Available online at www.sciencedirect.com
SCIENCE DIRECT
Artificial Intelligence 164 (2005) 23–46
Artificial Intelligence
www.elsevier.com/locate/artint

The complexity of contract negotiation

Paul E. Dunne *, Michael Wooldridge, Michael Laurence

Department of Computer Science, University of Liverpool, Liverpool L69 7ZF, United Kingdom

Received 7 February 2003
Available online 14 March 2005

Abstract

The use of software agents for automatic contract negotiation in e-commerce and e-trading environments has been the subject of considerable recent interest. A widely studied abstract model considers the setting in which a set of agents have some collection of resources shared out between them and attempt to construct a mutually beneficial optimal reallocation of these by trading resources. The simplest such trades are those in which a single agent transfers exactly one resource to another—so-called 'one-resource-at-a-time' or '*O-contracts*'. In this research note we consider the computational complexity of a number of natural decision problems in this setting.

Keywords: Negotiation; Multiagent systems; Computational complexity

1. Introduction

Mechanisms for automatically negotiating the allocation of resources in a group of agents form an important body of work within the multiagent systems field. Typical abstract models derive from game-theoretic perspectives in economics and among the issues that have been addressed are strategies that agents may use to negotiate, e.g., [9,12,14], and protocols for negotiation in agent societies, e.g., [2,10].

In this paper, we investigate the computational complexity of one of the most fundamental questions that may be asked of such a negotiation setting: that of whether a particular

* Corresponding author.
E-mail address: ped@csc.liv.ac.uk (P.E. Dunne).

0004-3702/$ – see front matter
doi:10.1016/j.artint.2005.01.006

outcome is *feasible* under the assumption that negotiation participants will act rationally. The particular negotiation setting we consider—introduced by Sandholm [13]—relates to the reallocation of resources amongst agents. The idea is that, starting from some initial allocation, agents can negotiate to transfer resources between themselves to their mutual benefit. At each stage of negotiation, agents make deals by transferring resources to other agents, and receiving resources in return. The feasibility question in this setting may be informally understood as follows.

> Given some initial allocation P^s of resources to agents, and some potential final allocation P^t, is there a sequence of deals that will be individual rational to all involved, such that at the end of this sequence of deals, the allocation P^t will be realised?

It could be argued that a *positive* answer to this question does not imply that negotiation *will* be successful, as it merely implies the existence of an individual rational sequence of deals to get from P^s to P^t. The agents in question may have their own (perhaps irrational) reasons for rejecting some deals in this sequence. Moreover, unless the feasibility checking process is constructive, the agents may not be able to find the desired sequence of deals. A *negative* answer, however, surely rules out any chance of getting from P^s to P^t: for every possible sequence of deals realising this reallocation, some agent would suffer in the course of its implementation, and would therefore reject it.

Our main result is to show that this problem—and a number of natural variations of it—is NP-hard. We also investigate the complexity of a number of related problems: for example, we show that the problem of determining whether a particular allocation is Pareto Optimal is co-NP-complete.

2. Preliminary definitions

The scenario that we are concerned with is encapsulated in the following definition.

Definition 1. A *resource allocation setting* is defined by a triple $\langle \mathcal{A}, \mathcal{R}, \mathcal{U} \rangle$ where

$$\mathcal{A} = \{A_1, A_2, \dots, A_n\}; \qquad \mathcal{R} = \{r_1, r_2, \dots, r_m\}$$

are, respectively, a set of (at least two) agents and a collection of (non-shareable) resources. A *utility function*, u, is a mapping from subsets of $\mathcal{R}$ to rational values. Each agent $A_i \in \mathcal{A}$ has associated with it a particular utility function u_i, so that $\mathcal{U}$ is $\langle u_1, u_2, \dots, u_n \rangle$. An *allocation* P of $\mathcal{R}$ to $\mathcal{A}$ is a partition $\langle P_1, P_2, \dots, P_n \rangle$ of $\mathcal{R}$. The utility function, u_i, is *monotone* if $u_i(S) \leqslant u_i(T)$ whenever $S \subseteq T$. The value $u_i(P_i)$ is called the *utility* of the resources assigned to A_i.

Starting from some initial allocation—P_0—individual agents negotiate in an attempt to improve the utility of their holding. A number of interpretations have been proposed in order to define what constitutes a 'sensible' transfer of resource from both an individual agent's viewpoint and from the perspective of the overall allocation. Thus in negotiating a change from an allocation P_i to Q_i (with $P_i, Q_i \subseteq \mathcal{R}$ and $P_i \neq Q_i$) there are three possible outcomes for the agent A_i:

- $u_i(P_i) < u_i(Q_i)$ A_i values the allocation Q_i as superior to P_i;
- $u_i(P_i) = u_i(Q_i)$ A_i is indifferent between P_i and Q_i; and
- $u_i(P_i) > u_i(Q_i)$ A_i is worse off after the exchange.

In a setting in which agents are self-interested, in order for an agent to accept an exchange with the last outcome, the notion of a *pay-off* function is used: in order to accept the new allocation, A_i receives some payment sufficient to compensate for the resulting loss in utility. Of course, such compensation must be made by other agents in the system who in providing it do not wish to pay in excess of any gain in resource. In defining notions of 'pay-off', the interpretation is that in any transaction each agent A_i makes a payment, π_i: if $\pi_i < 0$ then A_i is given $-\pi_i$ in return for accepting a contract; if $\pi_i > 0$ then A_i contributes π_i to the amount to be distributed among those agents whose pay-off is negative. Formally, such a notion of 'sensible transfer' is captured by the concept of *individual rationality*.

Definition 2. Let $\langle \mathcal{A}, \mathcal{R}, \mathcal{U} \rangle$ be a resource allocation setting. A *deal* is a pair $\langle P, Q \rangle$ where $P = \langle P_1, \ldots, P_n \rangle$ and $Q = \langle Q_1, \ldots, Q_n \rangle$ are distinct partitions of $\mathcal{R}$. We use δ to denote an arbitrary deal. The effect of implementing the deal $\langle P, Q \rangle$ is that the allocation of resources specified by P is replaced with that specified by Q.

A deal $\langle P, Q \rangle$ is said to be *individually rational* (IR) if there is a *pay-off vector* $\pi = \langle \pi_1, \pi_2, \ldots, \pi_n \rangle$ satisfying,

(a) $\sum_{i=1}^{n} \pi_i = 0$.
(b) $u_i(Q_i) - u_i(P_i) > \pi_i$, for each agent A_i, *except that* π_i is allowed to be 0 if $P_i = Q_i$, i.e., should the deal $\langle P, Q \rangle$ leave the agent A_i with no change in its resource then it is not *required* that A_i be rewarded (have $\pi_i < 0$).

