Computable Contracting and Contract Theory

A New Research Direction

Sajjandeep Sivia

Supervised by Prof. Christopher Clack and Dr. Manuela Dal Borgo

Candidate Number: FHVZ1
Module Code: BASC0024
University College London
May 2021

Computable Contracting and Contract Theory

A New Research Direction

Supervised by Prof. Christopher Clack and Dr. Manuela Dal Borgo

Abstract

Contracts securely align incentives between counterparties allowing them to exploit the surplus gained from cooperation. Computable contracting aims to create machine and human understandable contracts. Contract theory mathematically formalises contractual arrangements to investigate incentive issues and provide optimal solutions to contracting problems. Most research in computable contracting investigates how to make legal concepts machine expressible, but little work focuses on how incentives are affected by the creation of new contractual mechanisms nor on how these new contractual mechanisms can be used to solve perennial incentive problems within contracts. This paper seeks to introduce contract theorists to computable contracting and visa versa in order to stimulate interdisciplinary work between the two fields. Through a literature review of computable contracting and contract theory literature, this paper considers how the insights from contract theory and computable contracting can fruitfully interact to advance the research aims of both disciplines. The paper uses the MIR framework to operationalise the concepts from both disciplines to perform the interdisciplinary synthesis. This paper identifies two key areas for potential fruitful interdisciplinary research for computable contracting practitioners and contract theorists. Firstly, contract theorists can investigate how the new information revelation mechanisms affect the incentive balances in various contracts and utilise these new cheap state verification mechanisms to design optimal contracts. This approach is directly applied to the hold-up problem. Secondly, the contract theory ontology can be used conceptually to inform the design of contract analytics functions; likewise the structured data sets created by computable contracting methods can advance empirical contract theory.

Preface and Acknowledgements

The story of my dissertation begins with, like many good stories, undergraduate arrogance. I first became interested between the intersections of computer science and law after studying ‘Cyberlaw and Governance of Digital Markets’ (ESPS2304) under Dr. Alessandro Spano in my second year. Over the 2020 summer, I came across ‘smart contracts’ when researching trends in law and technology. I soon found the smart contracts and computable contracts research groups at UCL. Recalling contract theory’ being mentioned in a lecture in ‘Interdisciplinary Game Theory’, I wondered whether there had been any research combining the two fields. I found almost nothing. My dissertation topic was decided, I would combine them!

With hindsight, this was a gargantuan research topic, almost impossible to make sense of. Approaching the topic head-on with little strategy and little knowledge of these two sprawling and high-technical fields led me down many research paths. My misguided efforts included trying to use the formal proof methods learnt in my ‘Logic’ course to prove the principal-agent model was computable and trying to mathematically formalise derivatives contracts using contract theory. Eventually, I clarified my research aim and chose to explore more general avenues for interdisciplinary collaboration between the two fields and introduce academics from both fields to each other’s research. I hope this research can be of some use to this end.

‘Interdisciplinary Game Theory’ (BASC0017), taught by my secondary supervisor Dr. Manuela Dal Borgo, was instrumental to researching, writing and conceptualising this dissertation. The course familiarised me with the formal methods and notation within game theory, which was essential for me to make sense of the contract theory literature when researching. It also provided me with the methodological background to apply game theoretic methods in interdisciplinary contexts.

I also have to thank my brilliant, and kind-hearted, mathematics lecturer, Dr. Isidoros Strouthos. Alongside pointing me towards Leibniz’s work on law, his ‘Logic’ (MATH0050) course this year allowed me to engage with some of the more technical literature on programming languages and logic approaches to contracting.

Mathias Dewatripont and Patrick Bolton's textbook 'Contract Theory' was essential in my research. The foundations built by their thorough explanation of a variety allowed me to engage with the academic literature on contract theory. Additionally, I'd like to thank Daniel Barron for making his notes for a PhD course he taught on contract theory at Harvard public. They were instrumental to this dissertation and gave simple explanations of a wide-range of contract theory models from influential papers.

Crucially, I'd like to thank Prof. Chris Clack, not only for being my supervisor and advising me, but for producing the most comprehensive whilst accessible pieces of literature on computable contracting and smart contracting. His papers are a paragon of interdisciplinary communication, whose style I've tried to emulate.

Of course, I thank all the other authors in my bibliography. My research stands on all their shoulders.

Finally, I'd like to thank my mum and sister for more than everything.

I like to think
(it has to be!)
of a cybernetic ecology
where we are free of our labors
and joined back to nature,
returned to our mammal
brothers and sisters,
and all watched over
by machines of loving grace.

- Richard Brautigan, 1968, p.194

Contents

1 Introduction

Commercial contracts allow parties to exploit the surplus gained from cooperation by securely aligning interests between counterparties. By setting enforceable agreed rules align incentives enabling co-operation between self-interested counterparties who have imperfect information about each other can engage in fruitful commercial relationships (Scott and Triantis, 2005). However, contracting has lagged behind in terms of technological innovation compared to other areas of business. The acceleration of information technologies has yet to be felt in the "drafting, reviewing, execution of contracts" (Clack and Cummins, 2020 p.1). A Linklaters report argues that "the production, execution and performance of legal agreements", in the long term, will "be transformed by the trends of digitisation and automation that are reshaping so many other spheres of human activity" (Harley, 2017 p.2).

Computable contracting is the next step in "mechanizing the drudgery of the practice of law" (Kelso, 1945 p.392). Computable contracting seeks to transform inert natural language contracts and increase their business connectivity through computational methods. The ultimate aim of computable contracting is machine and human understandable contracts (Clack and Cummins, 2020). Clack and Cummins (2020) emphasises the iterative nature of the computable contracting vision; the field aims for an orthogenetic evolution towards increasing the computability of contracts.

Realising computable contracting will require a "range of technologies and approaches" - it is a fundamentally interdisciplinary endeavour (Clack and Cummins, 2020 p.1). Today, researchers draw on knowledge from linguists, lawyers, computer scientists and business experts (Clack, 2021). Researchers invite perspectives from industry experts, regulators and lawyers, to ensure their research is actionable. By involving agents from influential private firms, the adoption of computable contracting is accelerated. Computable contracting is an instrumentalist endeavour, directed towards increasing the computability of contracts and realising contracts that are simultaneously machine and human understandable; it is therefore inherently pragmatic in its epistemology and methodology. Methods and knowledge are valid if they advance research towards the vision of the widespread adoption of simultaneously machine and human understandable contracts. This paper aims to widen the interdisciplinary dialogue to include contract theorists.

Contract theory is a subdiscipline of economics which mathematically formalises contractual relations using game theoretic methodology to explain why parties enter into contracts, the incentives underlying contractual mechanisms and how to design optimal contracts. The underlying epistemological similarity between the two disciplines is representing contracts in highly formal languages. Contract theory can be of use to the development of computable contracting by identifying perennial incentive and informational issues which these new computationally-enabled contractual mechanisms can remedy. These changing contractual mechanisms and information revelation mechanisms in contracting will be of particular interest to contract theory researchers. The research within computable contracting focusses on designing computer languages and computer programs to automate contracting, there is little research on how the new mechanisms created by the increasing computability of contracts can affect the incentive balances within commercial contracts.

A few papers have briefly pointed towards the possibility of fruitful interaction between contract theory and computable contracting (Dütting, Roughgarden and Talgam-Cohen, 2019; Clack and Cummins, 2020). A few papers have analysed some computable contracting methods (in particular smart contract) from an information economics perspective (Holden and Malani 2017; Tinn, 2018; Gans, 2019; Cong He 2019; Chen, Cong Xiao, 2021). This is the first paper synthesising these two relatively esoteric, but highly complementary, areas of interdisciplinary research. In addition to aiming to fill this gap in research, this paper suggests areas for future research, making the case for a cross-pollination of ideas between contract theory and computable contracting.

The "central notions of contract theory" have "wide applicability", but contract theory articles are "hard to penetrate even for a well-trained reader", posing barriers to their interdisciplinary application (Bolton and Dewatripont 2005, p.3). Computable contracting involves highly technical concepts from theoretical computer science, law and linguistics, however practitioners aim to make literature as accessible as possible so papers are accessible to both lawyers and computer scientists, whose collaboration is essential for research.

This paper follows this tradition in order to maximise access and impact.

Global challenges "require the joint involvement of researchers from different disciplinary backgrounds" (Tobi and Kampen, 2018 p.1209). This paper utilises 'methodology for interdisciplinary research framework' (MIR) to approach this broad interdisciplinary research aim (Tobi and Kampen, 2018). After identifying the affinities in research aims of computable contracting and contract theory, the concepts within contract theory had to be 'operationalised' in order to analyse how this theoretical perspective can inform computable contracting. In order to do this, the research aims of computable contracting were analysed. Computable contracting is a fundamentally instrumentalist endeavour, aiming in its research to advance the computability of contracts (Clack and Cummins, 2020). Accordingly, concepts from contract theory were operationalised to further the instrumentalist aims of computable contracting.

Epistemologically, this work also draws on Leibniz's legal work. 'Contract theory' and 'computable contracting' can be contextualised into an over 300 year history of academic work to formalise legal proceedings. The breadth of the aims and scope of Leibniz's work reflects the enlightenment sense of human perfectibility, disciplinary holism and theological naturalism. Today's work in formalising the law (both contract theory and computable contracting) reflects the atomism, division of labour and instrumentalism within modern academic research. Leibniz is known for his contributions to mathematics, computation and the use of formal methods in philosophy, however he also produced many transdisciplinary legal works.

Artosi and Sartor (2016) argues Leibniz's work as a jurist is characterised by three principles. Firstly, legal problem-solving and research require an interdisciplinary dialogue between law and many other fields (e.g. logic, statistics, theology, politics etc.). Secondly, legal work requires intradisciplinary dialogue, by considering multiple jurisprudential theories and traditions when considering legal problems. Thirdly, legal work requires a pragmatic selection of "highly innovative tools, including, but not limited to, those afforded by contemporary developments in logic and mathematics" (Artosi and Sartor 2016, p.6). These principles inform this paper's approach to legal research. Of course, in the first aspect, being an interdisciplinary dialogue between law, mathematics, computer science and economics. In the second aspect, jurisprudential perspectives (critical legal theory) are considered in relation to computable contracting, smart contracts and contract theory in Section 5. The third aspect is the central to this dissertation, computational law and contract theory. This paper, computational law and contract theory, all utilise "a large toolbox of reasoning methods and cognitive tools" to solve legal problems (Artosi and Sartor 2016, p.6). Leibniz's approach to law provides an interdisciplinary tradition for this paper to draw upon and can further be used to contextualise computable contracting research - and contract theory research to a lesser degree - in the academic tradition.

