old and new candidate solutions according to their objective values; and
3. keeping only the highest-ranked n candidate solutions for the next round.

The algorithm ends either when a pre-defined number of evolutions have been explored or when another stopping criterion has been reached (e.g., when the objective values of candidate solutions no longer change significantly).

Different meta-heuristic algorithms use different techniques for encoding solutions and deriving new ones, as well as for ranking solutions. Here, we focus on evolutionary search techniques, which derive a new candidate solution from each existing candidate solution by applying a *mutation operator* randomly picked from a pre-defined set.

6.3.2 Search-based model engineering.

SBME [123, 62] aims to apply meta-heuristic search techniques in the context of model-driven engineering (MDE). Specifically, SBME techniques search for models that are optimal as defined by some objective functions.

To understand SBME, we first need to briefly introduce key notions of MDE, such as model, meta-model, and model transformation. MDE's central tenet is that software should be developed using high-level models, expressed in domain-specific modelling languages, rather than by directly writing programs in general-purpose modelling languages such as Java or C. Key to this is the ability to define modelling languages and automatically and efficiently manipulate models expressed in these languages. Meta-models support this by providing a formalised representation of a modelling-language's abstract syntax; that is, the concepts of the language and their interactions. Typically in MDE, meta-models are expressed as class diagrams. Models are considered valid iff they are an instance of the meta-model; that is, if every model element is an instance of a corresponding meta-model element and all connections between model elements are specified according to the associations defined in the meta-model. Model transformations, finally, are programs that take models as input and produce new models as outputs (possibly instances of different meta-models).

By employing SBME techniques for specifying optimisation problems we benefit from three main advantages:

1. we can use the concept of model transformations to simplify the definition of complex search operators that can ensure consistency of the generated offspring;
2. the use of models allows us to use the user's domain expertise to consistently encode complex problems and solutions and, we can ensure that the search space exploration is done without generating inconsistent solutions;
3. this approach does not require the step of genotype to phenotype mapping that would otherwise be required in traditional genetic programming approaches.

6.4 Multi-Audience Persuasion as a Search-Based Model Engineering Problem

To represent a multi-audience persuasion game (MAPG) as a search-based model driven engineering problem, we must first define a metamodel that encodes the space of possible solutions. This is shown in Figure 6.2 and we explain now how this corresponds to our MAPGs (Definition 3). The persuader's AF ($\langle A_p, R_P \rangle$ in Def. 3) and the persuadees' AFs ($\langle A_i, R_i \rangle$ in Def. 3) are represented by the `PersuaderAF` class and the `PersuadeeAF` class respectively. An `MAPG` has exactly one persuader and multiple persuadees, captured by the multiplicity constraints in Figure 6.2 (1, resp. * for many). The persuader framework contains all `Argument`s, denoted by the composition link between `PersuaderAF` and `Argument`, while persuadee frameworks contain some subset of the arguments. Arguments may attack one another (captured by the `attacks` edge in Figure 6.2), and exactly one argument is distinguished as the `topic`. Multiple arguments can be identified as forming the strategy of an MAPG (S in Def. 3), captured via the `strategy` link between the `MAPG` and `Argument` classes.

For a particular persuader, audience and topic argument, we are interested in finding a strategy that is effective and also, perhaps, efficient. To do this with our SBME approach, we mutate the strategy using two mutation operators: the first adds a new argument to the strategy; the second removes an argument from the strategy. The rest of the model does not change. Applying these mutations to the solution candidates allows exploration of any strategy in the search space. Two objective functions are used to evaluate any strategy found: one determines its effectiveness and one determines its efficiency. We can then apply an evolutionary search algorithm to find strategies that

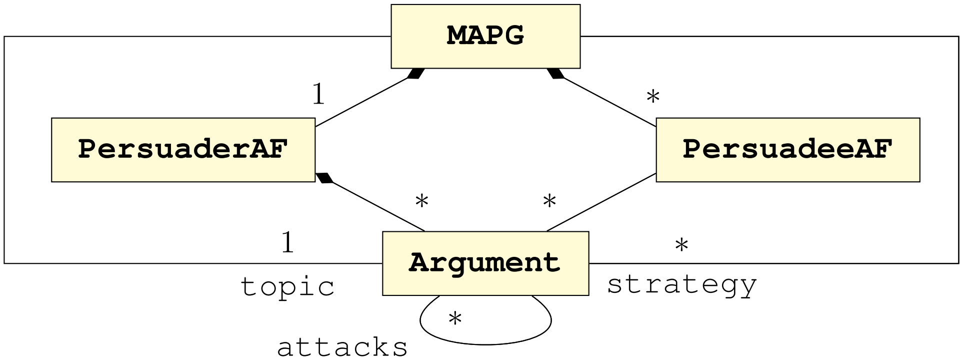


Figure 6.2: Metamodel for multi-audience persuasion games, as a class diagram.

perform well against these objectives as follows: (1) randomly generate a population of instances of the metamodel; (2) apply a random mutation operator to each member of the population, evaluate these against the objective function(s), select the most promising individuals for the next generation; (3) repeat (2) until the configured number of generations has been reached.

6.5 Evaluation of Application of SBME to Find Strategies for MAPGs

We run experiments to investigate performance of our approach, looking both at the quality of solutions found and the time taken to find them. We use the SBME tool MDEOptimiser (MDEO)[1] to run a population based evolutionary algorithm over models that instantiate the metamodel we define in the previous section, where only the strategy is mutable. The algorithm is run for 250 generations with population size 30. We use Tweety [109] together with the *argmat-sat*[2] argument solver to determine whether a particular persuadee is convinced by a strategy. This is one of the fastest solvers for the required preferred decision problem, but it is worth noting that the performance of the solver that is used has a significant effect on time taken to evaluate a strategy.

Across experiments, we vary: the number of arguments in the persuader's framework (af-size); the structure of the persuader's framework (struct); the number of persuadees (p-num); and the number of arguments known to each persuadee, expressed as a proportion of the number of arguments in the persuader's framework (p-size).

[1] https://mde-optimiser.github.io/

[2] http://sites.google.com/site/argumatrix/argmat-sat

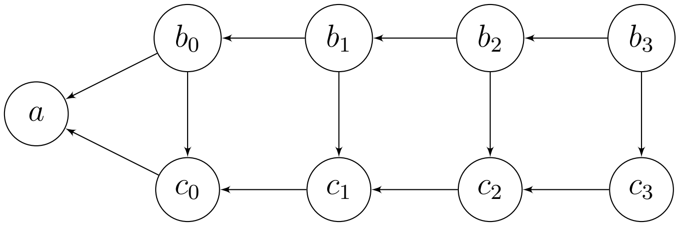


Figure 6.3: An example ladder. The topic is a.

6.5.1 Framework structures

We perform experiments with the following argumentation framework structures.

Ladders and Cycles

These are used in Black et al.'s evaluation of their strategies for one-to-one persuasion dialogues [16]. We reuse them here as they are designed to be especially challenging for persuasion, due to the existence of arguments that may be both beneficial or detrimental for a persuader, depending on the persuadee's beliefs.

Definition 36. *A* ***ladder*** *of size n with topic t is an argumentation framework $AF = \langle A, R \rangle$ where:*

- $A = \{t\} \cup \{b_i, c_i : i < n\}$*, and*
- $R = \{(b_0, t), (c_0, t)\} \cup \{(b_i, b_{i-1}), (c_i, c_{i-1}) : 0 < i < n\} \cup \{(b_i, c_i : i < n\}$.

An example ladder is shown in Figure 6.3.

Definition 37. *A* ***cycle*** *of size n and with topic a is an argumentation framework $AF = \langle A, R \rangle$ where:*

- $A = \{t\} \cup \{b_i, c_i : i < n\}$*, and*
- $R = \{(b_i, t), (c_i, b_i) : i < n\} \cup \{(b_{i-1}, b_i), (c_{i-1}, c_i) : 0 < i < n\} \cup \{(b_{n-1}, b_0), (c_{n-1}, c_0)\}$.

An example cycle is shown in Figure 6.4.

Trees

These are rooted-tree argumentation frameworks, whose root is the topic argument. See Chapter 5, Definition 29 for the formal definition. As a bipartite AF, these are expected to be less challenging for the persuader than ladder or cycle AFs, since there is no risk in asserting an argument that supports the topic.

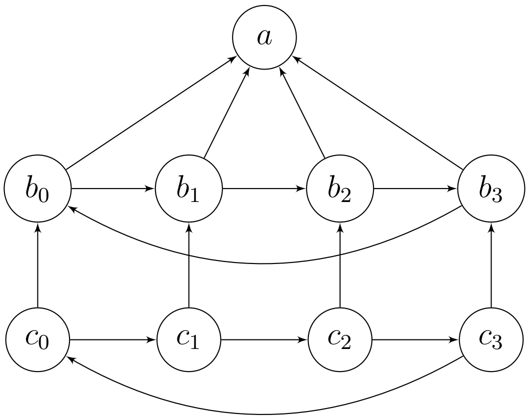


Figure 6.4: An example cycle. The topic is a.