Definition 2 captures one view of a deal being 'sensible' with respect to the perspective of single agents. We require also concepts of '*global*' optimality. We consider two commonly used versions of this: Pareto Optimality and (Utilitarian) Social Welfare.

Definition 3. Let P be an allocation of $\mathcal{R}$ among $\mathcal{A}$. The *utilitarian social welfare* resulting from P, denoted $\sigma_u(P)$, is given by $\sum_{i=1}^{n} u_i(P_i)$.

The allocation P is *Pareto optimal* if for all allocations Q differing from P, it holds

$$\left(\bigvee_{i=1}^{n} \left[u_i(Q_i) > u_i(P_i) \right] \right) \Rightarrow \left(\bigvee_{i=1}^{n} \left[u_i(Q_i) < u_i(P_i) \right] \right). \tag{1}$$

Thus a Pareto optimal allocation is one in which no agent can attain better than its current utility except at the cost of leaving some agent worse off.

We make frequent use of the following result throughout the remainder of the paper.

Fact 4 [7]. *A deal $\langle P, Q \rangle$ is IR if and only if $\sigma_u(Q) > \sigma_u(P)$.*

In a typical application it is unlikely that an initial allocation P_0 to $\mathcal{A}$ will either maximise social welfare or be Pareto optimal, thus the agents involved seek to find a sequence of deals that will terminate in an optimal allocation. Given the setting it is clearly the case that there are allocations P_{opt} and Q_{opt} with the properties that $\sigma_u(P_{opt})$ maximises social welfare and for which Q_{opt} is Pareto optimal—of course, P_{opt} and Q_{opt} may not be unique. If the object is to maximise social welfare then clearly the deal $\langle P_0, P_{opt}\rangle$ will achieve this in a single round. It is unreasonable, however, to view such a deal as a viable solution: although always IR (if it represents a strict increase of social welfare) it is questionable whether it could be identified as the *first and only* deal required. The total number of possible allocations is n^m, and so for moderately large numbers of resources (m) there are too many feasibly to enumerate (even when $n = 2$). In addition, it may not be possible to implement the optimising contract in a *single* transaction even if only two agents are involved: the environment in which the trading process is implemented may not be suited to handling transactions in which large numbers of resources are involved; similarly, the protocol used for negotiation and contract description may not allow arbitrarily large numbers of resources to be dealt with.

In order to develop a realistic framework for negotiation, Sandholm [13] (using Smith's Contract-Net model [16]), presents a number of classes of *contract type*. In this article we are concerned with the following of these.

Definition 5 [13]. Let $\delta = \langle P, Q\rangle$ be a deal involving an allocation of $\mathcal{R}$ among $\mathcal{A}$. We say that δ is a *cluster contract* (C-contract) if there are distinct agents A_i and A_j for which,

(C1) $P_k = Q_k$ if and only if $k \notin \{i, j\}$.
(C2) There is a unique (non-empty) set S for which $Q_i = P_i \cup S$ and $Q_j = P_j \setminus S$ (with $S \subseteq P_j$) or $Q_j = P_j \cup S$ and $Q_i = P_i \setminus S$ (with $S \subseteq P_i$).

Thus a C-contract involves one agent transferring a subset of its allocation to another agent (without receiving any subset of resources in return).

The definition of C-contract permits an arbitrarily large number of resources to be transferred from one agent to another in a single deal. For the class of contracts of interest in our subsequent results, we wish to impose a bound on the maximum number of resources that can be moved in one deal. We thus introduce the notion of $C(k)$-*contracts*.

Definition 6. For a resource allocation setting $\langle \mathcal{A}, \mathcal{R}, \mathcal{U}\rangle$ and value $k \leqslant m = |\mathcal{R}|$, we say that δ is a k-*bounded cluster contract*, ($C(k)$-contract) if δ is a C-contract in which S—the set of resources transferred—contains *at most* k elements. When $k = 1$, we use the term *one contract* (O-contract): the name given to such deals in [13].

We recall that a $C(k)$-contract $\langle P, Q\rangle$ will be IR if and only if $\sigma_u(Q) > \sigma_u(P)$.

A sequence of deals $\Delta = \langle \delta_1, \delta_2, \ldots, \delta_t\rangle$ for which $\delta_i = \langle Q_{i-1}, Q_i\rangle$ is called a *contract path* realising the deal $\langle Q_0, Q_t\rangle$. The *length* of a contract path is the total number of deals comprising it. Given a predicate Φ over deals, we say that a contract path Δ is a Φ-*path* if $\Phi(\delta_i)$ is true of every deal δ_i within Δ.

Our main results concern Φ-paths where $\Phi(\delta)$ is the predicate which is true if and only if δ is an individually rational $C(k)$-contract. In the case of $k = 1$, i.e., IR O-contracts, such paths are attractive from an implementation viewpoint since these only involve agent-to-agent negotiation concerning a single resource at a time. In addition, starting from a given allocation, the number of O-contracts that are consistent with it is exactly $m(n-1)$, as opposed to n^m possible allocations. Thus heuristic methods may be able to find improved allocations by exploring the search space through O-contracts alone.

Appealing as the latter approach is, there are, nevertheless, problems associated with it. The following results were established by Sandholm [13].

Fact 7. *Let P_0 be any initial allocation of $\mathcal{R}$ to $\mathcal{A}$ and P_t be any other allocation.*

(a) *The deal $\langle P_0, P_t\rangle$ can always be realised by a contract path in which every deal is an O-contract.*
(b) *There are resource allocation settings, $\langle \mathcal{A}, \mathcal{R}, \mathcal{U}\rangle$ within which there are IR deals $\langle P_0, P_t\rangle$ that cannot be realised by any IR C-contract path.*

We note that Fact 7(b) holds even if we are concerned with settings involving only two agents and the allocation P_t concerned is one that maximises social welfare.

In total, IR C-contracts (and thereby also the more restricted IR $C(k)$ and IR O-contracts) in themselves may not suffice to form an IR contract-path realising a specific deal.

In this paper we are concerned with the following decision problem:

Definition 8. The decision problem *IR-k-path* (IR^k) is given by
Instance: A 5-tuple $\langle \mathcal{A}, \mathcal{R}, \mathcal{U}, P^s, P^t\rangle$ in which $\langle \mathcal{A}, \mathcal{R}, \mathcal{U}\rangle$ is a resource allocation setting, $P^{(s)}$ and $P^{(t)}$ are allocations of $\mathcal{R}$ to $\mathcal{A}$ in which $\sigma_u(P^{(t)}) > \sigma_u(P^{(s)})$.
Question: Is there an IR $C(k)$-contract path that realises the deal (P^s, P^t)?