This paper seeks to be a comprehensive introduction to contract theory and computable contracting to each other's fields of research. The aim of this dissertation is to stimulate interdisciplinary work between contract theorists and practitioners of computable contracting. Accordingly, this paper includes many citations, to allow interested readers to explore literature which they may find stimulating and is written accessibly, defining technical terms. Section 2 introduces computable contracting, contextualises its research into the wider history of computational law and legal informatics, introduces current research challenges and directions and details how computable contracting methods are being used in industry. Section 3 introduces contract theory, gives a basic model of the principal-agent model and explores the contract design methodology and some recent developments. Section 4 synthesises the findings of computable contracting and contract theory, first exploring generally how new information revelation mechanisms can affect incentives in contracts and then more specifically looking at how these mechanisms can be used in the 'hold-up' problem, it finally explores how the contract theory ontology could be used for contract analytics functions and considers how contract standardisation could stimulate work in empirical contract theory. Section 5 discusses how computable contracting fits into the broader trends of interdiscplinary research and reflects on the differences in epistemologies and methodologies in contract theory and computable contracting.

2 Computable Contracting

Commercial contracting is lengthy, cumbersome and manual, incurring significant transaction costs in commerce, however, they are necessary to manage legal risk. A 2018 Harvard Business Review article estimated that "inefficient contracting causes firms to lose between 5

Computable contracting aims to create legal contracts that are both understandable by humans and computers (Surden, 2012; Agarwal, Xu and Moghtader, 2016; Clack and Cummins, 2020). Making contracts machine understandable will structurally enable digital automation of a far larger set of operations. At this "maximal level [...] all clauses of the contract are written as a computer program and are available for query, analysis, verification and automated execution" (The CodeX Insurance Initiative Working Group, 2021 p.1). Surden (2012), alongside coining the term 'computable contracting', highlights potential for computable contracts to reduce the high transaction costs which, today, are corollary with commercial contracting. The full implementation of this vision will provide large cost reductions, efficiency improvements and risk reduction. Additionally, the increase in devices providing trusted streams of data, driven by the adoption of DLTs and the 'Internet of Things' (IoT) devices, increases the set of contractible data streams (Suliman et al., 2019). Stanford University's 'CodeX' research stands out as a hub for innovative research on the intersections of computer science and law, housing many cutting edge legal informatics projects (e.g. the 'Stanford Computable Contracts Initiative', the 'CodeX Insurance Initiative', the 'Computational Law' project etc.).

Commercial contracts are complex arrangements which, today, cannot be reduced into computational form. Computable contracting is an iterative vision, aiming for an orthogenetic evolution towards increasing computability of contracts. However, these complex arrangements can be decomposed into automatable aspects (e.g. transactions, title transfers, recording information). Computable contracting practitioners, although seeking increased computation within contracts, do not automate the entirety of commercial contracts, as many provisions within contracts are left vague due to costs in drafting for all contingencies, which can be interpreted later in the case of a dispute. However, this will allow 'vagueness' to be “employed in a more deliberate way” (Clack and Cummins, 2020 p.5). “The evolution towards computable contracting will take years, and [... it] will build on many of the contracting technologies already being used today”, accordingly this paper uses the term ‘computable contracting methods’ to refer to technologies which will increase the computability of contracts, advancing progress towards the vision of machine and human understandable computable contracts (e.g. smart contracts, contract analytics, visualisation tools etc.) (Clack and Cummins 2020, p.14). Today computable contracting methods are being applied iteratively, with practitioners expecting the adoption of ‘hybrid computable contracts’, where parts of contracts are expressed as machine and human understandable code (Surden et al., 2021).

This section will explore the history of computable contracting and contextualise it into the longer history of computational law and introduce recent developments in computable contracting research (Section 2.1), then it will catalogue how computable contracting methods are being implemented in industry today (Section 2.2).

2.1 Contextualising Computable Contracting

Computable contracting fits into a long academic history of attempts to automate various aspects of legal practice. As modern computers emerged, the new tools available prompted a renaissance of research formalising legal processes (Kelso, 1945). One of the most pressing areas was how to use computers for information retrieval. This gap in capacity spawned the discipline of ‘legal informatics’, which has now, with emergence of computational reasoning and explosion of related academic research, expanded to include many endeavours including computable contracts, machine understandable and executable pieces of legislation.

The more ambitious side of this academic history involves the creation of universal languages to express all material and logical concepts. Leibniz envisioned a characteristica universalis - a universal formal symbolic language which could express all worldly and logical concepts (Rutherford, 1996). The language would be an ‘alphabet of human thought’ from which conclusions can be deduced via a calculus ratiocinator which applies logical rules to the propositions expressed in the language (Rutherford, 1996). Leibniz envisioned these two tools to be used in a variety of settings - mathematics, the natural sciences and metaphysics (Rutherford, 1996). All legal decisions could be resolved deductively and definitely, avoiding the 'chicanery' of lawyers (Artosi, Pieri and Sartor, 2013). Leibniz wrote his graduate dissertation - 'Disputatio Inauguralis de Casibus Perplexis in Jure' (Inaugural Disputation on Ambiguous Legal Cases) - on reducing 'ambiguous legal cases' to fundamental logical constants and propositions using combinatorics (Artosi, Pieri and Sartor, 2013). He continued similar work in 'De legum interpretatione, rationibus, applicatione, systemate' (the reasoning, application and system of legal interpretation) where he systematised and formally constructed a logic of legal reasoning (Armgardt, 2015). Leibniz never attempted to construct his characteristica universalis, however, Stephen Wolfram, has taken up similar aspirations. His 'Wolfram Language' is a 'computational language' which leverages natural language to provide a means of computationally expressing, organising and logically operating on natural language objects, which he envisions to be used within computational law(Wolfram, 2016; Wolfram, 2019).

More modestly, Allen (1957) develops a framework to append the logical connectives from symbolic logic to clauses and provisions when drafting legal documents to increase clarity of legal documentation. This work can be seen as an early step towards machine and human understandability of legal documents, since the contracts are expressed in a hybrid language. Allen's model logically structures and connects clauses written in natural language with symbolic logic. McCarty (1977) develops a legal reasoning computer programme - 'TAXMAN' - which deduced whether a restructured corporation was exempt from income tax after entering key pieces of information (e.g. year of incorporation, location of incorporation etc.). There have been many 'Domain Specific Languages' (DSLs) for legal purposes over the years, an early example is Stamper's LEGOL language which expressed "complex rules and regulations" in computer code (1977 p.102). Similarly, Sergot et al. (1986) expresses the 'British Nationality Act' in Prolog, a logic programming language. In the late 1990s, Ian Grigg proposed 'Ricardian Contracts' which use markup languages to annotate some provisions in natural language contracts pieces of code which are used to execute these obligations by computer (Grigg, 2004; Clack, 2021). Although the code and contract are separate, they are contained within the same document, creating legal documents which are "both readable by people and parsable by programs" (Grigg, 2004 p.4).

Today, computable contracting research focuses on the creation of a machine and human understandable contract drafting language (Clack, 2021). The predominant approach to engineering such a language is to design a legal 'domain specific language' (a computer language designed to be used in a specific setting) which is also a 'controlled natural language' (a highly constrained and structured variant of a natural language which restricts grammar and vocabulary to eliminate ambiguity and ensure machine readability). Clack (2021) provides an excellent introduction to current research directions of languages for smart and computable contracts. Translating the syntax of natural language contracts into computer code, whilst retaining the precise semantic meaning of the legal agreement, is the key challenge in computable contracting research - this is known as 'validating' the code (Clack, 2021). Particularly as computable contracting aims to automate high-value contracts (e.g derivatives contracts) (Clack and McGonagle, 2019; Clack, 2021). A current challenge in the validation of computable contracts and smart contract code is translating legal-specific concepts of time into these languages (Clack and Vanca, 2018). Natural language legal contracts are written by lawyers to be understood by lawyers, assuming some shared understanding of the law. Programmers cannot accurately semantically analyse natural language contracts and to translate them into some contracting DSL. Additionally, when these 'higher level' languages (closer to natural language) are translated down the 'language stack' all the way into binary, computer scientists and lawyers must work together to ensure that these 'lower level' languages syntactically and semantically reflect the relations specified by the contract. Interdisciplinary dialogue is essential for validating the computer code (Clack 2021).

2.2 Computable Contracting Methods in Practice

Currently within legal practice, computable contracting is implemented in a limited form. Today, computable contracting methods bridge the divide between commercial contracts and the business systems they govern, but are iteratively moving contracting towards increased computability. Clack and Cummins (2020) identifies 4 main horizons in which computable contracting methods are being realised: 'contract lifecycle management' , 'contract standardisation', 'contract analytics' and 'smart contracts'.

'Contract lifecycle management' (CLM) systems streamline and automate many of the operational tasks surrounding commercial contracts, including: management, compliance, negotiation and They allow the data from contracts and business systems to be efficiently extracted and fed into each other, making operational business tasks and contract drafting more efficient. CLM systems automate around the contract, but do leave the actual natural language text of contracts untouched, instead they provide efficient data streams between contracts and business systems (Clack and Cummins, 2020).

'Contract standardisation', today, is done through the creation of document templates and templating systems. Common contracts and clauses are added to libraries of templates into which specific information regarding the particular relation can be entered. For some simple and commonplace contracts (e.g. an insurance contract, sale of a property etc.) the only information which needs to be entered is the names of the contracting parties (Clack and Cummins, 2020). 'Contract assembly' systems (e.g. HotDocs, Contract Express, Docassemble) utilise 'contract standardisation' to create modular contracts, where clauses and sets of clauses are retrieved from libraries and pieced together to form contracts. Contract standardisation is lowering legal costs by reducing drafting and negotiation time (Kerrigan, 2019). Contract standardisation can increase the efficiency of 'contract analytics' approaches by creating structured sets of contract data (Bommarito, 2021). Structured contractual data can also allow visualisations of contractual information to be automatically created. Many industries trade associations have already created contracting standards (e.g the ISDA in finance and JCT in construction) (ISDA 2019; JCT 2020). The British Cabinet Office has created a standardised 'Model Services Contract' for public sector procurement (2020).