Competition Frameworks

We take three AFs from the set used in The Second International Competition on Computational Models of Argumentation, specifically one derived from a planning problem (with 490 arguments), one based on a Barabási-Albert network (with 160 arguments), and one translated from assumption-based argumentation (with 691 arguments).

For Ladders, Cycles, and Trees we can vary the size of the AF, but for the competition frameworks this is fixed. The framework is used as the persuader's AF. The `p-num` persuadees' AFs are uniformly random sub-graphs of the persuader's AF, each composed of `p-size` × `af-size` arguments (recall, `p-size` is a proportion), one of which is ensured to be the topic argument.

6.5.2 Alternative approaches for comparison

As no existing work allows generation of strategies for multi-audience persuasion games, we benchmark our approach against two naive alternative approaches, described below.

- **Brute-force (BF)** searches through all possible assertions (that is, the power set of the arguments in the persuader's AF) to find a strategy that maximises the number of persuadees that are convinced. If, during the search, a strategy is found that convinces all persuadees then the search terminates, otherwise the search is exhaustive. This approach is computationally intractable for large games, as shown in Table 6.3, but for smaller AFs it is feasible to use this approach to determine an optimal solution.

- **Random asserter (RA)** first selects a uniformly random number of arguments to assert, from 0 to the size of the persuader's AF. Then a uniformly random subset of this size is selected from the arguments in the persuader's AF to assert.

6.5.3 Hardware details

We ran our experiments on Amazon Web Services Elastic Compute Spot instances. The experiments have been configured to run inside a Docker container running Java 1.8.0 and Amazon Linux. Each experiment has been performed on an individual machine, with 2 CPU cores and 2.5GB RAM allocated to the container. For each experiment, we ran MDEO 10 and RA 10 times, so as to consider both average and best performance. The complete implementation and the obtained results can be downloaded from GitHub[3].

6.5.4 C1: MDEO finds strategies that are effective

We compare the performance of our approach to RA and BF, considering here the single objective to maximise the number of persuadees who are convinced. We use three settings: (1) small games where `struct` ∈ {`cycle, ladder, tree`} of `af-size` ∈ {21, 51, 101} (e.g. 20 arguments + 1 topic argument), with `p-num` ∈ {1, 2, 5} persuadees, with `p-size`= {.25, .5, .75}; (2) larger games where `struct` ∈ {`cycle, ladder, tree`} of `af-size` ∈ {51, 101, 201}, with `p-num` ∈ {10, 50, 100} persuadees, with `p-size` ∈ {.25, .5, .75}; (3) games using the competition frameworks, with `p-num` = 50 and `p-size` = .5.

As BF is an exhaustive search, the strategies it returns are guaranteed to be optimal. However, BF is computationally intractable and so we are unable to compute the best outcome for larger games (with a 24 hour time-limit). For the games where we were able to use BF to determine the best outcome (where `af-size` ≤ 21) *MDEO always found an optimal solution*.

Tables 6.1 and 6.2 compare performance of RA and MDEO, showing the average effectiveness of each solution found (Ma for MDEO, Ra for RA) and the effectiveness of the best solution found (Mb for MDEO, Rb for RA). For smaller games (Table 6.1) both approaches generally found the best solutions, but the average effectiveness of the strategies found using MDEO is significantly better than for RA. For larger games, Ta-

[3] `https://github.com/mde-optimiser/comma-18-mapg`

ble 6.2, we see that MDEO produces better average solutions than RA, and the best solutions of MDEO are better than the best of RA. Cycle AFs proved difficult for both approaches, often resulting in failure to find a strategy that increases the number of convinced persuadees. We plan to investigate whether by giving MDEO a larger population of solutions, or more evolutions, we may be able to find solutions for cycle AFs at the cost of additional computational resources.

Results for the competition frameworks are shown in Table 6.4. Our approach was unable to cope with the largest of these frameworks (with 691 arguments), timing out after 24 hours, but was able to find effective strategies for the smaller competition frameworks.

6.5.5 C2: MDEO can find solutions to large problems

For a single objective to maximise the number of convinced persuadees, we compare the average time taken by MDEO to find a strategy with the time taken by BF. Table 6.3 shows the results for small games. For games with `af-size` larger than 11, the MDEO approach is almost always faster than BF search. Exceptions to this (e.g. Ladder-21, with `p-num` and `p-size` 25%) are when BF gets 'lucky', and quickly finds a solution that convinces all persuadees. For games with `af-size` of 11, BF is faster. However, closer observation of the MDEO search reveals that the best solution is actually found in earlier generations. Therefore, for these scenarios, MDEO runtime can be improved by specifying a lower number of generations, without an effect on the quality of solution produced.

For larger games, with `af-size` up to 201 arguments and number of persuadees up to 50, MDEO returns results within 90 minutes. (Full results are omitted here for space reasons but can be found in our repository. This demonstrates the scalability of MDEO, both to the number of arguments in the domain, but also with increasing numbers of persuadees. Table 6.4 shows results for the competition frameworks: MDEO took more than 24 hours to run for the largest of these, just over an hour for the framework with 480 arguments, and less than 16 minutes for the smallest competition framework.

6.5.6 C3: MDEO can find strategies that satisfy multiple objectives

Here we seek strategies that aim to both maximise the number of convinced persuadees and minimise the number of arguments asserted. We compare both the efficiency and effectiveness of the strategies produced by MDEO and RA for this multi-objective case. To compare the quality of search solutions with two objectives we use the hypervolume (HV) unary quality indicator, proposed by Zitzler et al. [122]. The HV measures the volume of objective space dominated by a set of objectives that form a Pareto front. The HV metric must be maximised and the Pareto front with the higher HV value is considered better. To use RA to determine a Pareto front, each run consisted of a batch of 10 applications of RA (so in total we ran RA 10×10 times for each experiment). For space reasons, we consider only games based on the larger AFs, including competition frameworks, and do not consider cycles, for which it is hard to find a solution that satisfies a single objective.

For the larger tree and ladder problems, we compare the hypervolumes over 10 runs, included as box plots in Figure 6.5. In almost all cases, the average hypervolume obtained by MDEO is higher than the one obtained with RA, indicating that MDEO outperformed RA. Furthermore, MDEO performance is more consistent than that of RA (in the box plots, vertically smaller plots indicate a smaller variance in the individual Pareto fronts). For the competition-based games, the hypervolumes are shown in Table 6.4. We see that for games with frameworks with 160 and 480 arguments MDEO is able to return a solution in a reasonable time.

In two of the evaluated scenarios (indicated on Figure 6.5 with asterisks), RA found a better solution than MDEO. We have repeated these experiment by adding two additional mutation operators that can assign and remove 10 arguments each time, instead of one. This allows MDEO to outperform RA in these cases, indicating that mutation operators which only change a single argument may sometimes be insufficient to allow the search to escape from a local maximum.

Across all experiments, the average time taken for MDEO to find the strategy for the two objective case was not statistically longer than the time for the single objective case. Indeed, due to the non-deterministic search of MDEO, there were many scenarios in which the two objective cases were faster. This demonstrates that there is no computational overhead for adding the additional objective.

6.6 Conclusions and Discussion

We have shown that we can use techniques from SBME to represent the multi-audience persuasion setting as a meta-model, to which we can apply evolutionary search to find

Table 6.1: Average and best effectiveness of solutions found by MDEO (respectively, Ma, top left, Mb, bottom left) and by RA (respectively, Ra, top right, and Rb, bottom right) for small games. The results in bold are the better performing approach for a game. Asterisks show results where all persuadees are convinced.