It is important to note that the value k (which restricts the number of resources in a cluster contract), does *not* form part of an *instance* of IR^k.

In keeping with the use of the term O-contract for $C(1)$-contract, we denote the decision problem IR^1 by IRO.

The main results of this article concern IR^k when k is *constant* and IR^k when the cluster size (k) is a predefined function of the number of resources. Specifically we prove the following:

(a) IR^k is NP-hard for all constant values of k. This holds even when $\langle \mathcal{A}, \mathcal{R}, \mathcal{U}\rangle$ is a setting comprising two agents. The special case IRO remains NP-hard when both utility functions are monotone.
(b) For $k : \mathbb{N} \to \mathbb{N}$, satisfying $k(m) \leqslant m/3$, IR$^{k(m)}$ is NP-hard, again even in the case of resource allocation settings involving exactly two agents.
(c) IR$^{m/2}$ is NP-hard, again even in the case of resource allocation settings involving exactly two agents.

Our proofs of these results are given in Theorems 12–15.

We first note that the result of Theorem 15 does *not* imply (from the proof presented) either of the preceding theorems. It may seem to be the case that, when $h < k$, a lower bound on the complexity of IR^k implies a similar lower bound on the complexity of IR^h by virtue of the fact that within any resource allocation setting, all IR $C(h)$-contracts are also IR $C(k)$-contracts. As we shall, however, illustrate in proving (c), it is *not* necessarily the case that we can deduce IR^h to be NP-hard from a proof that IR^{h+k} is so: in order for this to hold, the construction used in demonstrating the latter must be such that any positive instances formed by the reduction to IR^{h+k} admit IR $C(h)$-contract paths. In the case of Theorem 14, while it is the case that our proof subsumes the result of Theorem 12, the construction for the latter case is rather less involved and has the additional advantage that the extension to monotone utility functions with IRO follows easily. For this reason, we have presented separate proofs of these results.

Before proceeding, we address one issue that is raised by Fact 7. Consider the following argument deriving from this fact.

(a) *Every* deal $\langle P_0, P_t\rangle$ can be realised by a sequence of O-contracts.
(b) There are IR deals which *cannot* be realised by a sequence of IR C-contracts.
(c) Therefore, to implement any IR deal $\langle P_0, P_t\rangle$ why not use an O-contract path some of whose constituent deals may fail to be IR?

In other words, why might it be necessary for *every* deal to be IR?

One answer to this question is offered by the scenario, outlined in [4], that we now describe. We observe that the issue underlying this argument is relevant with respect to any class of restricted contract types, i.e., the fact that O-contracts are referred to is purely for illustrative purposes. For simplicity, let us assume that we have a resource allocation setting $\langle \mathcal{A}, \mathcal{R}, \mathcal{U}\rangle$ involving exactly two agents $\{A_1, A_2\}$. These negotiate an allocation of $\mathcal{R}$ working with the following protocol.

> A reallocation of resources is agreed over a sequence of stages. Each stage consists of A_1 issuing a proposal to A_2 of the form (*buy*, r, p), offering to purchase r from A_2 for a payment of p; or (*sell*, r, p), offering to transfer r to A_2 in return for a payment p. The response from A_2 is simply *accept* (following which the exchange is implemented) or *reject*. A final allocation is fixed either when A_1 is 'satisfied' or as soon as A_2 *rejects* any offer.

This is, of course, a very simple negotiation setting; however, consider its operation when A_1 wishes to bring about an allocation P_t and can thus devise a plan—a sequence of O-contracts—to realise this from an initial allocation P_0.

While A_2 *could* be better off if P_t is realised, it may be the case that the only proposals A_2 will accept are those under which it does not lose, i.e., A_2 is not prepared to suffer a short-term loss even if it is suggested that a long-term gain will result. Thus if some agents are sceptical about the *bona fides* of others then they will be inclined to accept *only* deals from which they can perceive an *immediate* benefit, i.e., those which are individually rational.

There are several reasons why an agent may embrace such attitudes within the schema outlined: once a deal has been implemented A_2 may lose utility but no further proposals are made by A_1 so that its loss is 'permanent'. We note that even if we enrich the basic protocol so that A_1 can describe P_t to A_2 *before* any formal exchange of resources takes place, if $\langle P_0, P_t\rangle$ is implemented by an O-contract path (via the sequence of stages outlined), A_2 may still reject offers under which it suffers a loss, since it is unwilling to rely on the subsequent O-contracts that would ameliorate its loss actually being proposed.[1]

Although the position taken by A_2 in the setting just described may appear unduly cautious, we would claim that it clearly reflects actual behaviour in certain arenas. In contexts other than automated allocation and negotiation models in multiagent systems, there are many examples of actions by individuals where promised long-term gains are insufficient to engender the acceptance of short term loss, e.g., 'chain letter' schemes although having a natural lifetime bounded by the size of the population in which they circulate, typically break down before this is reached. Despite the possibility of significant gain after a temporary loss, recipients may be disinclined to invest the expense requested to propagate the chain: such behaviour is not seen as overly sceptical and cautious. In the same way, the 'rational' response to the widespread e-mail fraud by which one is asked to furnish bank account details and working capital in order to facilitate the release of significant funds in return for a percentage of these, is to ignore the request. As a final example, it is considered standard practice to delete without reading, unexpected e-mail attachments regardless of what incentives to open such may be promised by the accompanying message text.

In summary, the critical question underpinning such views is this: in a reallocation of resources conducted over a sequence of stages, should either agent suffer a loss in utility why should they have any 'confidence' that this loss will eventually be reversed? It is inevitable, in view of Fact 7(b) that there will sometimes be IR deals which, if implemented by a sequence of unrestricted O-contracts, will lead to such a loss for one agent.

In the scenario we have described, an agent A_1 wishing to realise an IR deal $\langle P_0, P_t\rangle$ with an extremely cautious agent A_2 faces the following dilemma: whether to formulate a plan to realise $\langle P_0, P_t\rangle$, e.g., an O-contract path, regardless of whether this path is IR; or whether to try and realise $\langle P_0, P_t\rangle$ by an IR O-contract path. In favour of the first option is the fact that such a plan can *always* be formulated; a problem will be, however, that the plan may never be implemented in full: A_2 may reject deals under which it suffers a loss or A_1 may suffer a loss which is never put right. The second alternative—construct an IR O-contract path—has in its favour the fact that neither agent has a *rational* motive to refrain from making or accepting offers until the allocation P_t has been effected. The drawback, however, is that it may not be possible to construct such a plan.

Nevertheless, it would seem reasonable for A_1, before resorting to adopting an arbitrary O-contract path, at least to determine if some IR O-contract path (or, more generally, some IR $C(k)$-contract path) does exist. One consequence of our results is that such an approach is unlikely to be computationally feasible.