More ambitiously, 'contract analytics' software utilises artificial intelligence (AI) and natural language processing (NLP) technologies to analyse and retrieve information from natural language contracts (Nay, 2021). The recent increases in computing power and efficiency of NLP has made contract analytics commercially viable

(Susskind 2016). These pieces of software can be used to edit contract templates more efficiently (e.g. changing pronouns, conjugating verbs correctly etc.), rather than having junior lawyers carry out this work manually. Junior lawyer proofread contracts, extract particular pieces of information and present them in summary charts; this "achingly tedious" work requires "extreme attention to detail" and often leads to human error (Waisberg,, 2021 p.201). Contract analytics software can also use the extracted information to generate automated reports and graphics. Predominantly, these pieces of software ensure terms are consistent throughout contracts in order to reduce risks and increase clarity. Kira Systems, used by many large international law firms (e.g DLA Piper, Freshfields Bruckhaus Deringer, Clifford Chance etc.) parses natural language contracts to find user-specified provisions and creates summary charts which can be refined by the user (Waisberg, 2021). Structured data greatly increases the efficiency of nascent 'NLP' methods; 'contract standardisation' methods allow for greater effectiveness of 'contract analytics' software. The potential for legal analytics insights has been investigated since the 1960s (see Schubert, 1962; Aubert, 1963; Lawlor, 1963). In the future, contract analytics software aims to look for conflicting provisions, missing provisions and to link contingent clauses. Contract analytics software is reducing transaction costs in large commercial deals by increasing the efficiency of processing contractual data.

Smart contracts are considered to be the next stage in the orthogenetic evolution towards computable contracts (Clack and Cummins, 2020). Over the years there have been many definitions of smart contracts; there is no single stable definition of a 'smart contract' (ISDA, 2017; Lauslahti et al., 2018; Clément, 2019; Clack 2021). The lack of a universal definition has been problematic for academic study, with academic papers (from a variety of disciplines) using the term 'smart contract' to describe subtly different objects. With the field in an embryonic stage, definitions will continue to change as the field develops (Clack 2021). For clarity's sake, below some of the key terms and concepts captured by the term 'smart contract' will be explained.

Nick Szabo (1994) first coined the term 'smart contract' to describe "computerized transaction protocol that executes the terms of a contract", intending for 'smart contract' to be a very broad term. This paper will work with a portmanteau definition in order to capture the widest possible set of 'smart contracts', whilst staying true to Szabo's original intentions for the term. So, a 'Smart Contract', is "an automatable and enforceable agreement. Automatable by computer, although some parts may require human input and control. Enforceable either by legal enforcement of rights and obligations or via tamper-proof execution of computer code" (Clack, Bakshi and Braine 2016 p.2). "Human input and control" is essential as many commercial contracts often govern long-term relationships laws change so must be. However, the definition also allows for tamper-proof smart contracts, which cannot be altered nor stopped, to be considered smart contracts. The term 'smart contract' can be broken down further into two distinct concepts: smart contract code and smart legal contracts (Stark, 2016; Clack, Bakshi and Braine, 2016; ISDA and Linklaters, 2017). The term 'smart contract code' describes pieces of code, recorded and executed on a blockchain, which makes transfers (e.g. of control rights, money, status) when predefined conditions are met (Clack, 2018). Whereas, 'smart legal contract' describes legal contracts, which have, at least, some part being automated by smart contract code (Clack, 2018).

Today, aspects of legal contracts are automated externally with different pieces of code each written by each counterparty being run at each end. Smart contract code within smart legal contracts allows counterparties to use the same piece code to automate certain provisions whilst being run on a single single system. This avoids disparities in the translation of the contract into computer code (e.g. different interpretations of the contract, semantic errors in the computer code etc.) and prevents temporal issues in execution, reducing some of the risks which comes with automating provisions with code.

The term 'smart contract' has become associated with blockchain technologies, after being used to describe procedures stored in the Ethereum blockchain (Buterin 2014). However, smart contracts are not necessarily run on blockchain data structures. Within the Ethereum blockchain once certain pre-defined parameters are met in the blockchain, tamper proof DLT-based smart contracts automatically use this trusted information to execute transactions - the code cannot be edited once it has been agreed (Buterin, 2014). These tamper proof smart contracts are self-executing offering "security and inviolability" to counterparties (Clément, 2019 p.275). The self-executing model confines smart contracts to "relatively narrow sets of operations" (Harley 2017 p.2). However, this tamper-proof mechanism, which prevents renegotiation can be utilised to solve arising from the renegotiation of contracts (see Section 4.2).

Facebook has created a smart contracting language for its DLT-based currency 'Diem' called 'Move' and the Bank of England has considered provisions for adding a 'smart contracting layer' to its proposed 'central bank digital currency' in its latest publicised discussion paper (Blackshear et al., 2020, Bank of England, 2020).

3 Contract Theory

Contract theory mathematically formulates what contractual clauses economically rational (self interested and utility maximising) parties will agree to. At their most simple, contracts in contract theory are a set of state contingent transfers or actions between counterparties. A canonical example is an insurance contract: if the state of the world is X then the insurer transfers the customer Y. These formulations at their most basic contain no private information, hidden actions nor uncertainty, these modulations, and others, are factored in to model different contracting scenarios. Contract theory has been applied to many areas including optimal design of contracts for regulation (Baron and Myerson 1982; Laffont and Tirole 1993), taxation (Mirrlees 1971; Goldman, Leland and Sibley 1984), venture capital financing (Berglof, 1994) and employment (Holmström, 1979; Grossman and Hart, 1992, Holmström 2017) .

Information asymmetry permeates economic activity. Sellers know more about their products and services than buyers, managers know more about the financial outlook of companies they manage than investors, patients know more about their health than insurers and so on. Contract theory explores three main types of information asymmetry which result in incentive problems: adverse selections, moral hazard and contractual incompleteness (Bolton and Dewatripont 2005).

Contractual incompleteness describes when contracts do not specify what actions are to be taken in all relevant states of the word. Ideally, parties would always be able to write contracts which incentivise Pareto efficient outcomes in every possible state of the world.

Until Grossman and Hart (1986)'s seminal paper, all contract theory literature assumed contracts to be 'complete'. That is to say, these early papers contract models contained instructions for every situation, with no room for unanticipated scenarios. However, contract theory studies how contracts come to be in our less than ideal world where there exist many constraining factors. Incomplete contracts, conceptually, reflect the reality that predicting future events is costly. Counterparties often leave contracts incomplete intentionally as the ex-ante costs of drafting and negotiating a clause governing a very unlikely scenario outweigh the benefits (Anderlini and Felli 1999). Additionally, implementation and monitoring costs may outweigh gains. Similarly, practitioners of computable contracting have argued that vagueness in commercial contracts often "reflects the limited resources available for drafting" (Clack and Cummins 2020 p.1).

Adverse selection problems arise from asymmetric information between contracting parties on some relevant property of their counterparties (e.g. skills, product quality, etc.). Parties have information on their own types but not about other parties types, this information asymmetry can be used to hide relevant information from counterparties - for example a worker could signal that they have some skill to comply with an employment contract without actually having the skill. Solutions to adverse selection often involve screening mechanisms, where a set of contracts are offered to agents, with each contract being preferable for a certain type of hidden information, so by choosing the optimal contract for them the agent reveals their hidden information (Maskin and Riley, 1984). Screening mechanisms are used to incentivise agents to self-select contracts which fit their hidden characteristics. These adverse selection problems will be of interest to computable contracting practitioners when considering how to utilise computable contracting methods to better screen for information in counterparties (e.g. creditworthiness).

Moral hazard arises from hidden actions between counterparties. When a party's actions cannot be perfectly monitored, they can be incentivised to take selfish actions. For example depositors do not know how banks are managing their money, owners do not know how managers are controlling their firms etc. Moral hazard arises after the contract has been agreed - when parties take hidden actions - whilst adverse selection occurs before the contract is signed when information is hidden to coerce counterparties. Moral hazard problems may be of interest to computable contracting practitioners when designing mechanisms for monitoring agents.

3.1 The Principal-Agent Model

The agency problem is the basic problem within contract theory. How can self-interested parties assure that their mutually-interdependent counterparty doesn't use their agency to harm them? This basic problem is captured by the fundamental model of contract theory: the hidden-action principal-agent problem (Bolton and Dewatripont, 2005). Holmström (2017) provides a comprehensive overview of extensions and modulations of this foundational model. This section introduces a basic model of the principal-agent problem to demonstrate the methodology and ontology of contract theory.

The model begins with two players, the principal and the agent. The principal offers a contract to the agent, then the agent chooses to accept or refuse, if accepted the agent undertakes some costly hidden action (which benefits the principal to some degree) and then the principal pays the agent according to the magnitude of the contractible signal which is correlated with effort (Holmström, 1979; Grossman and Hart, 1992). The key detail in the model is that effort is not contractible since the principal cannot observe effort nor verify it to a third party. This is a situation of information asymmetry, the agent observes the amount of effort put in whereas the principal can only see the output produced by the action. There is a moral hazard since the agent can take a less costly action since the action is hidden from the principle. So, the optimal contract incentivises the agent to expend maximum effort by making the agent's remuneration a function of the contractible signal of effort (Holmström, 1979).

Consider two counterparties: an owner (principal) and a manager (agent). The owner is looking to hire a manager to manage a firm she owns. She proposes a contract describing a wage scheme to the manager. The manager then decides whether to accept the contract or not. If the manager accepts the contract, he then goes to work and expends some effort. The owner doesn't observe this effort but she does observe the output which comes from the effort. The effort the manager expends correlates to the output but he does not have full control over the output. He doesn't directly gain from the output, but the owner does. How can the owner design the optimal contract such that output is maximised?

To demonstrate the contract theory methodology, the scenario is written formally below:
x is the output produced by the agent.

$$x \in \mathbf{R}$$

There are n possible outputs.

$$x_1, ..., x_n$$

The probability of outcome x_i depends on the chosen effort, e, and is defined by:

$$P_i(e)$$

For all effort levels e:

$$P_i(e) > 0$$

So the probability of observing output x_i for any given level of effort is > 0.

x_i can be the result of any input e due to the probability function, $P_i(e)$

The principals utility function is defined as:

$$B(x - w)$$

Where x is the output produced and w is the wage paid to the agent.

$$B'(x - w) > 0$$

and

$$B''(x-w) \leq 0$$

So the principals utility function is increasing and either concave or linear.