	p-num	1		1		1		2		2		2		5		5		5	
	p-size	25%		50%		75%		25%		50%		75%		25%		50%		75%	
struct	af-size	Ma	Ra	Ma	Ra	Ma	Ra	Ma	Ra	Ma	Ra	Ma	Ra	Ma	Ra	Ma	Ra	Ma	Ra
		Mb	Rb	Mb	Rb	Mb	Rb	Mb	Rb	Mb	Rb	Mb	Rb	Mb	Rb	Mb	Rb	Mb	Rb
Ladder	11	**1.0***	0.2	**1.0***	0.08	0.0	0.0	**0.0***	-1.03	**0.0**	-0.5	**0.0**	-0.47	**2.0**	-0.61	**1.0**	-1.29	0.0	0.0
		1.0*	1.0*	1.0*	1.0*	0.0	0.0	0.0*	0.0*	0.0	0.0	0.0	0.0	2.0	2.0	1.0	1.0	0.0	0.0
Ladder	21	**0.0***	-0.58	0.0	0.0	0.0	0.0	**1.0***	-0.48	**0.0***	-1.08	0.0	0.0	**0.0***	-2.87	**2.0**	-0.19	**0.0**	-0.56
		0.0*	0.0*	0.0	0.0	0.0	0.0	1.0*	1.0*	0.0*	0.0*	0.0	0.0	0.0*	0.0*	2.0	2.0	0.0	0.0
Ladder	51	**0.0***	-0.54	**0.0***	-0.63	**0.0***	-0.53	**2.0***	0.28	**0.0**	-0.48	**0.0***	-0.92	**2.0***	-1.32	**1.0**	-0.45	0.0	0.0
		0.0*	0.0*	0.0*	0.0*	0.0*	0.0*	2.0*	2.0*	0.0	0.0	0.0*	0.0*	2.0*	2.0*	1.0	1.0	0.0	0.0
Cycle	11	**1.0***	0.01	0.0	0.0	0.0	0.0	**1.0**	0.04	0.0	0.0	0.0	0.0	**1.0**	-1.35	0.0	0.0	0.0	0.0
		1.0*	1.0*	0.0	0.0	0.0	0.0	1.0	1.0	0.0	0.0	0.0	0.0	**1.0**	0.0	0.0	0.0	0.0	0.0
Cycle	21	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	**1.0**	0.0	0.0	0.0	**1.7**	-0.85	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	**1.0**	0.0	0.0	0.0	**2.0**	0.0	0.0	0.0	0.0	0.0
Cycle	51	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Tree	11	**0.0***	-0.12	**0.0***	-0.5	0.0	0.0	**2.0***	0.51	**1.0***	-0.49	0.0	0.0	**2.0***	0.99	**2.0**	0.1	**1.0**	0.16
		0.0*	0.0*	0.0*	0.0*	0.0	0.0	2.0*	2.0*	1.0*	1.0*	0.0	0.0	2.0*	2.0*	2.0	2.0	1.0	1.0
Tree	21	**0.0***	-0.88	0.0	0.0	**0.0***	-0.75	**0.0**	-0.63	**2.0***	0.52	**0.0**	-0.44	**0.0**	-1.56	**2.0***	1.1	**0.0**	-2.19
		0.0*	0.0*	0.0	0.0	0.0*	0.0*	0.0	0.0	2.0*	2.0*	0.0	0.0	0.0	0.0	2.0*	2.0*	0.0	0.0
Tree	51	**0.0***	-0.67	0.0	0.0	0.0	0.0	**1.0**	0.16	0.0*	0.0*	0.0	0.0	**1.0***	0.34	**3.0**	0.17	0.0	0.0
		0.0*	0.0*	0.0	0.0	0.0	0.0	1.0	1.0	0.0*	0.0*	0.0	0.0	1.0*	1.0*	**3.0**	2.0	0.0	0.0

Table 6.2: Average and best effectiveness of solutions found by MDEO (respectively, Ma, top left, Mb, bottom left) and by RA (respectively, Ra, top right, and Rb, bottom right) for large games. The results in bold are the better performing approach for a game. Asterisks show results where all persuadees are convinced.

	p-num	10		10		10		50		50		50		100		100		100	
	p-size	25%		50%		75%		25%		50%		75%		25%		50%		75%	
struct	af-size	Ma	Ra	Ma	Ra	Ma	Ra	Ma	Ra	Ma	Ra	Ma	Ra	Ma	Ra	Ma	Ra	Ma	Ra
		Mb	Rb	Mb	Rb	Mb	Rb	Mb	Rb	Mb	Rb	Mb	Rb	Mb	Rb	Mb	Rb	Mb	Rb
Ladder	51	**1.0**	-4.59	**2.6**	-1.55	0.0	**0.06**	**7.0**	-15.93	**8.8**	-10.25	**4.0**	-4.61	**12.9**	-28.44	**15.8**	-21.28	**12.8**	-10.0
		1.0	1.0	**3.0**	2.0	0.0	**1.0**	7.0	7.0	**10.0**	7.0	**5.0**	1.0	13.0	13.0	**17.0**	10.0	**14.0**	5.0
Ladder	101	**1.0**	-4.23	**3.2**	0.03	**4.1**	-0.81	**10.6**	-16.16	**10.0**	-8.85	**4.8**	-8.26	**19.9**	-27.27	**7.7**	-21.74	**12.4**	-5.78
		1.0	1.0	**4.0**	3.0	**5.0**	4.0	**11.0**	10.0	**10.0**	9.0	**6.0**	2.0	**22.0**	21.0	**9.0**	7.0	**13.0**	11.0
Ladder	201	**3.0**	-2.81	**0.1**	-2.09	**1.0**	-1.28	**7.4**	-17.34	**7.8**	-8.73	**0.2**	-6.13	**20.0**	-31.48	**18.7**	-17.39	**9.9**	-8.75
		3.0	3.0	1.0	1.0	1.0	1.0	**8.0**	7.0	**10.0**	8.0	**2.0**	1.0	20.0	20.0	**21.0**	16.0	**12.0**	7.0
Cycle	51	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	**1.9**	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	**2.0**	0.0	0.0	0.0	0.0	0.0
Cycle	101	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Cycle	201	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Tree	51	**4.0**	-0.18	**1.0**	-2.68	**1.0**	-0.63	**4.0**	-14.78	**6.2**	-15.66	**1.0**	-7.37	**35.0**	-17.85	**27.0**	-4.8	**0.0**	-0.67
		4.0	4.0	1.0	1.0	1.0	1.0	4.0	4.0	8.0	8.0	1.0	1.0	35.0	35.0	27.0	27.0	0.0	0.0
Tree	101	**1.0**	-4.44	**2.0**	-3.5	**4.0***	1.75	**11.0**	-11.69	**5.0**	-1.3	**1.0**	-5.56	**20.0**	-20.23	**36.0**	-22.96	**7.0**	-9.68
		1.0	1.0	**2.0**	1.0	4.0*	4.0*	11.0	11.0	**5.0**	3.0	1.0	1.0	**20.0**	7.0	**36.0**	25.0	7.0	7.0
Tree	201	**5.0**	-2.71	**5.6**	2.71	**6.2**	1.11	**25.0**	-9.37	**8.0**	-4.98	**8.3**	-5.11	**50.0**	-12.13	**42.0**	-11.87	**7.0**	-3.98
		5.0	4.0	6.0*	6.0*	**7.0**	6.0	**25.0**	21.0	**8.0**	4.0	**9.0**	3.0	**50.0**	23.0	**42.0**	29.0	**7.0**	3.0

Table 6.3: Comparison of average time taken by MDEO (M Time, in HH:MM:SS:ms) with time taken by BF for large games. The faster approach is in bold. N/A indicates that the solution took longer than 24 hours to find.

	p-num	1	1	1	2	2	2	5	5	5
	p-size	25%	50%	75%	25%	50%	75%	25%	50%	75%
struct	af-size	M Time	M Time	M Time	M Time	M Time	M Time	M Time	M Time	M Time
		BF Time	BF Time	BF Time	BF Time	BF Time	BF Time	BF Time	BF Time	BF Time
Ladder	11	00:00:12	00:00:11	00:00:13	00:00:20	00:00:23	00:00:22	00:00:52	00:00:55	00:00:52
		00:00:00	**00:00:02**	**00:00:06**	**00:00:00**	**00:00:13**	**00:00:13**	**00:00:31**	**00:00:32**	**00:00:32**
Ladder	21	00:00:12	**00:00:12**	**00:00:12**	00:00:22	00:00:22	**00:00:22**	00:00:53	**00:00:54**	**00:00:58**
		00:00:00	01:54:49	01:56:35	**00:00:00**	**00:00:00**	03:54:35	**00:00:00**	18:36:18	18:58:09
Ladder	51	**00:00:13**	**00:00:14**	**00:00:15**	**00:00:24**	**00:00:27**	**00:00:26**	**00:00:54**	**00:00:59**	**00:01:05**
		N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Cycle	11	00:00:11	00:00:12	00:00:13	00:00:22	00:00:21	00:00:23	00:00:52	00:00:51	00:00:58
		00:00:01	**00:00:07**	**00:00:05**	**00:00:13**	**00:00:15**	**00:00:15**	**00:00:41**	**00:00:36**	**00:00:41**
Cycle	21	**00:00:11**	**00:00:12**	**00:00:13**	**00:00:22**	**00:00:22**	**00:00:24**	**00:00:53**	**00:00:55**	**00:00:56**
		03:36:53	03:32:46	03:56:49	06:45:16	07:17:00	07:40:23	10:02:09	18:41:30	13:20:00
Cycle	51	**00:00:13**	**00:00:14**	**00:00:15**	**00:00:24**	**00:00:26**	**00:00:29**	**00:00:57**	**00:01:03**	**00:01:03**
		N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Tree	11	00:00:12	00:00:12	**00:00:11**	00:00:22	00:00:21	**00:00:23**	00:00:52	**00:00:52**	**00:00:52**
		00:00:00	**00:00:00**	00:00:25	**00:00:01**	**00:00:03**	00:00:50	**00:00:04**	00:02:10	00:02:10
Tree	21	00:00:12	**00:00:12**	00:00:13	**00:00:22**	00:00:23	**00:00:25**	**00:00:53**	00:00:55	**00:00:53**
		00:00:01	07:12:59	**00:00:00**	07:24:38	**00:00:06**	13:57:47	19:45:43	**00:00:05**	18:54:44
Tree	51	**00:00:12**	**00:00:14**	**00:00:15**	**00:00:24**	**00:00:25**	**00:00:27**	**00:00:56**	**00:00:57**	**00:01:04**
		N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A

persuader strategies that maximise the number of convinced persuadees. Our evaluation demonstrates that the approach produces strategies that are effective, and that it does so efficiently even for large and complex scenarios. Further, we have shown how MDEO can be adjusted to a multi-objective problem, in which the persuader minimises the number of arguments asserted, while maximising the number of convinced persuadees. Given their performance, it is likely that alternate approaches that attempt to find optimal dialogue strategies would not be able to scale to multiple persuadees.