[1] We note that even if A_1 attempts to construct an ordering under which any 'irrational' deal reduces the value of its own holding, there is one problem: A_2 may reject subsequent offers after the 'irrational' deals so that A_1 is worse off.

The next section of this article presents these results with conclusions and open questions raised in the final section.

3. Complexity results

Before proceeding with our results we describe our representation for typical instances in which resource allocation settings $\langle \mathcal{A}, \mathcal{R}, \mathcal{U} \rangle$ feature. The key issue here concerns the collection of utility functions $\mathcal{U}$ and how these should be encoded. A form in which the value attached to each subset of $\mathcal{R}$ is explicitly provided will result in an instance occupying space exponential in $|\mathcal{R}|$ and would not be considered reasonable in practice. On the other hand, using some encoding of $\mathcal{U}$ as a set of *Turing machine programs*, $\mathcal{M}$ say, it becomes necessary to assume certain properties in interpreting their computational behaviour, e.g., that the value of $u_i(S)$ as returned by the program M_i is defined from the content of M_i's tape after exactly some specified number of moves such as $|\mathcal{R}|$ since without such it would not be possible to establish membership in NP (or, indeed, any other complexity class).

Ideally, we wish a representation, $\rho(u)$, of the utility function $u : 2^{\mathcal{R}} \rightarrow \mathbb{Q}$ to satisfy the following informally phrased criteria:

(a) $\rho(u)$ is 'concise' in the sense that the length, e.g., number of bits, used by $\rho(u)$ to describe the utility function u within an instance is 'comparable' with the time taken by an optimal program that computes the value of $u(S)$.
(b) $\rho(u)$ is 'verifiable', i.e., given some binary word, w, there is an efficient algorithm that can check whether w corresponds to $\rho(u)$ for *some* u.
(c) $\rho(u)$ is 'effective', i.e., given $S \subseteq \mathcal{R}$, the value $u(S)$ can be efficiently computed from the description $\rho(u)$.

It is, in fact, possible to identify a representation form that satisfies all three of these criteria: we represent each member of $\mathcal{U}$ in a manner that does not *require* explicit enumeration of each subset of $\mathcal{R}$ and allows (a) to be met; uses a 'program' form whose syntactic correctness can be efficiently verified, hence satisfying (b); and for which termination in time linear in the program length is guaranteed, so meeting the condition set by (c). The class of programs employed are the so-called *straight-line programs*, which have a natural correspondence with combinational logic networks [3].

Definition 9. An (m, s)-*combinational network* C is a directed acyclic graph in which there are m *input nodes*, Z_m, labelled $\langle z_1, z_2, \ldots, z_m \rangle$ all of which have in-degree 0. In addition, C has s *output nodes*, called the *result vector*. These are labelled $\langle t_{s-1}, t_{s-2}, \ldots, t_0 \rangle$, and have out-degree 0. Every other node of C has in-degree at most 2 and out-degree at least 1. Each non-input node (*gate*) is associated with a Boolean operation of at most two arguments.[2] We use $|C|$ to denote the number of *gate* nodes in C. Any Boolean instantiation

[2] In practice, we can restrict the Boolean operations employed to those of binary conjunction ($\wedge$), binary disjunction ($\vee$) and unary negation ($\neg$).

of the input nodes to $\alpha \in \langle 0,1\rangle^m$ naturally induces a Boolean value at each gate of C: if h is a gate associated with the operation θ, and $\langle g_1, h\rangle$, $\langle g_2, h\rangle$ are edges of C then the value $h(\alpha)$ is $g_1(\alpha)\theta g_2(\alpha)$. Hence α induces some s-tuple $\langle t_{s-1}(\alpha), \ldots, t_0(\alpha)\rangle \in \langle 0,1\rangle^s$ at the result vector. For the (m,s)-combinational network C and $\alpha \in \langle 0,1\rangle^m$, this s-tuple is denoted by $C(\alpha)$.

Although often considered as a model of parallel computation, (m,s)-combinational networks yield a simple form of sequential program—straight-line programs—as follows. Let C be an (m,s)-combinational network to be transformed to a straight-line program, $\mathrm{SLP}(C)$, that will contain exactly $m + |C|$ lines. Since C is directed and acyclic it may be topologically sorted, i.e., each gate, g, given a unique integer label $\tau(g)$ with $1 \leqslant \tau(g) \leqslant |C|$ so that if $\langle g, h\rangle$ is an edge of C then $\tau(g) < \tau(h)$. The line l_i of $\mathrm{SLP}(C)$ evaluates the input z_i if $1 \leqslant i \leqslant m$ and the gate for which $\tau(g) = i - m$ if $i > m$. The gate labelling means that when g with inputs g_1 and g_2 is evaluated at $l_{m+\tau(g)}$ since g_i is either an input node or another gate its value will have been determined at l_j with $j < m + \tau(g)$.

Definition 10. Let $\mathcal{R}$ be as previously with $|\mathcal{R}| = m$, and u a mapping from subsets of $\mathcal{R}$ to rational values, i.e., a utility function. The (m,s)-network C^u is said to *realise* the utility function u if: for every $S \subseteq R$ with α_S the instantiation of Z_m by $z_i = 1$ if and only if $r_i \in S$, it holds

$$u(S) = \frac{val(C(\alpha_S))}{m}$$

where for $\beta = \langle \beta_{s-1}, \beta_{s-2}, \ldots, \beta_0\rangle \in \langle 0,1\rangle^s$, $val(\beta)$ is the whole number[3] whose s-bit binary expansion is β, i.e.,

$$val(\beta) = \sum_{i=0}^{s-1} \beta_i * 2^i,$$

where β_i is treated as the appropriate integer value from $\{0,1\}$.

These ideas allow any utility function u_i in $\mathcal{U}$ to be encoded using an appropriate (m,s_i)-combinational network, $C^{(i)}$ in such a way that $u_i(S)$ can be evaluated in time linear in the number of nodes in $C^{(i)}$ by determining the value of each gate under the related instantiation α_S and then dividing this value by m.

We give some concrete examples of this approach in the proof of Theorem 11. These are primarily intended to illustrate its feasibility and, having presented these, we will not complicate subsequent proofs with similarly detailed constructions. Regarding such constructions with respect to (a) of the representation criteria given, we note as a consequence of the simulations presented in [8,15] (see, e.g., Dunne [3, pp. 28–36]), that any deterministic algorithm with worst-case run-tine, $T(n)$ can be translated into a combinational

[3] Although this definition assumes utility functions to have non-negative values, were it the case that some function with $u(S) < 0$ was to be represented we can achieve this by using an additional output bit, $t_{\pm}$ to flag whether $val(C(\alpha))$ should be treated as positive ($t_{\pm} = 0$) or negative ($t_{\pm} = 1$).

network of size $T(n)\log T(n)$. It follows that from a high-level *algorithmic* description of how u_i is computed, an appropriate combinational network can be built.