The agents utility function is defined as:

$$U(w) - v(e)$$

Where $U(w)$ is the utility the agent derives from the wage payment and $v(e)$ is the disutility the agent derives from exerting effort.

$$U'(w) > 0$$

and

$$U''(w) \leq 0$$

So the utility the agent derives from the wage is also increasing and either concave or linear

$$v'(e) > 0$$

So the disutility the agent derives from exerting effort is increasing, but can be concave, convex or linear.

So to ensure that the agent exerts maximal effort, the following is optmised when designing the contract.

$$\pi(e^k) = max \sum_{i=1}^{n} P_i(e^k) B(x_i - w_i^k)$$

Where π is the expected utility and e^k is the maximal effort the agent exerts.

3.2 Contract Design

Contract theory is being used to design a wide variety of optimal contracts - for example to design 'Payment for Ecosystem Services' programs that aim to reduce deforestation (Li, Ashlagi and Lo, 2020), to incentivise electric vehicles to provide ancillary power to the power grid during demand surges (Gao et al., 2012; Zhang et al., 2018), to design federated learning systems which use distributed computational power to train machine learning algorithms (Kang et al., 2019a; Kang et al., 2019b; Lim et al., 2021). Early contract theory literature, from the 1970s and 1980s, is mainly concerned with designing optimal compensation mechanisms, corporate structures and financing mechanisms (Bolton and Dewatripont, 2005).

Contract design problems are usually formulated as an optimisation problem to maximise a party's payoff (usually the principal). Overall welfare optimisation can also be how these problems are formulated (Bolton and Dewatripont, 2005). In multi-agent problems, these optimisation problems quickly become unmanageable and require linear programming methods to solve them (Babaioff et al., 2012; Babaioff and Winter, 2014; Dütting, Roughgarden and Talgam-Cohen, 2019). With many contracts taking place in increasingly complex and structured market environments (e.g. ridesharing, crowdsourcing), optimal mechanism and contract design is becoming increasingly relevant (Dütting, Roughgarden and Talgam-Cohen, 2019). Contract design and mechanisms design approaches will be particularly useful when designing the architecture of these new types of computer-mediated marketplaces (Roughgarden and Talgam-Cohen, 2019).

Optimal contract design has been criticised for creating contractual mechanisms which are overly complex (Milgrom and Holmström 1987; Dütting, Roughgarden and Talgam-Cohen, 2019). Mainstream contract theory suggests optimal contracts should be based upon all relevant information to a transaction (e.g. type distribution, risk appetite) (Allen and Winton, 1995). Given the complexity of this set of 'relevant information', optimal contracts tend to be extremely complex (Hart and Holmström 1987). However, in some relatively complex models, the optimal contract is linear rather than complex (Milgrom and Holmström, 1987). The complexity described by optimal contracting literature far exceeds the complexity of contractual mechanisms empirically observed (Stiglitz, 2017). In practice, contracts tend to be simple and have a linear form - rewards are usually calculated using a linear function.

Optimal contracts often include non-monotonic payoff functions

(Dütting, Roughgarden and Talgam-Cohen, 2019). That is to say, the payoff for an agent may decrease as their effort increases according to some optimal contracts, creating a non-intuitive contract which incentives agents who expend high amounts of effort to hide output. Computable contracting practitioners have emphasised the importance of keeping contracts easily accessible to humans (Surden, 2012; Clack and Cummins, 2020). To operationalise the optimal contract design methodology within computable contracting, contracts must be as simple as possible whilst retaining optimality to keep them accessible to humans.

Over the past two decades, there has been a lively interaction between microeconomists and computer scientists in the fields of mechanism design and signalling which has resulted in the field of 'algorithmic game theory' (AGT). The field has had many high-profile use cases e.g. designing sponsored search algorithmics for Google (Lahie et al., 2007) and designing incentives in peer-to-peer networks (e.g. BitTorrent) (Qiu and Srikant, 2004). The AGT community is aiming to recreate this success by applying their knowledge to contract theory to create 'algorithmic contract theory' (ACT) (Talgam-Cohen and Dütting, 2019). Contract theory's game theoretic roots, increasing relevance and strong ties to optimisation provides a "huge opportunity" for the AGT community for further fruitful research (Dütting, Roughgarden and Talgam-Cohen, 2019 p.385). The algorithmic lens provides a method, through linear programming, to create approximately optimal solutions when optimal solutions are inappropriate (usually due to complexity) (Dütting, Roughgarden and Talgam-Cohen, 2019; Dütting, Roughgarden and Talgam-Cohen, 2021). Recently, this methodology has been used to design approximately optimal simple contracts for Payment for Ecosystem Services (PES) programs, which seek to reduce deforestation (Li, Ashlagi and Lo, 2020). This approach is limited in that this optimisation method requires perfect knowledge of any distributions used (e.g. probability density functions of type, output, risk etc.). However, linear programming optimisation allows computer scientists to "to systematically explore complex economic design spaces, and to identify "sweet spots" of the design space where there are plausibly realistic solutions that simultaneously enjoy rigorous performance guarantees" (Dütting, Roughgarden and Talgam-Cohen, 2019 p.385). Additionally, the algorithmic approach to contract theory has been used to show simple contracts are more robust to unmodeled eventualities than optimal contracts (Dütting, Roughgarden and Talgam-Cohen, 2019). In light of this, it is being used to compute 'approximately optimal' linear solutions to contracting problems (Dütting, Roughgarden and Talgam-Cohen, 2019; Gao et al., 2020; Dütting, Roughgarden and Talgam-Cohen, 2021; Gao et al., 2021). As this field gathers momentum, these 'approximately optimal' linear solutions may be of interest to computable contracting researchers and end users when designing contracts.

4 Synthesis

Computable contracting methods create the "possibility to design new, more flexible, types of contracts" through new contractual monitoring, enforcement and control mechanisms (Tinn, 2018 p.40). The new set of mechanisms computable contracts create will influence the incentive balance of commercial contracts (Section 4.1), with some of these new mechanisms being able to remedy some perennial incentive issues within commercial contracts (Section 4.2). Furthermore, contract theorists can identify some of these problems which computable contracting methods can remedy. The contract theory ontology can also be a useful when approach newly created sets of structured legal data (Section 4.3)

4.1 New Information Mechanisms and New Incentives

The increases in the efficiency, speed and reliability of information transfer between counterparties provided by computable contracting methods can make compliance and oversight of contracts extremely cheap alongside increasing the set of contractible variables. With increased trust being a key argument for the adoption of DLTs, how can the change in information distribution be utilised in contracting settings? Although computable contracting and smart contracting are by no means contingent upon hosting using blockchain data structures, these data structures provide space for a larger set of trusted data streams which can be contracted upon. There is a growing set of literature on the intersections of blockchain and information economics (Holden and Malani, 2017; Tinn 2018; Cong and He 2019; Chen, Cong Xiao, 2021). This section surveys this growing literature and analyses how contract theory approaches to computable contracting methods can provide increases to the optimality of contracts.

The creation of trusted data streams through blockchain technologies enlarges the contractual space by creating a larger set of contractible variables which are admissible in a court of law (Cong and He 2019). There are three types of information in mainstream contract theory: private information (only known to one party), observable information (known to the contracting parties but not observable by third parties) and verifiable information (information which can be verified by a third party) (Lind and Nyström, 2011). The distinction between verifiable and observable variables opens up the space for incomplete contracts (Grossman and Hart, 1986; Hart and Moore, 1990; Hart, 1995). Parties are far less likely to contract on variables which are observable but not verifiable (Hart and Holmström 1987). In the case of a dispute, provisions contingent on or governing information which cannot be verified by third parties cannot be used in litigation since there is insufficient admissible evidence. Including clauses on unverifiable variables amounts to wasting money in drafting these ineffectual clauses (Anderlini and Felli 1994). As the incompleteness of contracts increases, the space for litigation increases, in turn increasing legal risk (Bolton and Dewatripont, 2005). The verifiability of information pertaining to contractual provisions is "a key issue in contract enforcement" (Stiglitz 2017 p.13). The distinction between observable and verifiable information and the suboptimality which comes from unverifiable variables in contracting is useful when considering how computable contracting technologies (particularly DLT-based technologies) affect the incentive balance of commercial contracts. Blockchain, as a shared repository of information which is dynamically updated, is a source of verifiable rather than observable information. Furthermore, many jurisdictions have made specific provisions to clarify that evidence stored on blockchains is admissible in a court of law (e.g. Vermont, Arizona, Ohio, Delaware) (Caytas 2017). By enlarging the set states of the world which can be verified, blockchain data structures can "greatly reduce the scope of non contractible contingencies" which underpins the incompleteness of contracts (Cong and He 2019 p.9).

Contract theorists have cited the unreliability of information as a key reason why parties require a large amount of assets to be attractive debtors for external financiers (Tirole, 2006). The costs in ascertaining whether counterparties are trustworthy makes it necessary for contracts and organisations to be structures assuming counterparties are untrustworthy (Williamson, 1985). Debt contracts are attractive since they are robust to renegotiations and are easily enforced - liquidating a defaulting debtors assets is a near certainty (Hermalin and Katz, 1991; Hart and Moore, 1998; Dewatripont, Lergos and Matthews, 2003). When theorising optimal contracts for corporate financial structure, 'costly state verification' has featured in many models (Townsend, 1979; Allen and Winton, 1995). Townsend (1979) is the first paper using this costly state verification principle. These models are augmented versions of the basic principal-agent model. In these models, investors (principals) have to incur a large cost to verify ex-post variables within the firm (agents) (e.g. output, actions by managers etc.). Given these assumptions, many of these costly state verification models conclude that the optimal contract (to maximise both investment and return) is debt (Diamond, 1984; Gale and Hellwig, 1985; Williamson, 1986; Lacker and Weinberg, 1989; Allen and Winton, 1995; Winton 1995; Hart and Moore, 1998). These theories are referred to as 'control theories' since the allocation of certain rights (e.g. liquidation rights, cash flow rights) in the contract precludes the debt solution. However, considering the asymmetry of risk-sharing in debt contracts to fund investment in firms, this solution is limited in its efficiency, but regardless is the optimal solution given the assumptions.