A key advantage of this SBME approach is that the high-level metamodel which encodes multi-audience persuasion games is easy to interpret and to adjust to other types of strategic argumentation problems. Having demonstrated here the potential of SBME for solving strategic argumentation problems, we plan to apply SBME techniques to other settings, such as those in which persuadees are able to respond to assertions of the persuader, and in which the persuader has a probabilistic model of the persuadees' AFs.

Currently, we have only explored scenarios in which persuadees have random initial beliefs, however we could investigate how the similarity of their initial beliefs affects the dynamics of the game; it has been shown that the similarity of participants beliefs can have surprising effects on the outcome of deliberation dialogues [78].

Table 6.4: Results for MDEO and RA on competition frameworks (BA:Barabási-Albert, PP:Planning-problem, AB:Assumption-based). *HV* shows average hypervolume, *eff* shows effectiveness (best and average). Times shown are averages. N/A indicates the solution took longer than 24 hours to find.

		Single Objective			Multi Objective		
		MDEO time	MDEO eff	RA eff	MDEO time	MDEO HV	RA HV
BA(160)	Avg	00:15:31.438	**10**	-2.64	00:15:04.175	**0.20**	0
	Best		**10**	1			
PP(480)	Avg	01:04:00.672	**34**	9.86	01:00:16.931	0.640	**0.649**
	Best		34	34			
AB(691)	Avg	N/A	N/A	0	N/A	N/A	0
	Best		N/A	0			

Another avenue of future work is the implementation of more specific mutation operators that can select arguments which have a higher chance of increasing the strategy effectiveness. For example, we could use a heuristic that estimates the utility of asserting a particular argument and specify the total utility as an additional objective to be maximised [79]. We are also interested in exploring if we can reduce the search time by trimming from the search space the arguments that do not support the topic.

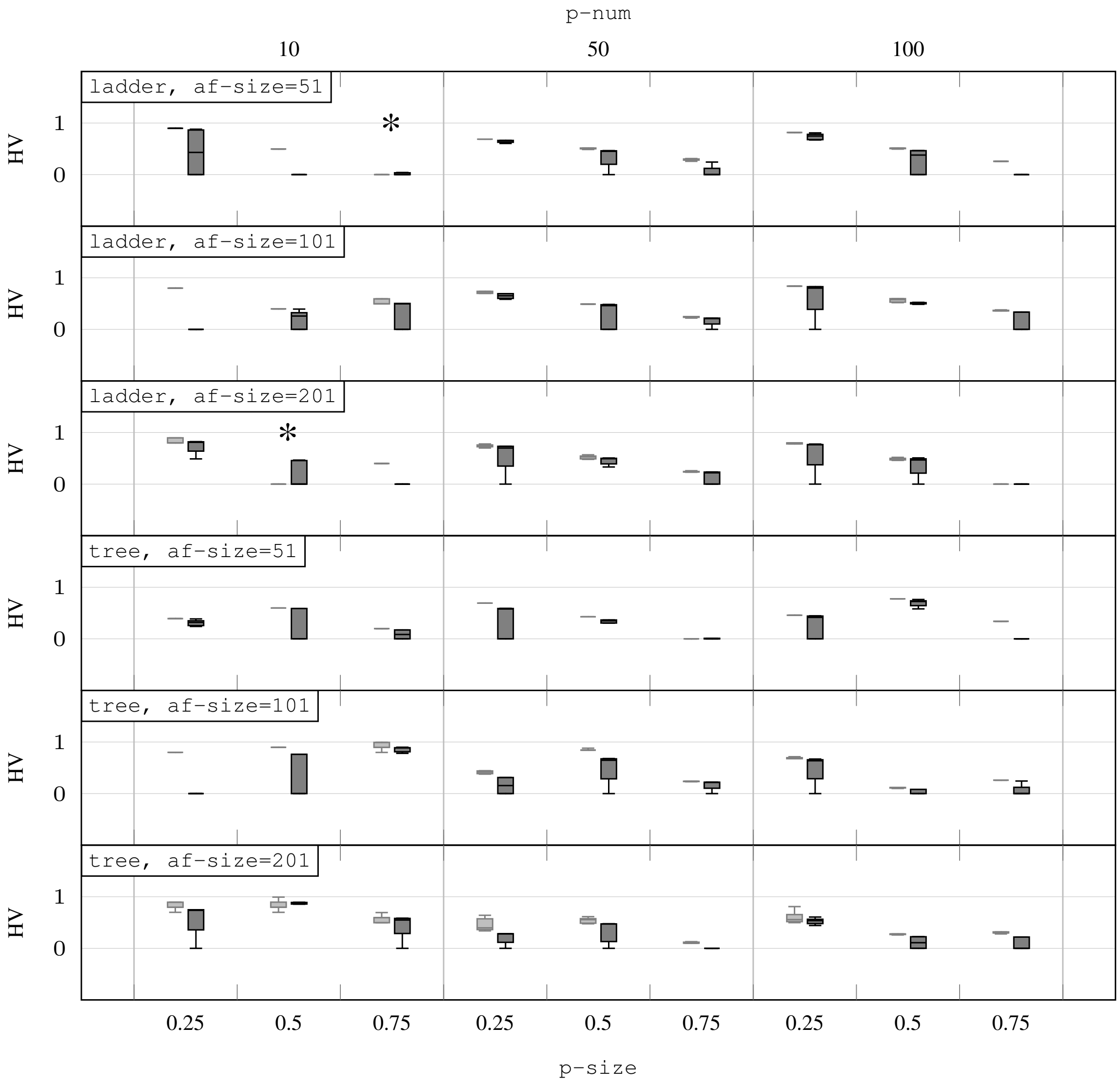


Figure 6.5: Multi-objective performance of MDEO and RA. Ticks on the top x axis shows number of persuadees in the scenario; bottom x axis shows number of arguments known to each persuadee (as a proportion of af-size). The size of the persuader's AF and the graph structure are included in the top left corner of each row. For each comparison, the light gray box plot on the left shows the spread of HVs obtained by MDEO and the dark gray box plot on the right shows the spread of HVs obtained by RA.

Chapter 7

Conclusions

7.1 Introduction

This chapter is structured as follows. In Section 7.2, a summary of the thesis is provided, highlighting the major contributions of the work. This is followed by an analysis of the limitations of our research in Section 7.3, and then a discussion about some potential directions for future work in Section 7.4. Finally, in Section 7.5, the chapter concludes with some closing remarks.

7.2 Summary and contributions

In this thesis we have been concerned with the performance of dialogue systems. We have investigated to what extent the performance of dialogue systems are affected by their domains, and furthermore we have proposed two approaches to strategic reasoning in dialogue systems that are efficient in their performance. We summarise the contributions below.

7.2.1 The impact of domain on dialogues

We have investigated the relationship of domain and the performance of dialogue systems. By simulating deliberation dialogues, in which agents argue over a set of possible actions to take and the values that the actions can promote or demote, we found that the specific properties of the similarity of participants' initial beliefs can have surprising correlations with the outcome of dialogue. Counter to our intuition, the results showed that the more similar participants' starting beliefs the less likely they are to come to an agreement.

We built a set of benchmarks of randomly generated argumentation frameworks, constructed around three types of framework: Dung argumentation frameworks and two popular generalisations, extended argumentation frameworks and collective-attack frameworks. From these generated structures, we measured their emergent semantic-based properties, specifically properties are known to affect the performance of argumentation systems. Specifically, we measured the size of extensions, the resistance, and the proportion of subsets of the framework in which a topic argument is acceptable. Furthermore, we investigated these properties in two case studies of real-world applications of argumentation that use structures similar to some of those we randomly generated. The underlying structure of graph was found to be a strong indicator of these properties in both the randomly generated graph as well as the case studies.

We conclude that domain and framework structure have profound, and often unpredictable, relationships with the behaviour of dialogue systems. Thus, evaluations of dialogue systems should consider a range of possible domains in their evaluations, and ideally use multiple types of framework structure, or use structures derived from real-world sources.