The decision problem $\textsc{ir}^k$ concerns the existence of a suitable contract path from one allocation to another having greater social welfare. For completeness, it is useful to present three results concerning the existence of resource allocations meeting particular criteria. These problems are respectively,

Welfare Improvement (WI)
Instance: A tuple $\langle \mathcal{A}, \mathcal{R}, \mathcal{U}, P\rangle$ where $\mathcal{A}$, $\mathcal{R}$, and $\mathcal{U}$ are as before, and P is an allocation.
Question: Is there an allocation Q for which $\sigma_u(Q) > \sigma_u(P)$?

Welfare Optimisation (WO)
Instance: A tuple $\langle \mathcal{A}, \mathcal{R}, \mathcal{U}, K\rangle$ where $\mathcal{A}$, $\mathcal{R}$, and $\mathcal{U}$ are as before, and K is a rational number.
Question: Is there an allocation P for which $\sigma_u(P) \geqslant K$?

Pareto Optimal (PO)
Instance: A tuple $\langle \mathcal{A}, \mathcal{R}, \mathcal{U}, P\rangle$ as for WI.
Question: Is the allocation P Pareto optimal?

Kraus [9, p. 43] proves NP-hardness of a weaker form of the problem WO, whereby in addition to the total social welfare having to attain some specified value the allocation must be such that each agent accrues some designated guaranteed utility.

Theorem 11. *Even if* $|\mathcal{A}| = 2$ *and the utility functions are monotone*

(a) WI *is* NP-*complete.*
(b) WO *is* NP-*complete.*
(c) PO *is* CO-NP-*complete.*

Proof. We first demonstrate that the three problems are in the classes stated, recalling that the utility functions $\mathcal{U}$ are encoded by (m, s_i)-combinational networks $C^{(i)}$ as described in Definition 10. For (a), given an instance $\langle \mathcal{A}, \mathcal{R}, \mathcal{U}, P\rangle$ of WI simply non-deterministically guess an allocation $Q = \langle Q_1, \ldots, Q_n\rangle$ and compute

$$\sigma_u(Q) = \sum_{i=1}^{n} \frac{val(C^{(i)}(\alpha_{Q_i}))}{|\mathcal{R}|}$$

accepting if this exceeds $\sigma_u(P)$. For (b) a similar approach is used with an instance accepted if the guessed allocation Q has $\sigma_u(Q) \geqslant K$. Finally, for (c) we may use a CO-NP algorithm to check that for all allocations Q the Pareto Optimality condition given in Definition 3(1) holds.

We now prove NP-hardness for WI, WO and CO-NP-hardness for PO.

For part (a) we use a reduction from 3-SAT, instances of which are propositional formulae $\Phi(X_n)$ in conjunctive normal form with each clause of Φ defined by exactly three literals. Let

$$\Phi(X_n) = \bigwedge_{i=1}^{m} C_i = \bigwedge_{i=1}^{n} (y_{i,1} \vee y_{i,2} \vee y_{i,3})$$

be an instance of this problem, where $y_{i,j}$ is some literal x_k or $\neg x_k$.

Given $\Phi(X_n)$ we construct an instance $\langle \{A_1, A_2\}, \mathcal{R}, \langle u_1, u_2 \rangle, P \rangle$ in which

(a) $\mathcal{R} = \{x_1, x_2, \ldots, x_n, \neg x_1, \ldots, \neg x_n, C_1, \ldots, C_m\}$,
(b) $P = \langle \emptyset; \mathcal{R} \rangle$.

For W a set of literals, i.e.,

$$W \subseteq \{x_1, x_2, \ldots, x_n, \neg x_1, \neg x_2, \ldots, \neg x_n\}$$

we say that W is *useful for* $\Phi(X_n)$ if it satisfies *both* of the conditions below

(1) For each $1 \leqslant k \leqslant n$, W contains *at most one* of the literals $x_k, \neg x_k$.
(2) The partial instantiation of X_n under which each $y \in W$ is assigned true, i.e.,

$$x_i := \begin{cases} 1 & \text{if and only if } x_i \in W, \\ 0 & \text{if and only if } \neg x_i \in W, \end{cases}$$

satisfies $\Phi(X_n)$. Note that if neither $x_i \in W$ nor $\neg x_i \in W$ then this partial instantiation does not assign any value to x_i.

Now with $S \subseteq \mathcal{R}$, let $Lits(S)$ be the set

$$Lits(S) = S \cap \{x_1, x_2, \ldots, x_n, \neg x_1, \ldots, \neg x_n\}.$$

The utility functions $\langle u_1, u_2 \rangle$ are now given by,

$$u_1(S) = \begin{cases} 0 & \text{if } S = \emptyset, \\ \frac{|S|+1}{2n+m} & \text{if } Lits(S) \text{ is useful}, \\ \frac{|S|}{2n+m} & \text{if } Lits(S) \text{ is } not \text{ useful}, \end{cases}$$

$$u_2(S) = \begin{cases} 2 & \text{if } S = \mathcal{R}, \\ 1 + \frac{|S|}{2n+m} & \text{if } Lits(\mathcal{R} \setminus S) \text{ is useful}, \\ 1 + \frac{|S|-1}{2n+m} & \text{if } Lits(\mathcal{R} \setminus S) \text{ is not useful}. \end{cases}$$

Both of these are monotone. Furthermore given $\Phi(X_n)$ we may construct the combinational networks $C^{(1)}$ and $C^{(2)}$ as follows. Let the inputs for each network be $\langle z_1, \ldots, z_{2n+m} \rangle$ with z_i set to represent the presence of x_i (if $i \leqslant n$), the presence of $\neg x_{i-n}$ (if $n < i \leqslant 2n$) and the presence of C_{i-2n} if $(2n < i \leqslant 2n+m)$.

For $C^{(1)}$ we simply use a combinational network that computes the binary representation of $Useful(Z_{2n}) + \sum_{i=1}^{2n+m} z_i$ where

$$Useful(Z_{2n}) = \bigwedge_{i=1}^{n} (\neg z_i \vee \neg z_{n+i}) \ \wedge \ \bigwedge_{i=1}^{m} (z_{i,1} \vee z_{i,2} \vee z_{i,3}).$$

Here, $z_{i,j}$ is the variable from $\{z_1, \ldots, z_{2n}\}$ matching the literal $y_{i,j}$ of clause C_i. Thus, given S a subset of the literals over X_n, the term $(\neg z_i \vee \neg z_{n+i})$ in the corresponding instantiation induced over Z_{2n} will evaluate to $\top$ if and only if at most one of the literals $\{x_i, \neg x_i\}$ occurs in S. Similarly, for each clause $C_i = (y_{i,1} \vee y_{i,2} \vee y_{i,3})$ defining $\Phi(X_n)$ S contains at least one literal from C_i if and only if the term $(z_{i,1} \vee z_{i,2} \vee z_{i,3})$ evaluates to $\top$ for the instantiation of Z_{2n} defined from S.