Blockchain technologies can allow parties to credibly pledge a larger set of control rights, widening the set of possible optimal solutions to contracting problems. For example, future cash flows can be credibly pledged to their creditors, which can in turn reduce the asset requirements for external financing (Tinn 2018). However, these DLT enforced cash-flow pledges do not automatically make contracts complete (in the contract theory sense of the term). The generation of the cash flow lies outside of this smart contract code commitment to direct the cash-flow to the creditor, as seen earlier in the principal-agent model, "generating cash flows still requires human effort that is neither contractible nor verifiable" (Tinn 2018 p.3). Ultimately "contingencies traditional contracts cannot specify are also hard to program into smart contracts" (Chen, Cong and

Xiao, 2021 p.30).

Linking various pieces of financial information to financing contracts will allow investors to update their beliefs more frequently and make decisions more quickly (Tinn 2018). However this can bring in 'rational herding' type problems (Cong and Xiao, 2019). 'Rational herding' describes when market participants act according to the behaviour of other participants rather than acting on the information within the market (Chamley 2004). Rapid information cascades can affect the behaviour of counterparties using computable contracts, this requires more research and consideration.

Williams (1989) provides a model similar to a DLT-based solution, by arguing for 'ex-ante monitoring systems' (e.g. accounting controls) which place strictures on the amount of the firm's output managers in the firm can control, however these limits do not verify the state of the accounts of the firm. Even with these ex-ante mechanisms, debt remains the optimal contract. When state verification is costly, debt contracts allow for efficient monitoring with low state verification costs (Tinn 2018). Here is another example where the new mechanisms provided by computable contracting methods can change the underlying incentive balance within contracts.

Costly state verification also bounds the efficiency of insurance contracts (Bond and Crocker, 1997). To reduce the impact of costly state verification, DLT-based smart contracts are being utilised to make 'parametric insurance' products (Sheth and Subramanian, 2019). These products automatically trigger payouts once certain parameters are met (e.g. when a sensor detects soil reaching a particular level of saturation). DLT-based smart contracts allow the information to be input into the system trustlessly and for the payouts to be triggered automatically. AXA, the multinational insurance firm, has experimented with these technologies to develop parametric flight insurance products and continues to explore the area (Scharrer, Pieciak and Siciliano, 2021). Parametric insurance is currently being employed by Caribbean governments to insure against catastrophic climate risks (e.g. hurricanes, cyclones, earthquakes etc.); the speed of payouts under parametric insurance allows these governments to access money quickly in times of crisis (Horton, 2018). Although this product does not use DLT-based smart contracts, they can further reduce state verification costs, which will in turn expand access to insurance products into poorer countries that tend to be more exposed to climate risks (Horton, 2018). These products can be used to protect against more commonplace climate risks, like crop-failure, giving underinsured farmers across the world some protection against climate risk (Surminkski, Bouwer and Linnerooth-Bayer, 2016; Sheth and Subramanian, 2019).

Information design considers how players with hidden information may optimally reveal their information to guide the actions of their opponents (Milgrom, 1981; Crawford and Sobel, 1982; Kamenica and Gentzkow, 2011). Additionally there is work on optimal monitoring mechanisms to incentivise effort expenditure from agents (Boleslavsky and Kim, 2018; Georgiadis and Szentes, 2020). Dynamic information design problems have been investigated, where informed players reveal information over time and can also learn information (Horner and Lambert, 2016; Ely, 2017; Ely and Szydlowski, 2020). Bergman and Bonatti (2019) provide a comprehensive survey of recent information design literature. These optimal monitoring and information design approaches may be of interest to computable contracting practitioners and end-users when designing contractual mechanisms which alter information revelation structures.

4.2 The Hold-Up Problem and Smart Contracts

If markets are so efficient, as economists claim, why do around half of all domestic transactions occur and a third of international transactions occur within firms rather than between firms in markets (Coase, 1937; Hart, 2017; Holden and Malani, 2019)? Why not conduct business on markets through contracts? Transaction costs between firms dissuade them from using markets for trade making intrafirm trade more attractive, in turn providing a rationality for vertical and horizontal integration of firms (Coase 1937; Hart 2017). In general, there are two types of transaction costs ex-post and ex-ante (Williamson and Masten, 1999). Ex-post transactions costs occur once parties enter into a transaction (e.g. negotiating, drafting contracts, bounded rationality issues). If these ex-post haggling costs are high, firms transact intrafirm if they are low firms utilise markets for transactions. Williamson (1975) explores ex-ante transaction costs which occur before parties enter into a transaction (e.g. sunk investment costs). If renegotiation is possible, then there is a risk of these sunk costs being lost. Coase and Williamson's contributions informally explain why some firms would prefer to transact intrafirm.

Grossman and Hart (1986) provides a formal model (referred to as property rights theory) which describes how residual control rights, which are corollary with asset ownership, explains the boundaries of the modern firm. Residual control rights concern how owners decide to utilise or make decisions regarding assets they own. Two firms engaged in an arm-length contract both have residual control rights over the assets they individually own. However, if the firms merge, then the owner has residual control rights over the assets of both of the firms. If a firm is engaging with a transaction with another, a contract must be written specifying how counterparties agree to use their residual control rights to get some outcome. Residual control rights are conceptualised as a good in this model, so "like any other good: there is an optimal allocation of them" (Hart, 2017 p.1733). In the real world, contracts are incomplete, they do not take into account every single state of the world.Until this paper, all the formal literature on contracts assumed contracts were complete (Hart, 2017). When firms transact through contract not all contingencies (e.g. the exact qualities of a product) are drafted into the contract due to drafting costs (Grossman and Hart, 1986; Anderlini and Felli, 1994).

With these assumptions (incomplete contracts and residual control rights) in mind, consider two firms: Sony and Nvidia. Sony requires 100,000 graphics processing units (GPU) for its latest 'PlayStation' and offers a Nvidia contract of £150 per GPU, so the contract's value totals £15,000,000. It will cost Nvidia £7,500,000 to design the new GPU and £50 to produce each unit, giving them a profit of £2,500,000. Nvidia invests the £7,500,000 designs and builds the new GPU. Sony argues the GPU will produce too much heat when running graphically-intensive games. Microsoft offers Nvidia £140 per GPU for its latest 'Xbox', less than Sony's offer since the GPU won't fit perfectly within the case of the 'Xbox' so some design modification on the case will need to be made. This offer totals £14,000,000. Sony offers £14.70, per GPU, totalling £14,700,000. Nvidia has to choose whether to accept the £1,500,000 profit from Microsoft, the £2,200,000 profit from Sony or to bring the case to court. Regardless of whether Sony's claim is reasonable, it is better than the second best offer and litigation costs could outweigh the gains. This allows Sony to 'hold-up' Nvidia and pay a lower price than originally agreed. Sony can extract some of the gains from Nvidia's investment through the incompleteness of the contract.

Hold-up problem is an example of opportunism which is incentivised by incomplete contracting terms. Hold-up dissuades exante investment by firms since firms risk being held-up and having the gains of their non-contractible investment extracted by buyers (Williamson, 1975). Of course a solution would be create a "completely state-contingent" contract which includes provisions for all possible circumstances, however this is not possible in the real world (Holden and Malani, 2019 p.5). In the case of a dispute ex post renegotiation is necessary, opening up the possibility of haggling during the negotiation. Anticipating this, producers underinvest since they take on renegotiation risks of a lost sunk costs, in turn reducing overall gains from trade and output. Modern firms vertically and horizontally integrate other firms in their supply firms to enjoy the utilities of residual control rights. By preventing hold-up, residual control rights within firms encourage relationship-specific investments and prevent haggling costs. In the world of high transaction costs and incomplete contracts, the welfare benefits from the price mechanism can be outweighed by the trust and certainty from the authority relationship within the firm.

Contract theorists have used complex 'revelation mechanisms' and 'renegotiation design mechanisms' to design contracts that are robust to hold-up problems (Moore and Repullo, 1988; Aghion, Dewatripont and Rey, 1994; Maskin, 1999). These mechanisms haven't been observed empirically. They are ultimately ineffectual since they are not 'time consistent' commitments - parties anticipate that they can decide to negotiate outside the mechanism in the case of a dispute and (Holden and Malani, 2014).

Holden and Malani (2019) argue tamper-proof DLT-based smart contracts could remedy hold-up problems, by being able to commit parties to no renegotiation. With no possibility of renegotiation and automatically executing contracts parties cannot be held up and forced to make more credible upfront commitments. However, this may be an unattractive solution due to the possibility of unforeseen events or changes in environment requiring renegotiation. Smart contracts could utilise sensors to detect certain contractible specifications concerning product quality (Gans, 2019). Although inappropriate for many transactions, this solution could be utilised in some marginal cases where sensors can detect contractible variables (e.g. markets for raw materials).

More broadly, computable contracting methods therefore can change the boundaries of the firm by reducing the space for the incompleteness of contracts by making more states of the world verifiable. This in turn moves economic activity towards markets, allowing firms make use of the efficiency of price mechanism and allowing smaller firms to engage in more transactions without needing to be vertically or horizontally integrated. DLT-based smart contracts have been envisioned by some to enable a 'bazaar economy' of "nominally independent contractors in place of centralised firms" (Allen 2021 p.55). Increasing the space of contractible variables and decreasing transaction costs incentives more activity within markets.

4.3 Utilising the Contract Theory Ontology for Data Analysis

Contract theory's formal approach to contracting incentives results in an ontology of contractual incentive objects which provide an alternative means of formally conceptualising contractual agreements, which have the potential to be useful for contract analytics. The legal design process requires 'interdisciplinary teams' to build rich and practicable solutions; contract theorists may be able to enrich this process through their formal understanding of incentive problems within commercial contracts (Hagan, 2021). By explicating succinctly how certain mechanisms within contracts create particular incentive dynamics contract theorists can help to create 'user-centered contracts' which visualise incentive dynamics (Berger-Walliser and Haapio, 2013). For example, insurance models of contract theory can be used to create an ontology for visualising the magnitudes of risk appetite and risk premium within insurance contracts (Spence and Zeckhauser, 1978). Contract design methods provide competitive advantage for law firms who can create human-centred contracts by reducing the transaction costs arising from the time spent parsing legal contracts and making contracts more accessible to clients (Argyres and Mayer 2007). This competitive advantage can be extended through incentive analysis functions which explicate to contractors the effects of provisions on the incentive balance of contracts.