7.2.2 Efficient approaches to strategy in persuasion dialogues

Two approaches to strategic reasoning in persuasion dialogues have been presented in this thesis. In particular, unlike previous approaches to generating dialogue strategies for persuasion dialogues, we have proposed approaches that are not guaranteed to return an optimal strategy. Instead, the approaches are able to produce strategies more efficiently, using fewer computational resources. To do this, we have used approximate methods: we designed a heuristic for persuasion (Chapter 5), as well as using evolutionary search to find an acceptable strategy (Chapter 6).

By building simulation environments for the dialogues investigated, we were able to rigorously evaluate how well the approaches performed, both in terms of computational efficiency as well as effectiveness. Our evaluations were performed over different types of domain by varying the structure of framework used in the simulation. We have shown that our approaches have been able to scale to domains far larger than what is possible with approaches that seek to determine an optimal strategy. Furthermore, we have demonstrated that our approaches outperform simple baseline alternatives in terms of effectiveness.

Due to the efficiency of the evolutionary search approach, we were able to simulate dialogues that were strategically more complex than typical one-to-one persuasion dialogues. We were able to consider one-to-many persuasion dialogues in which the persuader attempts to convince as many persuadees as possible while still using the same assertions. Our approach was able to scale to scenarios with up to 100 persuadees. We have also considered dialogues in which there is a cost attached to asserting arguments, and so the persuader has the additional objective to minimise this cost when selecting which arguments to put forward; this consideration of the additional objective had very little impact on the performance of our approach.

7.2.3 Contributions

We highlight the key contributions from the summary above.

- We have created a simulation environment for deliberation dialogues, in which agents argue about the best joint action to take. The result of our simulations have shown a surprising relationship between the similarity of the initial beliefs of participants and the likelihood of an agreement being reached between them when engaging in a deliberation dialogue.

- We have analysed the impact that framework structures derived from generalised frameworks can have on a number of key properties that determine the performance of dialogue systems.

- We have provided a heuristic for estimating the benefit of asserting an argument in a persuasion dialogue, which we have demonstrated to suitable for determining efficient and effective dialogue strategies to sizes of domain beyond that which optimal approaches can cope with.

- We have provided an approach to generating persuasion dialogue strategies using evolutionary search. We have shown that the approach performs well in one-to-many persuasion dialogues, and dialogues in which the persuader are multiple objectives to consider when evaluating the utility of a strategy.

7.3 Critical assessment and limitations

In addition to their approximate natures, our approaches to strategic reasoning suffer from some limitations, which are discussed below.

7.3.1 Relevance to human dialogues and reasoning

In the wider field of argumentation, limited work has been done to investigate whether formal argumentation theory is relevant to how humans reason, and to what extent it is descriptive not just normative. While the general dialectical principles of argumentation appear to be intuitive to humans, thorough empirical studies investigating the link between formal argumentation and human psychology are rare. In the context of dialogue systems, including those studied in this thesis, it is likely that our models and tools are far removed from the complexities of human dialogues and reasoning to be adequately captured.

Rahwan *et al.* explore the key concept of reinstatement in argumentation, and whether the results arrived at by applying argumentation semantics to a reinstatement situation relates to the way humans comprehend such a situation [91]. Their results show that although reinstatement of arguments is a concept that humans use in their reasoning, humans do not recognise a completely renewed acceptance of the reinstated argument as occurs in argumentation theory. With a focus on arguments constructed from natural language, Cerutti *et al.* examine to what extent human's evaluation of an arguments' acceptability agrees with that which is determined by argumentation theory: across three different domains, they found that humans match what the theory predicts in most cases, but there are exceptions [29]. Rosenfeld and Kraus evaluate their approach to applying machine learning to strategic reasoning by investigating how effective it is at persuading humans [95]. Their work is at the forefront of bridging the gap between human dialogue and machine dialogue, which could pave the way to the widespread use of argumentation technology in practical applications.

However, in this thesis, we have not tried to incorporate human comprehension or human persuasion in our approaches by explicitly addressing the way they reason. Nor have we used any human studies to evaluate the performance or effectiveness of the dialogue strategies proposed. Thus, our results are only valid in human domains on the assumption that argumentation theory is indeed an accurate representation of practical human reasoning.

7.3.2 Simplistic dialogue models

The dialogue models we investigated in this thesis are simplistic examples of dialogues. Other approaches to strategic argumentation in persuasion dialogues have investigated richer models. Some alternative approaches have considered opponent modelling, where the persuader has probabilistic knowledge of which arguments are known by their opponent which they can use when determining arguments to put forward (e.g., [57, 16, 50, 92]). Approaches have also captured more sophisticated persuadees, who may themselves assert arguments in response to those arguments put forward by the persuader, creating a more complex dynamic in the dialogue, where the persuader must respond to counter arguments that are put forward (e.g. [16, 50, 92]). Black *et al.*'s model allows for the arguments put forward by the persuader to induce new arguments within the persuadee's knowledge base, and so the persuader has to consider what knowledge may be revealed by their assertions and whether that will be advantageous to the persuasion or not [16].

In contrast, in this thesis, we have considered only simple dialogues in which persuaders have exact knowledge of the persuadees' arguments, and persuadees do not themselves assert arguments. One way in which we have considered more complex dialogues is in Chapter 6, where the dialogue model allows for there to be multiple persuadees, as opposed to typical models in which there is only a single persuadee. However, the dynamics of the multi-persuadee dialogue that we model remain straightforward in that persudees do not assert any arguments.

7.3.3 Reassurance of quality

The empirical evaluations of our proposed approximate approaches have shown that the strategies they produce perform well when compared to naive alternatives. However, given an individual solution produced by these approximate approaches, we may want to know how effective it is in comparison to the optimal solution. For example, the evolutionary search approach when used in a one-to-many dialogue may produce a strategy that convinces 20% of persuadees — this may be an optimal solution, or it may be a very poor result if the optimal solution can find a solution that convinces 90% of persuadees. Our approaches currently offer no reassurance of the quality of their proposed solutions.

Without computing the optimal solution, there is no way to tell for certain how close to optimal a given solution is. For some of the simple dialogues we tested, we were able to compare how the quality of the solution compared to an optimal solution by finding it through brute force search. However, for many of the dialogues we simulated, calculating the optimal solution is computationally intractable using current technology, and so such a comparison is impossible.

It may be possible to provide some estimate of the quality of the solutions produced by our approaches. In the case of the evolutionary search, we can use the proportion of the search space that has been explored (the more search space is explored, the more certain we can be the solution is of a high-quality), or the number of generations in which the best found solution did not change (if, by the end of the search, new generations are still producing better solutions then it is unlikely we have found an optimal solution). It is less clear how an estimate of quality could be computed in the case of the heuristic strategy.

7.4 Future work

7.4.1 Combining the heuristic and evolutionary strategies

The evolutionary algorithm presented in Chapter 6 can be refined in two directions. The first direction is to attempt to trim the search space, by preventing the generation of candidate strategies that we can definitely rule out. The second direction is to use a heuristic to guide the search, so that it is able to converge more quickly to effective and efficient strategies.

With regard to the first direction, it is possible to identify arguments that we would never want to assert in a persuasion setting. For example, an argument that directly attacks the topic argument will never be beneficial to assert. We can therefore, when generating new populations, use more sophisticated mutation operators that will avoid adding such arguments. This would reduce the search space, and thus would potentially increase the speed of the approach. However, identifying the arguments to avoid, and executing more complex mutation operators, would come at some increase in computational overhead. Further experiments would be required to find out whether the trade off is always worthwhile, and if not, whether there are any particular circumstances in which it is.

The heuristic proposed in Chapter 5 would seem to be a good starting point for investigating the second direction of whether the use of a heuristic to guide the evolutionary search would be beneficial. Since the heuristic provides an estimate of to what extent an argument attacks or defends the topic argument, it can be used as an additional optimisation function to increase the rate of convergence of the evolutionary search. The guidance is likely to be more helpful in earlier generations, where the average utility of the population is low. As the population converges on a minimum, the heuristic becomes less useful, and so it could be phased out at some point in the search (either after a number of generations, or after the rate of improvement of solutions starts to slow down). However, as with the search space trimming, introducing the heuristic to the evolutionary search comes at an increased computational cost, and further investigations are needed to see how beneficial the approach is.

7.4.2 Evaluating real structures of argumentation framework

In this thesis we have investigated the impact that the structure of argumentation framework used by a dialogue system has an impact on the system's performance. In our experiments, we used randomly generated frameworks with particular characteristics that have been used in practical applications of argumentation. This is only a first step in understanding the effect of structures. Due to developments in the field of argument mining, it is becoming increasingly feasible to obtain structures of argumentation framework from real-world sources, and thus it will be possible to investigate what emergent properties these structures have. This will ultimately allow us to design dialogue systems that perform well on the kinds of structures that exist in realistic domains.