The summation to compute the binary representation of the number of bits set to 1 within Z_{2n+m} can be carried out using the using the schema of Muller and Preparata [11], see, e.g., [3, pp. 112–114]. The whole number $val(C^1(\alpha_S))$ computed will be $|S|$, i.e., the number of variables set to 1 in α_S, if S is empty or not useful; and $|S| + 1$ if S is useful.

For $C^{(2)}$, a combinational network computes the binary representation of

$$\sum_{i=1}^{2n+m-1} 1 + \bigwedge_{i=1}^{2n+m} z_i + \sum_{i=1}^{2n+m} z_i + \mathit{Useful}(\neg z_1, \ldots, \neg z_n, \neg z_{n+1}, \ldots, \neg z_{2n}).$$

For $S \subseteq \mathcal{R}$, this will return $val(C^{(2)}(\alpha_S))$ as

$$\begin{aligned}
&4n + 2m = 2n + m - 1 + 1 + 2n + m + 0 \quad \text{when } S = \mathcal{R},\\
&2n + m + |S| = 2n + m - 1 + 0 + |S| + 1 \quad \text{when } \mathit{Lits}(\mathcal{R} \setminus S) \text{ is useful},\\
&2n + m + |S| - 1 = 2n + m - 1 + 0 + |S| + 0 \quad \text{when } \mathit{Lits}(\mathcal{R} \setminus S) \text{ is not useful}.
\end{aligned}$$

It is clearly the case that these descriptions can be constructed in polynomial-time from the formula $\Phi(X_n)$.

Now, noting that $\sigma_u(\langle \emptyset; \mathcal{R} \rangle) = 2$, we claim that there is an allocation, Q, having $\sigma_u(Q) > 2$ if and only if $\Phi(X_n)$ is satisfiable. To see this consider any non-empty $S \subseteq \mathcal{R}$ and the allocation $\langle S, \mathcal{R} \setminus S \rangle$ to $\langle A_1, A_2 \rangle$. We have,

$$\sigma_u\big(\langle S, \mathcal{R} \setminus S \rangle\big) = \begin{cases} \frac{|S|+1}{2n+m} + 1 + \frac{|\mathcal{R} \setminus S|}{2n+m} & \text{if } \mathit{Lits}(S) \text{ is useful},\\ \frac{|S|}{2n+m} + 1 + \frac{|\mathcal{R} \setminus S|-1}{2n+m} & \text{otherwise}. \end{cases}$$

In the former case we get, $\sigma_u(\langle S, \mathcal{R} \setminus S \rangle) = 2 + 1/(2n + m)$ and, in the latter, $\sigma_u(\langle S, \mathcal{R} \setminus S \rangle) = 2 - 1/(2n + m)$. Thus the allocation $\langle \emptyset, \mathcal{R} \rangle$ is welfare improvable if and only if there is an allocation S to A_1 for which $\mathit{Lits}(S)$ is useful: a condition that requires $\mathit{Lits}(S)$ to induce a satisfying instantiation of $\Phi(X_n)$, completing the proof that WI is NP-hard.

For part (b) we simply form the instance, $\langle \{A_1, A_2\}, \mathcal{R}, \langle u_1, u_2 \rangle, K \rangle$ with $\mathcal{R}$, $\langle u_1, u_2 \rangle$ as in part (a) and $K = 2 + 1/(2n + m)$.

For part (c), although continuing to employ a reduction from 3-SAT, we restrict instances of this to formulae that contain *exactly* n clauses, a variant shown to be NP-complete in [5, Theorem 2(b)]. We use $\mathcal{R}$ and $\langle u_1, u_2 \rangle$ as previously, but set $P = \langle P_1, P_2 \rangle = \langle \{C_1, \ldots, C_n\}, \{x_1, \ldots, x_n, \neg x_1, \ldots, \neg x_n\} \rangle$. In this case we have $u_1(P_1) = 1/3$ and $u_2(P_2) = 1 + (2n - 1)/(3n)$, so that $\sigma_u(P) = 2 - 1/(3n)$. We claim that this allocation is Pareto optimal if and only if $\Phi(X_n)$ is *unsatisfiable*. First suppose $\Phi(X_n)$ is unsatisfiable. Certainly for any allocation $Q = \langle S, \mathcal{R} \setminus S \rangle$ differing from $\langle P_1, P_2 \rangle$, it must be the case that $S = \emptyset$ or $\mathit{Lits}(S)$ is not useful. In the former case,

$$u_1(\emptyset) = 0 < u_1(P_1) = \frac{1}{3}$$

so that the Pareto Optimality condition of Definition 3(1) holds for $\langle P_1, P_2\rangle$ with respect to $\langle\emptyset, \mathcal{R}\rangle$.

If S is non-empty then

$$\sigma_u\big(\langle S, \mathcal{R} \setminus S\rangle\big) = u_1(S) + u_2(\mathcal{R} \setminus S) = 2 - \frac{1}{3n}$$

and so does not increase social welfare. It follows that, in this case,

$$\begin{gathered}\big(\big[u_1(S) > u_1(P_1)\big] \vee \big[u_2(\mathcal{R} \setminus S) > u_2(P_2)\big]\big)\\ \Rightarrow\\ \big(\big[u_1(S) < u_1(P_1)\big] \vee \big[u_2(\mathcal{R} \setminus S) < u_2(P_2)\big]\big).\end{gathered}$$

Hence if $\Phi(X_n)$ is unsatisfiable then P is Pareto optimal. On the other hand suppose $\Phi(X_n)$ *is* satisfiable. We can then demonstrate that P is *not* Pareto optimal by considering any set of literals $\{y_1, \ldots, y_n\}$ whose instantiation to true satisfies Φ. With such a set consider the allocation

$$Q = \langle Q_1, Q_2\rangle = \big\langle\{y_1, \ldots, y_n\}, \{\neg y_1, \ldots, \neg y_n, C_1, \ldots, C_n\}\big\rangle.$$

Certainly $Lits(Q_1)$ is useful, therefore

$$\begin{aligned} u_1(Q_1) &= \frac{n+1}{3n} > u_1(P_1),\\ u_2(Q_2) &= 1 + \frac{2}{3} > u_2(P_2). \end{aligned}$$

We deduce that the allocation P is Pareto optimal if and only if $\Phi(X_n)$ is unsatisfiable. □

We now proceed with the main results of this paper, showing that deciding if an individually rational $C(k)$-contract path exists between two allocations, is NP-hard for all *constant* values of k and when k can be a predefined function of the size of the resource set. In all cases the results hold in setting involving exactly two agents.