The structured legal data created by various computable contracting methods has the capacity to stimulate research on empirical contract theory. A perennial methodological issue cited in numerous empirical contract theory works is the "lack of adequate data" and the expensive procedure of parsing unstructured natural language contracts to obtain relevant data for empirical work (Chiappori and Salanié, 1997 p.944). Contracting standards initiatives (e.g the ISDA Master Agreement for derivatives contracts, the JCT Standard Building Contract for construction contracts and the Cabinet Office Model Services Contract) are beginning to create structured sets of legal data. However, an essential piece of information for empirical contract theory research is the outcome of contracts, contract standardisation initiatives however do not make the outcomes of contacts public (Chiappori and Salanié, 1997). The value of these structured datasets can be further enhanced by including contract outcomes. Sheth and Subramanian (2018) uses contract theory formalisms to model DLT-based smart contracts on the 'Etherisc' platform. With the blockchain being public, this model and dataset could be utilised for empirical contract theory work. Structured legal data created from the increasing computability of contracts are extremely valuable, from both a research and commercial perspective.

5 Conclusion

5.1 Instrumentalism, the Academy and Computable Contracting

Computable contracting can be viewed as part of this instrumentalist trend in academic research. There have been concerns that within academia "instrumentalism is replacing the liberal humanist orientation, resulting in a privileging of applied knowledge" (Currie and Vidovich, 2009 p.447). Across the academy, ranging from political science to biosciences to computer science, there has been some squeamishness over the increase in activity between industry and academia and the effect this has on research directions (e.g. on-campus presence from in the prevalence of venture capital funded ‘university spin-out companies’, university-managed venture capital funds (Etkowitz, Webster and Healey 1998; Weatherall 2000; Meyer 2003; Croce, Grilli and Murtinu 2014;). Computable contracting academia fits firmly within this instrumentalist trend within academia; many computable contracting scholars create startups and work closely with large influential firms to ensure their research is actionable. Contract theory has taken a more instumentalist turn in recent decades as it has sought to design optimal contracts for a variety of applied scenarios including online markets, crowdsourcing platforms and peer-to-peer energy grids (Dütting, Roughgarden and Talgam-Cohen, 2019). However, relative to computable contracting it is less obviously far less instrumentalist in its applications.

Applied knowledge is favoured in many interdisciplinary endeavours, but the normative principles guiding research fit firmly within liberal humanist norms. Argue similar point for environmental physics, energy systems, smart cities. Interdisciplinary research tends towards instrumentalism due to its ability to subsume politically and socially pressing complex topics (e.g. climate, healthcare, technology etc.) (McLeish and Strang 2014). Instrumentalism is by no means necessarily opposed to the liberal humanist norms which academic research has been associated with in the 20th century (Kitcher, 2003). Computable contracting serves as an example of the synthesis of both the grounding for much work is based in this ‘liberal humanist orientation’, especially those increasing access to law. The internalisation of ‘critical legal studies’ into the American mainstream, has led to a drive to increase the accessibility of contracts to the general public through visualisation through a flourishing new academic interdisciplinary discipline of ‘Legal Design’. ‘Legal Design’ work in the United States, aims to address the inaccessibility of the law to marginalised groups (Perry-Kessaris, 2019). Rather than restricting knowledge of law to “a group of priestly professionals”, computable contracting methods can democratise access to the law (Loevinger 1948 p.455). An accessible computable contracting system, aided by Human-Computer Interaction, could allow “unimaginable benefits for improving access to justice in all societies” (Agarwal, Xu and Moghtader, 2016 p.7). Sections 3.1 and 3.2 demonstrate the possibility for computable contracting methods to increase the contracting space by reducing transaction costs and increasing the set of contractible variables. This can expand access to insurance products to the global south. This increases accessibility of contracting incentivises more economic activity to take place in markets rather than in firms, resulting in a shift of market power away from large vertically integrated firms towards smaller players within the market.

5.2 Reflections on Epistemologies and Methodologies

The instrumentalism of computable contracting goes far deeper than industry-links and 'university spin-out' companies. Both it's epistemology and methodology are firmly pragmatic: methods and perspectives are valued if they advance the materialisation and adoption of human and machine understandable contracts. The field invites a plurality of epistemological perspectives to inform its research, including "computer science, law, logic and linguistics" (Clack, 2021 p.1).

Methodologically, lawyers think backwards when constructing contracts. They envision possible worlds where there could be some dispute and then draft clauses to prevent these disputes or provide a guide for managing these disputes (Clack, 2021). Whereas, computer scientists have to think forwards when constructing a computable contract - the computer code is engineered as a set of instructions to be followed. Contrastingly contract theory takes an atemporal approach to designing contracts. Models are designed to include certain information mechanisms, temporal mechanisms and transfer rules. The standard economic (and mechanics) ceteris paribus methodology is used to explore design spaces and optimisation methods are used to find optimal solutions to the assumptions in the models. These differences in the temporal conceptualisation within their respective methodologies of contract design in the various fields methodology reflects the differences in epistemological aims of these approaches. Lawyers create agreements and instructions which are to be referred to guide a relationship and to if there is some dispute, computable contracting practitioners design sets of instructions for computers which are executed in any case and contract theorists explore theoretical design spaces of contracts to better understand incentives within contracting relationships and the structures of optimal contracts.

This paper aims to make the case on a pragmatic instrumentalist basis that collaboration to include contract theory. The "substantial differences between the language, culture and perspective of lawyers and computer scientists" which are a challenge to computable contracting research are further exacerbated by this paper by adding contract theory, another highly-specific and technical academic field, into the mix (Clack 2021 p.1). However, this paper demonstrates the value of working between disciplines and the challenges in operationalising concepts from, in some aspects, epistemologically discordant disciplines.

5.3 Final Remarks

Contract theory and computable contracting, as fields of academic study, demonstrate the value in crossing disciplinary boundaries in the academy. They take a-traditional highly formal approaches to the law. In contract theory, this yields novel insights about the nature of firms, information and incentives. In computable contracting, the formal methods and interdisciplinary collaboration results in computing languages and systems which increase the efficiency of and access to law. Developing a framework for computable contracting will necessarily unify "many areas of relevant research" (Clack and Cummins, 2020 p.1). Contract theory is assuredly one of these relevant research areas. Contract theory, through its modelling methodology, has been criticised for producing insights and conclusion which are "either too abstract or too specific" (Stremitzer, 2018 p.84) This paper has used the MIR framework to operationalise both the abstract concepts and specific models within contract theory to analyse their applicability to the development of computable contracting, resulting in many areas of fruitful interaction. Contract theory can identify perennial incentive and informational issues within commercial contracts which computable contracting methods can remedy. It can also provide computable contracting practitioners with an alternative theoretical approach when considering how commercial contracts function and structure business relationships. Contract theory and computable contracting are two highly complementary fields with the potential for further fruitful interdisciplinary work.

6 Bibliography

Agrawal, A., Gans, J. and Goldfarb, A. eds., 2019. The economics of artificial intelligence: an agenda. University of Chicago Press.

Agarwal, S., Xu, K. and Moghtader, J., 2016. Toward Machine-Understandable Contracts. AI4J–Artificial Intelligence for Justice, p.1.

Aghion, P., Dewatripont, M. and Rey, P., 1994. Renegotiation design with unverifiable information. Econometrica: Journal of the Econometric Society, pp.257-282.

Alchian, A.A. and Demsetz, H., 1972. Production, information costs, and economic organization. The American economic review, 62(5), pp.777-795.

Allen, F. and Winton, A., 1995. Corporate capital structure, incentives and optimal contracting. Handbook of Finance, Amsterdam, North Holland, pp.693-717.

Allen, L.E., 1957. Symbolic logic: A razor-edged tool for drafting and interpreting legal documents. The Yale Law Journal, 66(6), pp.833-879.

Allen, J. G., Senior Research Fellow at Humboldt-Universität zu Berlin Centre for British. (2021). Bodies Without Organs: Law, Economics, and Decentralised Governance. Stanford Journal of Blockchain Law and Policy.

Anderlini, L. and Felli, L., 1994. Incomplete written contracts: Undescribable states of nature. The Quarterly Journal of Economics, 109(4), pp.1085-1124.

Argyres, N. and Mayer, K.J., 2007. Contract design as a firm capability: An integration of learning and transaction cost perspectives. Academy of management review, 32(4), pp.1060-1077.

Armgardt, M., 2015. Presumptions and Conjectures in Leibniz's Legal Theory. In Past and Present Interactions in Legal Reasoning and Logic, pp. 51-69). Springer, Cham.

Armour, J. and Eidenmuller, H., 2020. Self-Driving Corporations?. Harv. Bus. L. Rev., 10, p.87.

Artosi, A., Pieri, B. and Sartor, G. eds., 2013. Leibniz: logico-philosophical puzzles in the law: philosophical questions and perplexing cases in the law (Vol. 105). Springer Science and Business Media.

Babaioff, M., Feldman, M., Nisan, N. and Winter, E., 2012.

Combinatorial agency. Journal of Economic Theory, 147(3), pp.999-1034.

Babaioff, M. and Winter, E., 2014. Contract complexity. EC, 14, p.911.

Baird, K., Jeong, S., Kim, Y., Burgstaller, B. and Scholz, B., 2019. The economics of smart contracts. arXiv preprint arXiv:1910.11143.

Bank of England., 2020. Central Bank Digital Currency: Opportunities, challenges and design.

Baron, D.P. and Myerson, R.B., 1982. Regulating a monopolist with unknown costs. Econometrica: Journal of the Econometric Society, pp.911-930.

Barton, T.D., Berger-Walliser, G. and Haapio, H., 2013. Visualization: Seeing contracts for what they are, and what they could become. JL Bus. and Ethics, 19, p.47.

Ben-Menahem, H., 1993. Leibniz on hard cases. ARSP: Archiv für Rechts-und Sozialphilosophie/Archives for Philosophy of Law and Social Philosophy, 79(2), pp.198-215.

Berg, C., Davidson, S. and Potts, J., 2019. Blockchain technology as economic infrastructure: Revisiting the electronic markets hypothesis. Frontiers in Blockchain, 2, p.22.

Bergemann, D. and Bonatti, A., 2019. Markets for information: An introduction. Annual Review of Economics, 11, pp.85-107.

Berglof, E., 1994. A control theory of venture capital finance. JL Econ. Org., 10, p.247.

Blackshear, S., Cheng, E., Dill, D.L., Gao, V., Maurer, B., Nowacki, T., Pott, A., Qadeer, S., Rain. Russi, D., Sezer, S., Zakian, T., Zhou, Runtian., 2020. Move: A language with programmable resources.

Bolton, P. and Dewatripont, M., 2005. Contract theory. MIT press.