7.4.3 Dialogue strategy competition

A difficulty with evaluating the performance of an approach to generating dialogue strategies is that it is not possible to make a direct comparison with other approaches. This is because each approach assumes a different protocol for the dialogue, or different modelling of the persuadee(s) by the persuader, or a different distribution of arguments amongst participants. As a result, no two approaches to generating dialogue strategies have been evaluated on exactly the same type of dialogue.

A dialogue strategy competition, similar to the argument solver competitions, could provide a set of standardised strategic problems to which to evaluate the performance of different approaches. This would alleviate the issue above, and allow for detailed benchmarking between approaches in a systematic way. As a result, this would motivate the development of more efficient and effective dialogue strategy generators. However, as with the argument solver competitions, designing and organising the competition would be non-trivial. The main challenges for a dialogue competition are listed below.

- Unlike the argument solver competition, where problems are already well-defined in argumentation theory, there are no similarly canonical dialogue problems. A set of dialogue problems would need to be designed and decided upon. Which type(s) of dialogue should be included? Which objectives for dialogue success should be used? How should other dialogue participants behave in response? Should the persuader have models of other participants, what information should the models represent, and how accurate should they be? Should competitors argue against one another, or should they argue against standardised participants? It is not obvious what the answers to these questions should be. Any choice will, to some extent, bias the competition towards certain approaches. Therefore, a consensus should be reached within the community, with some consideration of which problems have the most application or relevance to real-world applications.

- Once the set of problems has been decided upon, there would still need to be a generation of specific instances of these problems. In the argument solver competitions, this relates to generating an AF on which to run the solver upon. In the dialogue setting, AFs would also be a central component and so they would have to be generated. But furthermore, additional parameters are needed to define a dialogue problem, such as which arguments are known by each participant at the start of the dialogue and which argument is the topic argument.

- A standardised format for solutions and problems would be required. Regarding solutions, the simplest option would be the have competitors output a file for later analysis. But what should the syntax of such input and output files look like? There would need to be a consideration on both the size of a file (for the practical reason of having to potentially store many of them) and the computational neutrality of the format (that is, it should, as much as possible, avoid being easier for specific approaches).

- A practical consideration are where the computational resources required to run the problems will come from. Further, the processing and analyse of the results would be non-trivial, and so there would be a significant amount of human resources to organise the competition.

Nevertheless, despite these hurdles, a competition could do much to improve the evaluation of the performances of current approaches to strategic reasoning in dialogue systems.

7.5 Closing remarks

The work presented in this thesis contributes to research on dialogue systems in argumentation. Specifically, we have demonstrated that small changes in the domain of dialogue can have significant and surprising effects on the performance of argumentation systems. Further, we have proposed novel approaches to strategic problems in persuasion dialogues, and we have demonstrated that they are computationally efficient. By designing our approaches to be approximate rather than optimal, we have been able to apply our approaches to dialogues and domains that are beyond that which current approaches could likely cope with.

In order to realise real-world applications of argumentation in dialectical settings it is vital to consider the practical issues of the performance of dialogue systems. Systems that are able to explain their reasoning in a way that is intuitive to non-expert human users are likely to become more prevalent as demand for scrutability and understanding of artificial intelligence rises. The contributions of this thesis are a step in the development of argumentation systems potentially becoming a key technology for explainable AI.

Bibliography

[1] L. Amgoud and S. Vesic. On revising argumentation-based decision systems. In *Proceedings of the 10th European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty*, pages 71–82. Springer, 2009.

[2] Aristotle. *Rhetoric*. Modern Library, 1954.

[3] Aristotle. *Topics. Books I and VIII.* Oxford University Press, 1997.

[4] K. Atkinson and T. Bench-Capon. Practical reasoning as presumptive argumentation using action based alternating transition systems. *Artificial Intelligence*, 171(10–15):855–874, 2007.

[5] K. Atkinson, T. Bench-Capon, and P. McBurney. PARMENIDES: Facilitating deliberation in democracies. *Artificial Intelligence and Law*, 14(4):261–275, 2006.

[6] K. Atkinson, T. Bench-Capon, and S. Modgil. *Argumentation for decision support*, pages 822–831. Springer, 2006.

[7] P. Baroni, M. Caminada, and M. Giacomin. An introduction to argumentation semantics. *The Knowledge Engineering Review*, 26(4):365–410, 2011.

[8] S. Beiker. Legal aspects of autonomous driving. *Santa Clara Law Review*, 52(4):1145–1158, 2012.

[9] T. Bench-Capon. Argument in artificial intelligence and law. *Artificial Intelligence and Law*, 5(4):249–261, 1997.

[10] T. Bench-Capon. Agreeing to differ: Modelling persuasive dialogue between parties without a consensus about values. *Informal Logic*, 22(3):231–245, 2002.

[11] T. Bench-Capon, S. Doutre, and P. Dunne. Audiences in argumentation frameworks. *Artificial Intelligence*, 171(1):42 – 71, 2007.

[12] M. Benlamine, S. Villata, R. Ghali, C. Frasson, F. Gandon, and E. Cabrio. Persuasive argumentation and emotions: An empirical evaluation with users. In M. Kurosu, editor, *Human-Computer Interaction: User Interface Design, Development and Multimodality*. Springer, 2017.

[13] S. Bistarelli, F. Rossi, and F. Santini. A comparative test on the enumeration of extensions in abstract argumentation. *Fundamenta Informaticae*, 140(3-4):263–278, 2015.

[14] E. Black and K. Atkinson. Choosing persuasive arguments for action. In *Proceedings of the 10th International Conference on Autonomous Agents and Multi-Agent Systems*, pages 905–912, 2011.

[15] E. Black, A. J. Coles, and S. Bernardini. Automated planning of simple persuasion dialogues. In *Proceedings of the 15th International Workshop on Computational Logic in Multi-Agent Systems*, pages 87–104. Springer, 2014.

[16] E. Black, A. J. Coles, and C. Hampson. Planning for persuasion. In *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems*, pages 933–942. International Foundation for Autonomous Agents and Multiagent Systems, 2017.

[17] E. Black and A. Hunter. An inquiry dialogue system. *Autonomous Agents and Multi-Agent Systems*, 19(2):173–209, 2009.

[18] E. Black and A. Hunter. Reasons and options for updating an opponent model in persuasion dialogues. In E. Black, S. Modgil, and N. Oren, editors, *Theory and Applications of Formal Argumentation*, pages 21–39. Springer, 2015.

[19] N. Bostrom and E. Yudkowsky. The ethics of artificial intelligence. In K. Frankish and W. Ramsey, editors, *The Cambridge handbook of artificial intelligence*, pages 316–334. Cambridge University Press, 2014.

[20] I. Boussaïd, P. Siarry, and M. Ahmed-Nacer. A survey on search-based model-driven engineering. *Automated Software Engineering*, 24(2):233–294, 2017.

[21] G. Brewka, S. Polberg, and S. Woltran. Generalizations of dung frameworks and their role in formal argumentation. *IEEE Intelligent Systems*, 29(1):30–38, 2014.

[22] R. Brochenin, T. Linsbichler, M. Maratea, J. Wallner, and S Woltran. Abstract solvers for Dung's argumentation frameworks. In *Proceedings of 3rd International Workshop on Theory and Applications of Formal Argumentation*, pages 40–58. Springer, 2016.

[23] B. Buchanan and E. Shortliffe. *Rule Based Expert Systems: The Mycin Experiments of the Stanford Heuristic Programming Project*. Addison-Wesley, 1984.

[24] M. Caminada. A discussion game for grounded semantics. In E. Black, S. Modgil, and N. Oren, editors, *Theory and Applications of Formal Argumentation*, pages 59–73. Springer, 2015.

[25] M. Caminada and Y. Wu. An argument game for stable semantics. *Logic Journal of the IGPL*, 17(1):77–90, 2009.

[26] D. Cartwright and K. Atkinson. Using computational argumentation to support e-participation. *IEEE Intelligent Systems*, 24(5):42–52, 2009.

[27] C. Cayrol, F. de Saint-Cyr, and M. Lagasquie-Schiex. Change in abstract argumentation frameworks: Adding an argument. *Journal of Artificial Intelligence Research*, 38:49–84, 2010.

[28] F. Cerutti, N. Oren, H. Strass, M. Thimm, and M. Vallati. A benchmark framework for a computational argumentation competition. In *Proceedings of the 5th International Conference on Computation Models of Argument*, pages 459–460. IOS Frontiers in AI and Applications, 2014.

[29] F. Cerutti, N. Tintarev, and N. Oren. Formal arguments, preferences, and natural language interfaces to humans: An empirical evaluation. In *Proceedings of the 21st European Conference on Artificial Intelligence*, ECAI'14. IOS Press, 2014.

[30] G. Charwat, W. Dvorak, S. Gaggl, J. Wallner, and S. Woltran. Methods for solving reasoning problems in abstract argumentation: A survey. *Artificial Intelligence*, 220:28 – 63, 2015.