Theorem 12. *For all constant, k,* IR^k *is* NP*-hard.*

Corollary 13. IRO *is* NP*-hard in resource allocation settings for which all utility functions are monotone.*

Theorem 14. *For $k : \mathbb{N} \to \mathbb{N}$ satisfying $k(m) \leqslant m/3$,* $\text{IR}^{k(m)}$ *is* NP*-hard.*

Theorem 15. $\text{IR}^{m/2}$ *is* NP*-hard.*

We have commented earlier on the relationship between these results and our reasons for presenting the proofs separately.

Before continuing it is noted that, in contrast to the complexity classifications for the three problems reviewed in Theorem 11, we do not present *upper bounds* for any of the cases considered: we prove NP-hardness but not NP-*completeness*, i.e., do not present algorithms establishing membership in NP.

Some comments on this point are in order, particularly since there may appear to be an 'obvious' NP algorithm available, namely: guess a sequence of $C(k)$-contracts to realise $\langle P^s, P^t\rangle$ and check whether this defines an IR $C(k)$-contract path. This algorithm, however, may not be implementable[4] with an NP computation. For example, in the case of O-contracts, there may be a *unique* IR O-contract path realising the deal $\langle P^s, P^t\rangle$ but containing *exponentially many* (in m) O-contracts: such paths fail to provide the polynomial length certificate required for membership in NP. Constructions, in instances where only two agents are involved, are given in [4, Theorems 3, 4], for both unrestricted and monotone utility functions. Although not presented explicitly in [4], it is easy to extend these to IR $C(k)$-contracts for any constant k. Of course the 'obvious' algorithm we have outlined will be realisable in NP for resource allocation settings that satisfy certain criteria. One such criterion is that the number of *distinct* values which $\sigma_u(P)$ can take is polynomially-bounded in m: i.e., if $|\{w\colon \exists$ an allocation P for which $\sigma_u(P) = w\}| \leqslant m^p$. In such settings, no IR contract-path can contain more than m^p deals. Thus, if instances of IR^k are restricted to those for which σ_u has this property, then the corresponding decision problem is in NP. While this may seem to be a rather trivial example, we mention it since, as will be clear from the constructions presented in the proofs, the resource allocation settings formed have precisely this property: the number of distinct values that $\sigma_u(P)$ may take is $O(m)$. We can therefore deduce that, with such a restriction applying, the resulting decision problem *is* NP-complete. The question of upper bounds on the complexity of IR^k when arbitrary resource allocation settings may form part of an instance, remains, however, an open issue.

We now proceed with the proofs of Theorems 12 and 14.

Proof of Theorem 12. Given an instance $\Phi(X_n)$ of 3-SAT, we form an instance $T_\Phi = \langle \mathcal{A}, \mathcal{R}, \mathcal{U}, P^s, P^t\rangle$ of IR^k for which there is an IR $C(k)$-contract path realising $\langle P^s, P^t\rangle$ if and only if $\Phi(X_n)$ is satisfiable. Without loss of generality, it may be assumed that $n \geqslant 2k$ (recalling that k is constant). We use

$$\mathcal{A} = \{A_1, A_2\},$$
$$\mathcal{R} = \{x_1, x_2, \ldots, x_n, \neg x_1, \ldots, \neg x_n\},$$
$$P^s = \big\langle \emptyset; \{x_1, \ldots, x_n, \neg x_1, \ldots, \neg x_n\}\big\rangle,$$
$$P^t = \big\langle \{x_1, \ldots, x_n, \neg x_1, \ldots, \neg x_n\}; \emptyset\big\rangle,$$
$$u_2(S) = |S|.$$

In order to define the utility function, u_1 we need to extend our definition of a set of literals S being *useful*. We say that S is an *effective* set of literals for $\Phi(X_n)$ if both of the following hold.

(a) For each $1 \leqslant i \leqslant n$, S contains *at most one* of the literals $x_i, \neg x_i$.

[4] Our use of '*may not*', as opposed to the more emphatic '*cannot*', is intended: there is a rather subtle (and, at present, unresolved) technical complication that precludes the latter form. We discuss this issue further in Section 4.1 below.

(b) If Ψ_S is the sub-formula (defined on at most $n - |S|$ variables) that results from $\Phi(X_n)$ by applying the partial instantiation of X_n under which each $y \in S$ is assigned true[5] then Ψ_S is satisfiable.

We note that every *useful* set S for $\Phi(X_n)$ is also an *effective* set, however, the converse does not hold in general.

Given the definition of an effective set of literals, we now define

$$u_1(S) = \begin{cases} 2|S| & \text{if } |S| \leqslant n - k \text{ or } |S| > n, \\ 2|S| & \text{if } n - k < |S| \leqslant n \text{ and } S \text{ is effective for } \Phi(X_n), \\ |S| & \text{if } n - k < |S| \leqslant n \text{ and } S \text{ is } \textit{not} \text{ effective for } \Phi(X_n). \end{cases}$$

The key feature of this definition concerns how efficiently $u_1(S)$ can be represented: certainly whenever $|S| \leqslant n - k$ or $|S| > n$ this is easy. Similarly, for $|S|$ outside this range, it is straightforward to determine whether S contains a literal y and its negation $\neg y$. This leaves the case: $n - k < |S| \leqslant n$ where for each y, S contains *at most* one of the literals $\{y, \neg y\}$. For this, whether $u_1(S)$ is $2|S|$ or $|S|$ depends on the induced subformula Ψ_S from Φ and whether this is satisfiable. From our definition, Ψ_S is defined over *at most* $k - 1$ variables, and was induced from an instance of 3-SAT. It follows therefore that Ψ_S is a CNF formula on $k - 1$ variables each of whose distinct clauses contains between 0 and 3 literals. Since k is *constant*, we can construct a suitable combinational network to recognise satisfiable CNF of this form and with the size of this network being constant (albeit a constant value which may be exponential in k). For example with $k = 2$, the *unsatisfiable* CNF formulae on a single variable z are those containing an empty clause or containing *both* (z) and $(\neg z)$ as clauses.

This technical detail dealt with, we can proceed with the argument that $\Phi(X_n)$ is satisfiable if and only if T_Φ is a positive instance of IR^k.