Bommarito, M. J. (2021) "Preprocessing Data," in Katz, D. M., Dolin, R., and Bommarito, M. J. (eds) Legal Informatics. Cambridge: Cambridge University Press, pp. 55–60.

Bond, E.W. and Crocker, K.J., 1997. Hardball and the soft touch: the economics of optimal insurance contracts with costly state verification and endogenous monitoring costs. Journal of Public Economics, 63(2), pp.239-264.

Börgers, T. and Krahmer, D., 2015. An introduction to the theory of mechanism design. Oxford University Press, USA.

Brautigan, R., 1968. All watched over by machines of loving grace. TriQuarterly, 11, p.194.

Buterin, V., 2014. A next-generation smart contract and decentralized application platform: White Paper.

Cabinet Office., 2020. Model Services Contract Guidance.

Catlin, T. and Lorenz, J.T., 2017. Digital disruption in insurance: Cutting through the noise. Digit. McKinsey.

Cawley, J. and Philipson, T., 1999. An empirical examination of information barriers to trade in insurance. American Economic Review, 89(4), pp.827-846.

Caytas, J., 2017. Blockchain in the US regulatory setting: Evidentiary use in Vermont, Delaware, and elsewhere. Columbia Science and Technology Law Review (May 30, 2017).

Chen, L., Cong, L.W. and Xiao, Y., 2021. A brief introduction to blockchain economics. In Information for Efficient Decision Making: Big Data, Blockchain and Relevance, pp. 1-40.

Cheung, S.N., 1983. The contractual nature of the firm. The Journal of Law and Economics, 26(1), pp.1-21.

Chiappori, P.A., Jullien, B., Salanié, B. and Salanie, F., 2006. Asymmetric information in insurance: General testable implications. The RAND Journal of Economics, 37(4), pp.783-798.

Chiappori, P.A. and Salanié, B., 1997. Empirical contract theory: The case of insurance data. European Economic Review, 41(3-5), pp.943-950.

Chiu, J. and Koeppl, T.V., 2019. Blockchain-based settlement for asset trading. The Review of Financial Studies, 32(5), pp.1716-1753.

Clack, C.D., 2018. Smart Contract Templates: legal semantics and code validation. Journal of Digital Banking, 2(4), pp.338-352.

Clack, C.D., 2021. Languages for Smart and Computable Contracts. arXiv preprint arXiv:2104.03764.

Clack, C.D., Bakshi, V.A. and Braine, L., 2016. Smart contract templates: foundations, design landscape and research directions. arXiv preprint arXiv:1608.00771.

Clack, C.D. and McGonagle, C., 2019. Smart Derivatives Contracts: the ISDA Master Agreement and the automation of payments and deliveries. arXiv preprint arXiv:1904.01461.

Clack, C.D. and Vanca, G., 2018, November. Temporal aspects of smart contracts for financial derivatives. In International Symposium on Leveraging Applications of Formal Methods (pp. 339-355). Springer, Cham.

Clément, M. (2019) "Smart Contracts and the Courts," in DiMatteo, L. A., Cannarsa, M., and Poncibò, C. (eds) The Cambridge Handbook of Smart Contracts, Blockchain Technology and Digital Platforms. Cambridge: Cambridge University Press (Cambridge Law Handbooks), pp. 271–287.

Colander, D., 1994. The art of economics by the numbers. New Directions in Economic Methodology, London: Taylor and Francis, pp.35-49.

Cong, L.W. and He, Z., 2019. Blockchain disruption and smart contracts. The Review of Financial Studies, 32(5), pp.1754-1797.

Cong, L.W. and Xiao, Y., 2019. Information cascades and threshold implementation. University of Chicago, Becker Friedman Institute for Economics Working Paper.

Crawford, V.P. and Sobel, J., 1982. Strategic information transmission. Econometrica: Journal of the Econometric Society, pp.1431-1451.

Croce, A., Grilli, L. and Murtinu, S., 2014. Venture capital enters academia: An analysis of university-managed funds. The Journal of Technology Transfer, 39(5), pp.688-715.

Currie, J. and Vidovich, L., 2009. The changing nature of academic work. The Routledge international handbook of higher education, pp.441-452.

Dewatripont, M., Legros, P. and Matthews, S.A., 2003. Moral hazard and capital structure dynamics. Journal of the European Economic Association, 1(4), pp.890-930.

Diamond, D.W., 1984. Financial intermediation and delegated monitoring. The review of economic studies, 51(3), pp.393-414.

Dow, D.R., 1992. Godel And Langdell–A Reply To Brown And Greenberg's Use Of Mathematics In Legal Theory. Hastings LJ, 44, p.707.

Dütting, P., Roughgarden, T. and Talgam-Cohen, I., 2019. Simple versus optimal contracts. In Proceedings of the 2019 ACM Conference on Economics and Computation (pp. 369-387).

Dutting, P., Roughgarden, T. and Talgam-Cohen, I., 2021. The complexity of contracts. SIAM Journal on Computing, 50(1), pp.211-254.

Ely, J.C., 2017. Beeps. American Economic Review, 107(1), pp.31-53.

Ely, J.C. and Szydlowski, M., 2020. Moving the goalposts. Journal of Political Economy, 128(2), pp.468-506.

Etzkowitz, H., Webster, A. and Healey, P. eds., 1998. Capitalizing knowledge: New intersections of industry and academia. suny Press.

European Commission., 2020. On Artificial Intelligence - A European approach to excellence and trust. Brussels.

Farrell, S., Machin, H. and Hinchliffe, R., 2017, February. Lost and found in smart contract translation—considerations in transitioning to automation in legal architecture. In UNCITRAL, Modernizing international trade law to support innovation and sustainable development. Proceedings of the congress of the United Nations commission on international trade law, Vol. 4, pp. 95-104.

Finkelstein, A. and McGarry, K., 2006. Multiple dimensions of private information: evidence from the long-term care insurance market. American Economic Review, 96(4), pp.938-958.

Fries, C.P., Kohl-Landgraf, P., Paffen, B., Weddigen, S., Del Re, L., Schütte, W., Bacher, D., Declara, R., Eichsteller, D., Weichand, F. and Streubel, M., 2019. Implementing a financial derivative as smart contract. Available at SSRN 3342785.

Gale, D. and Hellwig, M., 1985. Incentive-compatible debt contracts: The one-period problem. The Review of Economic Studies, 52(4), pp.647-663.

Gans, J.S., 2019. The fine print in smart contracts (No. w25443). National Bureau of Economic Research.

Gantz, D.A. and Schropp, S.A., 2009. Rice Age: Comments on the Panel Report in Turkey–Measures Affecting the Importation of Rice. World Trade Review, 8(1), pp.145-177.

Gao, Y., Chen, Y., Wang, C.Y. and Liu, K.R., 2012, November. Optimal contract design for ancillary services in vehicle-to-grid networks. In 2012 IEEE Third International Conference on Smart Grid Communications (SmartGridComm) (pp. 79-84). IEEE.

Gao, G., Han, X., Ning, L., Ting, H.F. and Zhang, Y., 2020, August. Robustness and Approximation for the Linear Contract Design. In International Conference on Algorithmic Applications in Management (pp. 273-285). Springer, Cham.

Gao, G., Han, X., Ning, L., Ting, H.F. and Zhang, Y., 2021. Principal–agent problem under the linear contract. Journal of Combinatorial Optimization, pp.1-16.

Ghazizadeh, E. and Sun, T., 2020, November. A Systematic Literature Review of Smart Contract Applications. In Proceedings of the Future Technologies Conference (pp. 877-888). Springer, Cham.

Goldman, M.B., Leland, H.E. and Sibley, D.S., 1984. Optimal nonuniform prices. The Review of Economic Studies, 51(2), pp.305-319.

Grigg, I., 2004, July. The ricardian contract. In Proceedings. First IEEE International Workshop on Electronic Contracting, 2004. (pp. 25-31). IEEE.

Grossman, S.J. and Hart, O.D., 1986. The costs and benefits of ownership: A theory of vertical and lateral integration. Journal of political economy, 94(4), pp.691-719.

Grossman, S.J. and Hart, O.D., 1992. An analysis of the principal-agent problem. In Foundations of Insurance Economics (pp. 302-340). Springer, Dordrecht.

Haapio, H. and Passera, S. (2021) "Contracts as Interfaces: Visual Representation Patterns in Contract Design," in Katz, D. M., Dolin, R., and Bommarito, M. J. (eds) Legal Informatics. Cambridge: Cambridge University Press, pp. 213–238.

Hagan, M. (2021) "Introduction to Design Thinking for Law," in Katz, D. M., Dolin, R., and Bommarito, M. J. (eds) Legal Informatics. Cambridge: Cambridge University Press, pp. 155–176.

Harley, B., 2017, Are Smart Contracts Contracts?. Clifford Chance.

Hart, O., 1995. Firms, contracts, and financial structure. Clarendon press. Hart, O., 2017. Incomplete contracts and control. American Economic Review, 107(7), pp.1731-52.

Hart, O. and Holmström, B., 1987, August. The theory of contracts. In Advances in economic theory: Fifth world congress (Vol. 1). Cambridge: Cambridge University Press.

Hart, O. and Moore, J., 1990. Property Rights and the Nature of the Firm. Journal of political economy, 98(6), pp.1119-1158.

Hart, O. and Moore, J., 1998. Default and renegotiation: A dynamic model of debt. The Quarterly Journal of Economics, 113(1), pp.1-41.

Hartline, J.D., 2013. Mechanism design and approximation. Book draft. October, 122.

Hermalin, B.E. and Katz, M.L., 1991. Moral hazard and verifiability: The effects of renegotiation in agency. Econometrica: Journal of the Econometric Society, pp.1735-1753.

Hill, C.A., 2001. Why contracts are written in legalese. Chi.-Kent L. Rev., 77, p.59.

Holden, R. and Malani, A., 2014. Renegotiation Design by Contract. The University of Chicago Law Review, 81(1), pp.151-178.

Holden, R.T. and Malani, A., 2019. Can blockchain solve the hold-up problem in contracts? (No. w25833). National Bureau of Economic Research.

Holmström, B., 1979. Moral hazard and observability. The Bell journal of economics, pp.74-91.

Holmström, B., 2017. Pay for performance and beyond. American Economic Review, 107(7), pp.1753-77.

Horner, J. and Lambert, N.S., 2016. Motivational ratings.

ISDA and Linklaters., 2017. Whitepaper: Smart Contracts and Distributed Ledger - A Legal Perspective.