[31] F. Delecroix, M. Morge, and J.C. Routier. A virtual selling agent which is proactive and adaptive. In *Proceedings of the 10th international conference on practical applications of agents and multi-agent systems*, pages 57–66. Springer, 2012.

[32] F. Dignum and G. Vreeswijk. Towards a testbed for multi-party dialogues. In F. Dignum, editor, *Advances in Agent Communication*, pages 212–230. Springer, 2004.

[33] P. M. Dung. On the Acceptability of Arguments and its Fundamental Role in Nonmonotonic Reasoning, Logic Programming and n-Person Games. *Artificial Intelligence*, 77(2):321–357, 1995.

[34] P. Dunne and M. Wooldridge. *Complexity of Abstract Argumentation*, pages 85–104. Springer, 2009.

[35] A. E. Eiben and J. E. Smith. *Introduction to Evolutionary Computing*. Springer, 2nd edition, 2015.

[36] M. Giacomin F. Cerutti, M. Vallati. On the effectiveness of automated configuration in abstract argumentation reasoning. In *Proceedings of the 6th International Conference on Computational Models of Argument*, pages 199–206. IOS Press, 2016.

[37] M. Giacomin F. Cerutti, M. Vallati. Where are we now? state of the art and future trends of solvers for hard argumentation problems. In *Proceedings of the 6th International Conference on Computational Models of Argument*, pages 207–218. IOS Press, 2016.

[38] M. Giacomin F. Cerutti, M. Vallati. On the impact of configuration on abstract argumentation. *Internation Journal of Approximate Reasoning*, 92:120–138, 2018.

[39] M. Vallati F. Cerutti, M. Giacomin. Algorithm selection for preferred extensions enumeration. In S. Parsons, N. Oren, C. Reed, and F. Cerutti, editors, *Proceedings of the 5th International Conference on Computational Models of Argument*, pages 221–232. IOS Press, 2014.

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

[41] S. Frank. The common patterns of nature. *Journal of Evolutionary Biology*, 22(8):1563–1585, 2009.

[42] S. Gabbriellini and P. Torroni. Arguments in social networks. In *Proceedings of the 12th International Conference on Autonomous Agents and Multi-agent Systems*, AAMAS 2013, pages 1119–1120. International Foundation for Autonomous Agents and Multiagent Systems, 2013.

[43] S. A. Gaggl, T. Linsbichler, M. Maratea, and S. Woltran. The 2nd international competition on computational models of argumentation. `http://argumentationcompetition.org/2017`, 2017.

[44] J. Glazer and A. Rubinstein. Debates and Decisions: On a Rationale of Argumentation Rules. *Games and Economic Behavior*, 36(2):158–173, 2001.

[45] J. Glazer and A. Rubinstein. On Optimal Rules of Persuasion. *Econometrica*, 72(6):1715–1736, 2004.

[46] M. Grabmair and K. Ashley. Facilitating case comparison using value judgments and intermediate legal concepts. In *Proceedings of the 13th international conference on artificial intelligence and law*, pages 161–170. ACM, 2011.

[47] N. Green. Towards creation of a corpus for argumentation mining the biomedical genetics research literature. In *Proceedings of the 1st workshop on Argumentation Mining*, ACL, pages 11–18, 2014.

[48] J. Grimshaw. *Argument structure*. MIT Press, 1990.

[49] D. Gunning. Explainable Artificial Intelligence (XAI). `http://www.darpa.mil/program/explainable-artificial-intelligence`, 2017. Accessed: 2018-01-23.

[50] E. Hadoux, A. Beynier, N. Maudet, P. Weng, and A. Hunter. Optimization of probabilistic argumentation with markov decision models. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence*, pages 2004–2010, 2015.

[51] E. Hadoux and A. Hunter. Strategic sequences of arguments for persuasion using decision trees. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence*, pages 1128–1134. AAAI Press, 2017.

[52] C. Hamblin. Fallacies. *Methuen*, 1970.

[53] D. Hardman and D. Hardman. *Judgment and Decision making: Psychological Perspectives*, volume 11. John Wiley & Sons, 2009.

[54] A. Hunter. Making argumentation more believable. In *Proceedings of the 19th AAAI Conference on Artificial Intelligence*, pages 269–274, 2004.

[55] A. Hunter. Towards higher impact argumentation. In *Proceedings of the 19th AAAI Conference on Artificial Intelligence*, pages 275–280, 2004.

[56] A. Hunter. Probabilistic strategies in dialogical argumentation. In *Proceedings of the 8th International Conference on Scalable Uncertainty Management*, pages 190–202. Springer, 2014.

[57] A. Hunter. Modelling the persuadee in asymmetric argumentation dialogues for persuasion. In *Proceedings of the 24th International Conference on Artificial Intelligence*, pages 3055–3061. AAAI Press, 2015.

[58] A. Hunter. Towards a framework for computational persuasion with applications in behaviour change. *Argument and Computation*, 9(1):15 – 40, 2018.

[59] A. Hunter and M. Thimm. Optimization of dialectical outcomes in dialogical argumentation. *International Journal of Approximate Reasoning*, 78:73–101, 2016.

[60] A. Hunter and M. Williams. Aggregating evidence about the positive and negative effects of treatments. *Artificial Intelligence in Medicine*, 56(3):173 – 190, 2012.

[61] S. Kaci and L. Van Der Torre. Preference-based argumentation: Arguments supporting multiple values. *International Journal of Approximate Reasoning*, 48(3):730–751, 2008.

[62] M. Kessentini, P. Langer, and M. Wimmer. Searching models, modeling search: On the synergies of SBSE and MDE. In *Proceedings of the 1st International Workshop Combining Modelling and Search-Based Software Engineering*, pages 51–54, 2013.

[63] D. Knuth. *The Art of Computer Programming: Sorting and Searching*. Pearson Education, 1997.

[64] E. Kok, J. Meyer, H. Prakken, and G. Vreeswijk. Testing the benfits of structured argumentation in multi-agent deliberation dialogues. In *Proceedings of the 11th*

International Conference on Autonomous Agents and Multiagent Systems, pages 1411–1412, 2012.

[65] D. Kontarinis, E. Bonzon, N. Maudet, and P. Moraitis. Empirical evaluation of strategies for multiparty argumentative debates. In N. Bulling, L. van der Torre, S. Villata, W. Jamroga, and W. Vasconcelos, editors, *Computational Logic in Multi-Agent Systems*, pages 105–122. Springer, 2014.

[66] J. Lawrence and C. Reed. Aifdb corpora. In S. Parsons, N. Oren, C. Reed, and F. Cerutti, editors, *Proceedings of the 5th International Conference on Computational Models of Argument*, pages 465–466. IOS Press, 2014.

[67] B. Liao, L. Jin, and R. Koons. Dynamics of argumentation systems: A division-based method. *Artificial Intelligence*, 175(11):1790–1814, 2011.

[68] M. Luck, P. McBurney, and C. Preist. A manifesto for agent technology: Towards next generation computing. *Autonomous Agents and Multi-Agent Systems*, 9(3):203–252, 2004.

[69] G. Luo, F. Pu, Y. Chen, and Y. Hang. argumatrix argmat-sat. `https://sites.google.com/site/argumatrix/argmat-sat`. Accessed: 2018-04-02.

[70] P. McBurney and S. Parsons. Chance discovery using dialectical argumentation. In *Proceedings of the JSAI 2001 International Workshop on Chance Discovery*, pages 414–424. Springer, 2001.

[71] P. McBurney and S. Parsons. *Dialogue Game Protocols*, pages 269–283. Springer, 2003.

[72] P. McBurney and S. Parsons. Dialogue games for agent argumentation. In G. Simari and I. Rahwan, editors, *Argumentation in Artificial Intelligence*, pages 261–280. Springer, 2009.

[73] P. McBurney, R. Van Eijk, S. Parsons, and L. Amgoud. A dialogue game protocol for agent purchase negotiations. *Autonomous Agents and Multi-Agent Systems*, 7(3):235–273, 2003.

[74] R. Medellin-Gasque, K. Atkinson, T. Bench-Capon, and P. McBurney. Strategies for question selection in argumentative dialogues about plans. *Argument & Computation*, 4(2):151–179, 2013.

[75] S. Modgil. An argumentation based semantics for agent reasoning. In *Proceedings of 1st International Workshop on Languages, Methodologies and Development Tools for Multi-agent Systems*, pages 37–53. Springer, 2007.

[76] S. Modgil. Reasoning about preferences in argumentation frameworks. *Artificial Intelligence*, 173(9):901 – 934, 2009.

[77] S. Modgil, F. Toni, F. Bex, I. Bratko, C. Chesñevar, W. Dvořák, M. Falappa, et al. The added value of argumentation. In S. Ossowski, editor, *Agreement Technologies*, pages 357–403. Springer, 2013.