Kamenica, E. and Gentzkow, M., 2011. Bayesian persuasion. American Economic Review, 101(6), pp.2590-2615.

Kang, J., Xiong, Z., Niyato, D., Yu, H., Liang, Y.C. and Kim, D.I., 2019a, August. Incentive design for efficient federated learning in mobile networks: A contract theory approach. In 2019 IEEE VTS Asia Pacific Wireless Communications Symposium (APWCS) (pp. 1-5). IEEE.

Kang, J., Xiong, Z., Niyato, D., Xie, S. and Zhang, J., 2019b. Incentive mechanism for reliable federated learning: A joint optimization approach to combining reputation and contract theory. IEEE Internet of Things Journal, 6(6), pp.10700-10714.

Katz, A., 1990. The strategic structure of offer and acceptance: game theory and the law of contract formation. Michigan Law Review, 89(2), pp.215-295.

Katz, A.W., 2005. Contractual incompleteness: A transactional perspective. Case W. Res. L. Rev., 56, p.169.

Katz, D. M., Dolin, R. and Bommarito, M. J. (eds) (2021) “Legal Informatics in the Industrial Context,” in Legal Informatics. Cambridge: Cambridge University Press, pp. 449–450.

Kelso, L.O., 1945. Does the Law Need a Technological Revolution. Rocky Mntn. L. Rev., 18, p.378.

Kerrigan, C. 2019. The Financing of Intangible Assets: TMT Finance and Emerging Technologies. LexisNexis.

Kitcher, P., 2003. Science, truth, and democracy. Oxford: Oxford University Press.

Kowalski, R. and Datoo, A., 2020. Logical English Meets Legal English for Swaps and Derivatives.

Laffont, J.J. and Tirole, J., 1993. A theory of incentives in procurement and regulation. MIT press.

Lahaie, S., Pennock, D., Saberi, A., and Vohra, R. (2007). Sponsored Search Auctions. In N. Nisan, T. Roughgarden, E. Tardos, and V. Vazirani (Eds.), Algorithmic Game Theory (pp. 699-716). Cambridge: Cambridge University Press.

Lauslahti, K., Mattila, J. and Seppala, T., 2017. Smart contracts–How will blockchain technology affect contractual practices?. Etla Reports, (68).

Leibniz, G.W., 1989. Dissertation on the Art of Combinations. In Philosophical Papers and Letters (pp. 73-84). Springer, Dordrecht.

Lerner, J., Leamon, A. and Hardymon, F., 2012. Venture capital, private equity, and the financing of entrepreneurship. Hoboken (N.J.): Wiley.

Li, W., Ashlagi, I. and Lo, I., 2020. Simple and Approximately Optimal Contracts for Payment for Ecosystem Services. Available at SSRN 3754041.

Lim, W.Y.B., Huang, J., Xiong, Z., Kang, J., Niyato, D., Hua, X.S., Leung, C. and Miao, C., 2021. Towards federated learning in uav-enabled internet of vehicles: A multi-dimensional contract-matching approach. IEEE Transactions on Intelligent Transportation Systems.

Lind, H. and Nyström, J., 2011. The Explanation of Incomplete Contracts in Mainstream Contract Theory: A Critique of the Distinction between "Observable" and "Verifiable". Evolutionary and Institutional Economics Review, 7(2), pp.279-293.

Loevinger, L., 1948. Jurimetrics–The Next Step Forward. Minn. L. Rev., 33, p.455.

Malone, T.W., Yates, J. and Benjamin, R.I., 1987. Electronic markets and electronic hierarchies. Communications of the ACM, 30(6), pp.484-497.

Maskin, E., 1999. Nash equilibrium and welfare optimality. The Review of Economic Studies, 66(1), pp.23-38.

Maskin, E. and Tirole, J., 1999. Unforeseen contingencies and incomplete contracts. The Review of Economic Studies, 66(1), pp.83-

114.

Massarotto, G. and Ittoo, A., 2021. Gleaning Insight from Antitrust Cases Using Machine Learning.

McLeish, T. and Strang, V., 2014. Leading interdisciplinary research: transforming the academic landscape. Stimulus paper (No. ST-28). Leadership Foundation for Higher Education.

Meyer, M., 2003. Academic entrepreneurs or entrepreneurial academics? Research–based ventures and public support mechanisms. Randd Management, 33(2), pp.107-115.

Milgrom, P.R., 1981. Good news and bad news: Representation theorems and applications. The Bell Journal of Economics, pp.380-391.

Mirlees, J., 1971. An exploration into the theory of optimal taxation. Review of Economic Studies, 38.

Moore, J. and Repullo, R., 1988. Subgame perfect implementation. Econometrica: Journal of the Econometric Society, pp.1191-1220.

Myerson, R.B., 1981. Optimal auction design. Mathematics of operations research, 6(1), pp.58-73.

Nay, J. J. (2021) "Natural Language Processing for Legal Texts," in Katz, D. M., Dolin, R., and Bommarito, M. J. (eds) Legal Informatics. Cambridge: Cambridge University Press, pp. 99–113.

Perry-Kessaris, A., 2019. Legal Design for Practice, Activism, Policy, and Research. Journal of Law and Society 46, no. 2, p.185

Qiu, D. and Srikant, R., 2004. Modeling and performance analysis of BitTorrent-like peer-to-peer networks. ACM SIGCOMM computer communication review, 34(4), pp.367-378.

Rich, B., 2018. How AI is changing contracts. Harvard Business Review, 12.

Roughgarden, T. and Talgam-Cohen, I., 2019. Approximately optimal mechanism design. Annual Review of Economics, 11, pp.355-381.

Rutherford, D., 1996. Demonstration and reconciliation: the eclipse of the geometrical method in Leibniz's philosophy. Leibniz's "New System"(1695), ed. Roger S. Woolhouse. Florence: Olschki, pp.181-201.

Salanié, B., 2005. The economics of contracts: a primer. MIT press.

Scharrer, R., Pieciak, M., Reyes, C., Siciliano, D., 2021. Computable Contracts in Insurance — Fireside Chat. FutureLaw 2021.

Schrepel, T., 2021. Computational Antitrust: An Introduction and Research Agenda.

Schmitz, P.W., 2001. The hold-up problem and incomplete contracts: a survey of recent topics in contract theory. Bulletin of economic research, 53(1), pp.1-17.

Schwartz, A., 2000. Contract theory and theories of contract regulation.

Scott, R.E. and Stephan, P.B., 2006. The limits of leviathan: Contract theory and the enforcement of international law. Cambridge University Press.

Scott, R.E. and Triantis, G.G., 2005. Incomplete contracts and the theory of contract design. Case W. Res. L. Rev., 56, p.187.

Sergot, M.J., Sadri, F., Kowalski, R.A., Kriwaczek, F., Hammond, P. and Cory, H.T., 1986. The British Nationality Act as a logic program. Communications of the ACM, 29(5), pp.370-386.

Shermin, V., 2017. Disrupting governance with blockchains and smart contracts. Strategic Change, 26(5), pp.499-509.

Spence, M. and Zeckhauser, R., 1978. Insurance, information, and individual action. In Uncertainty in Economics (pp. 333-343). Academic Press.

Stiglitz, J.E., 1989. Principal and agent. In Allocation, information and markets (pp. 241-253). Palgrave Macmillan, London.

Stiglitz, J.E., 2017. The revolution of information economics: the past and the future (No. w23780). National Bureau of Economic Research.

Stremitzer, A., 2018. Agency theory: methodology, analysis: a structured approach to writing contracts (p. 241). Peter Lang International Academic Publishers.

Suliman, A., Husain, Z., Abououf, M., Alblooshi, M. and Salah, K., 2019. Monetization of IoT data using smart contracts. IET Networks, 8(1), pp.32-37.

Surden, H., 2012. Computable contracts. UCDL Rev., 46, p.629.

Surden, G., Goodenough, O., Reyes, C., Williams, M., Chun, A., Cummins, J., Gerenseth, M., 2021. Computable Contracts 2021. FutureLaw 2021.

Surminski, S., Bouwer, L.M. and Linnerooth-Bayer, J., 2016. How insurance can support climate resilience. Nature Climate Change,

6(4), pp.333-334.
Talgam-Cohen, I., Dütting, P., 2019. Contract Theory: A New Frontier for AGT. 20th ACM Conference on Economics and Computation (EC'19).
The CodeX Insurance Initiative Working Group., 2021., COMPUTABLE CONTRACTS AND INSURANCE: An Introduction.
Tinn, K., 2018. 'Smart' Contracts and External Financing. Available at SSRN 3072854.
Townsend, R.M., 1979. Optimal contracts and competitive markets with costly state verification. Journal of Economic theory, 21(2), pp.265-293.
Vickrey, W., 1961. Counterspeculation, auctions, and competitive sealed tenders. The Journal of finance, 16(1), pp.8-37.
Weatherall, D., 2000. Academia and industry: increasingly uneasy bedfellows. The Lancet, 355(9215), p.1574.
Winton, A., 1995. Costly state verification and multiple investors: the role of seniority. The Review of Financial Studies, 8(1), pp.91-123.
Williams, J., 1989. Ex-ante monitoring, ex-post asymmetry, and optimal securities. University of British Columbia, Faculty of Commerce and Business Administration.
Williamson, O., 1985. The economic institutions of capitalism. New York: The Free Pr.
Williamson, S.D., 1986. Costly monitoring, financial intermediation, and equilibrium credit rationing. Journal of Monetary Economics, 18(2), pp.159-179.
Williamson, O. and Masten, S., 1999. The economics of transaction costs. Edward Elgar Publishing.
Wolfram, S., 2016. Computational Law, Symbolic Discourse and the AI Constitution.
Wolfram, S., 2019. What We've Built Is a Computational Language (and That's Very Important!).
Wright, M., Lockett, A., Clarysse, B. and Binks, M., 2006. University spin-out companies and venture capital. Research policy, 35(4), pp.481-501.
Zhang, K., Mao, Y., Leng, S., He, Y., Maharjan, S., Gjessing, S., Zhang, Y. and Tsang, D.H., 2018. Optimal charging schemes for electric vehicles in smart grid: A contract theoretic approach. IEEE Transactions on Intelligent Transportation Systems, 19(9), pp.3046-

3058.

Zhang, Y., Pan, M., Song, L., Dawy, Z. and Han, Z., 2017. A survey of contract theory-based incentive mechanism design in wireless networks. IEEE wireless communications, 24(3), pp.80-85.