[78] J. Murphy, E. Black, and M. Luck. Arguing from similar positions: An empirical analysis. In *Proceedings of the 4th International Workshop on Theory and Applications of Formal Argumentation*, pages 177–193. Springer, 2015.

[79] J. Murphy, E. Black, and M. Luck. A heuristic strategy for persuasion dialogues. In *Computational Models of Argument*, volume Frontiers in Artificial Intelligence and Applications 287, pages 411 – 418. IOS Press, 2016.

[80] J. Murphy, A. Burdusel, M. Luck, S. Zschaler, and E. Black. Deriving peruasion strategies. In *Proceedings of the 7th International Conference on Computational Models of Argument*. To appear, 2018.

[81] J Murphy, I Sassoon, M Luck, and E. Black. An investigation of argumentation framework characteristics. In E. Black, S. Modgil, and N. Oren, editors, *Proceedings of the 4th International Workshop on Theory and Applications of Formal Argumentation*, pages 1–16. Springer, 2018.

[82] S. Nielsen and S. Parsons. An application of formal argumentation: Fusing bayes nets in mas. In *Proceedings of 1st International Conference on Computational Models of Argument*, pages 33–44. IOS Frontiers in AI and Applications, 2006.

[83] S. Nielsen and S. Parsons. A generalization of dung's abstract framework for argumentation: Arguing with sets of attacking arguments. In N. Maudet, S. Parsons, and I. Rahwan, editors, *Argumentation in Multi-Agent Systems*, pages 54–73. Springer, 2007.

[84] N. Oren, T. Norman, and A. Preece. Loose lips sink ships: A heuristic for argumentation. In Rahwan I. Parsons S. Maudet, N., editor, *Proceedings of the 3rd*

International Workshop on Argumentation in Multi-Agent Systems, pages 121–134, 2006.

[85] S. Polberg. Intertranslatability of abstract argumentation frameworks. Technical report, Institute for Information Systems, Technical University of Vienna, 2017.

[86] H. Prakken. Combining sceptical epistemic reasoning with credulous practical reasoning. In *Proceedings of the 2006 Conference on Computational Models of Argument: Proceedings of COMMA 2006*, pages 311–322. IOS Press, 2006.

[87] H. Prakken. Formal systems for persuasion dialogue. *The Knowledge Engineering Review*, 21(2):163–188, 2006.

[88] K. Budzynska R. Duthie and C. Reed. Mining ethos in political debate. In *Computational Models of Argument*, volume Frontiers in Artificial Intelligence and Applications 287, pages 299 – 310. IOS Press, 2016.

[89] I. Rahwan. Argumentation in multi-agent systems. *Autonomous Agents and Multi-Agent Systems*, 11:115–125, 2005.

[90] I. Rahwan and K. Larson. Mechanism design for abstract argumentation. In *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems*, pages 1031–1038. International Foundation for Autonomous Agents and Multiagent Systems, 2008.

[91] I. Rahwan, M. Madakkatel, J. F. Bonnefon, R. Awan, and S. Abdallah. Behavioral experiments for assessing the abstract argumentation semantics of reinstatement. *Cognitive Science*, 34(8):1483–1502, 2010.

[92] T. Rienstra, M. Thimm, and N. Oren. Opponent models with uncertainty for strategic argumentation. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence*, pages 332–338, 2013.

[93] O. Rodrigues. A forward propagation algorithm for the computation of the semantics of argumentation frameworks. In *Theory and Applications of Formal Argumentation*, pages 120–136. Springer, 2018.

[94] O. Rodrigues, E. Black, M. Luck, and J. Murphy. On structural properties of argumentation frameworks: Lessons from iccma. *CEUR Workshop Proceedings*, 2171:22–35, 2018.

[95] A. Rosenfeld and S. Kraus. Strategical argumentative agent for human persuasion. In *Proceedings of the 22nd European Conference on Artificial Intelligence*, pages 320–328. IOS Press, 2016.

[96] B. Russel and A. Whitehead. *Principia Mathematica.* Cambridge University Press, 1925.

[97] Brem S. and Rips L. Explanation and evidence in informal argument. *Cognitive Science*, 24(4):573–604, 2010.

[98] I. Sassoon, J. Keppens, and P. McBurney. Preferences in argumentation for statistical model selection. In *Proceedings of the 6th International Conference on Computational Models of Argument*, pages 53–60. IOS Frontiers in AI and Applications, 2016.

[99] J. Schmidhuber. Deep learning in neural networks: An overview. *Neural Networks*, 61:85 – 117, 2015.

[100] Zohreh Shams and Nir Oren. A two-phase dialogue game for skeptical preferred semantics. In *European Conference on Logics in Artificial Intelligence*, pages 570–576. Springer, 2016.

[101] H. Shin. The Burden of Proof in a Game of Persuasion. *Journal of Economic Theory*, 64:253–264, 1994.

[102] H. Shin. Adversarial and Inquisitorial Procedures in Arbitration. *RAND Journal of Economics*, 29:378–405, 1998.

[103] H. Sturges. The choice of a class interval. *Journal of the American Statistical Association*, 21(153):65–66, 1926.

[104] Y. Tang and S. Parsons. Argumentation-based dialogues for deliberation. In *Proceedings of the 4th International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 552–559. ACM, 2005.

[105] R. Teach and E. Shortliffe. An analysis of physician attitudes regarding computer-based clinical consultation systems. In J. Anderson and S. Jay, editors, *Use and Impact of Computers in Clinical Medicine*, pages 68–85. Springer, 1987.

[106] M. Thimm. Strategic argumentation in multi-agent systems. *Künstliche Intelligenz*, 28(3):159–168, 2014.

[107] M. Thimm. Strategic argumentation in multi-agent systems. *Künstliche Intelligenz, Special Issue on Multi-Agent Decision Making*, 28(3):159–168, 2014.

[108] M. Thimm. Tweety - A comprehensive collection of Java libraries for logical aspects of artificial intelligence and knowledge representation. In *Proceedings of the 14th International Conference on Principles of Knowledge Representation and Reasoning*, pages 528–537. AAAI Press, 2014.

[109] M. Thimm. Tweety - A comprehensive collection of Java libraries for logical aspects of artificial intelligence and knowledge representation. In *Proceedings of the 14th International Conference on Principles of Knowledge Representation and Reasoning*, pages 528–537. AAAI Press, 2014.

[110] A. Toniolo, T. Norman, and K. Sycara. An empirical study of argumentation schemes for deliberative dialogue. In *Proceedings of the 20th European Conference on Artificial Intelligence*, pages 756–761, 2012.

[111] J. Tremblay and I. Abi-Zeid. Value-based argumentation for policy decision analysis: methodology and an exploratory case study of a hydroelectric project in Quebec. *Annals of Operations Research*, 1:233–253, 2016.

[112] T. Uno. An algorithm for enumerating all directed spanning trees in a directed graph. In *International Symposium on Algorithms and Computation*, pages 166–173. Springer, 1996.

[113] G. Vreeswik and H. Prakken. Credulous and sceptical argument games for preferred semantics. In *European Workshop on Logics in Artificial Intelligence*, pages 239–253. Springer, 2000.

[114] D. Walton. *Argumentation Schemes for Presumptive Reasoning*. Lawrence Erlbaum Associates, 1996.

[115] D. Walton and E. Krabbe. *Commitment in dialogue: Basic concepts of interpersonal reasoning*. SUNY press, 1995.

[116] D. Walton, A. Toniolo, and T. Norman. Missing phases of deliberation dialogue for real applications. In *Proceedings of the 11th International Workshop on Argumentation in Multi-Agent Systems*. Springer, 2014.

[117] M. Wardeh, T. Bench-Capon, and F. Coenen. PADUA protocol: Strategies and tactics. In K. Mellouli, editor, *Symbolic and Quantitative Approaches to Reasoning with Uncertainty*, pages 465–476. Springer, 2007.

[118] M. Wooldridge. *An introduction to multiagent systems*. John Wiley & Sons, 2009.

[119] E. Yudkowsky. Artificial intelligence as a positive and negative factor in global risk. *Global catastrophic risks*, 1(303):184, 2008.

[120] B. Yun, S. Vesic, M. Croitoru, P. Bisquert, and R. Thomopoulos. A structural benchmark for logical argumentation frameworks. In N. Adams, A. Tucker, and D. Weston, editors, *Advances in Intelligent Data Analysis XVI*, pages 334–346. Springer, 2017.

[121] B Yun, S Vesic, M Croitoru, P Bisquert, and R Thomopoulos. A structural benchmark for logical argumentation frameworks. In *International Symposium on Intelligent Data Analysis*, pages 334–346. Springer, 2017.

[122] E. Zitzler and L. Thiele. Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach. *IEEE transactions on Evolutionary Computation*, 3(4):257–271, 1999.

[123] S. Zschaler and L. Mandow. Towards model-based optimisation: Using domain knowledge explicitly. In *Proceedings of the 1st Workshop on Model-Driven Engineering, Logic and Optimization*, 2016.