First suppose that $\Phi(X_n)$ is satisfiable and let $\{y_1, \ldots, y_n\}$ be a set of n literals the instantiation of each to true will satisfy $\Phi(X_n)$. Consider the sequence of $2n$ O-contracts, $\Delta = \langle \delta_1, \delta_2, \ldots, \delta_{2n} \rangle$, in which $\delta_i = \langle P^{(i-1)}, P^{(i)} \rangle$, $P^{(0)} = P^s$ and $P^{(r)}$ is

$$\begin{cases} \langle \{y_1, \ldots, y_r\}; \mathcal{R} \setminus \{y_1, \ldots, y_r\} \rangle & \text{if } r \leqslant n, \\ \langle \{y_1, \ldots, y_n, \neg y_1, \ldots, \neg y_{r-n}\};\ \mathcal{R} \setminus \{y_1, \ldots, y_n, \neg y_1, \ldots, \neg y_{r-n}\} \rangle & \text{if } r > n. \end{cases}$$

The O-contract path described by Δ realises $\langle P^s, P^t \rangle$. Furthermore each δ_i is IR:

$$\sigma_u(P^{(i-1)}) = 2(i-1) + (2n - i + 1) = 2n + i - 1,$$
$$\sigma_u(P^{(i)}) = 2i + (2n - i) = 2n + i,$$

and for each $n - k + 1 \leqslant i \leqslant n$, the set of literals $P_1^{(i)}$ held by A_1 is effective from the fact that $\{y_1, \ldots, y_n\}$ induces a satisfying instantiation for $\Phi(X_n)$.

On the other hand, suppose that $\Delta = \langle \delta_1, \delta_2, \ldots, \delta_r \rangle$ with $\delta_i = \langle P^{(i-1)}, P^{(i)} \rangle$, $P^{(0)} = P^s$ and $P^{(r)} = P^t$ is an IR $C(k)$-contract path. Since at most k literals feature in any deal, in order to progress from $P^{(s)}$, in which A_1 holds no literals, to $P^{(t)}$ in which A_1 holds $2n$

[5] I.e., Ψ_S is formed from the set of clauses in Φ by removing any clause $C = y \vee D$ and replacing $C = \neg y \vee D$ with D when $y \in S$.

literals, it must be the case that at some point, $\delta_i = \langle P^{(i-1)}, P^{(i)} \rangle$ we have $|P_1^{(i-1)}| \leqslant n-k$ and $n-k < |P_1^{(i)}| \leqslant n$. Letting $d_{(less)}$ denote the value $|P_1^{(i-1)}| - (n-k)$ and $d_{(more)}$ the value $n-k-|P_1^{(i)}|$ so that $0 \leqslant d_{(less)} < d_{(more)} \leqslant k$ for this deal δ_i,

$$\sigma_u(P^{(i-1)}) = 3n - k - d_{(less)},$$
$$\sigma_u(P^{(i)}) = \begin{cases} 3n - k + d_{(more)} & \text{if } P_1^{(i)} \text{ is effective,} \\ 2n & \text{if } P_1^{(i)} \text{ is not effective.} \end{cases}$$

Thus if $P_1^{(i)}$ is *not* an effective set then the deal δ_i is not IR: $\delta_{(less)} \leqslant k-1$, and so, $\sigma_u(P^{(i-1)}) \geqslant 3n - 2k + 1 > 2n$. We deduce that the existence of an IR $C(k)$-contract path implies that $\Phi(X_n)$ is satisfiable. □

In the special case when $k = 1$, i.e., the decision problem IRO, we have the result of Corollary 13.

Proof of Corollary 13. Using the reduction from 3-SAT to IR^k from the proof of Theorem 12 the utility function u_2 is clearly monotone but the function u_1 is not. If, however, we modify the definition of u_1 to become

$$u_1(S) = \begin{cases} 2|S| & \text{if } |S| \neq n, \\ 2n & \text{if } |S| = n \text{ and } S \text{ is useful,} \\ 2n-1 & \text{if } |S| = n \text{ and } S \text{ is not useful,} \end{cases}$$

then not only does the argument of Theorem 12 continue to hold but the utility function u_1 is now monotone. □

Our final result deals with the case of IR $C(k(m))$-contract paths. Thus the number of resources that could be transferred in a single deal is not bounded by some constant value, as in the case of O-contracts or $C(k)$-contracts in general, but is now limited by some function of the total number of resources within the setting. For example, suppose $k(m) = \lfloor \sqrt{m} \rfloor$: given $\mathcal{A} = \{A_1, A_2\}, \mathcal{U} = \langle u_1, u_2 \rangle$, in the resource allocation setting $\langle \mathcal{A}, \{r_1, r_2, r_3, r_4\}, \mathcal{U} \rangle$, a $C(k(m))$-contract can move up to two resources between agents in a single deal. In the same setting, but with $|\mathcal{R}| = 16$, $C(k(m))$-contracts can now transfer up to 4 resources in a single deal.

The fact that the bound on the number of resources allowed to feature in a single deal is no longer constant, means that the reduction employed in proving Theorem 12 cannot be applied in general: we need to be able to specify the utility function u_1 in such a way that from a given instance of 3-SAT an appropriate polynomial-size representation of u_1 can be built. In these proofs, we used the fact that k is constant to demonstrate that testing if a set of literals is effective for $\Phi(X_n)$ can be carried out by testing satisfiability of CNF formulae defined on at most $k-1$ variables, and thus a 'compact' description of u_1 was possible. Although this construction can be effected by a polynomial-time reduction provided that $k(m) = \mathrm{O}(\log m)$—since u_1 need recognise only polynomially many (in m) cases—the same device, however, cannot be used for functions such as $k(m) = \lfloor \sqrt{m} \rfloor$ since testing if S is effective requires testing satisfiability of CNF formulae defined on $\sqrt{n}$ variables.

In order to deal with this complication we need to modify our construction.

Proof of Theorem 14. We employ a reduction from 3-SAT restricted to instances in which the number of clauses is exactly n as in the proof of Theorem 11(c). Let

$$\Phi(X_n) = \bigwedge_{i=1}^{n} C_i = \bigwedge_{i=1}^{n} (y_{i,1} \vee y_{i,2} \vee y_{i,3}).$$

We construct $T_\Phi = \langle \mathcal{A}, \mathcal{R}, \mathcal{U}, P^s, P^t \rangle$ an instance of IR^n as follows.

$$\mathcal{A} = \{A_{lits}, A_{clse}\},$$
$$\mathcal{R} = \{x_1, \ldots, x_n, \neg x_1, \ldots, \neg x_n, C_1, \ldots, C_n\},$$
$$P^s = \big\langle \{C_1, \ldots, C_n\}; \{x_1, \ldots, x_n, \neg x_1, \ldots, \neg x_n\} \big\rangle,$$
$$P^t = \big\langle \{x_1, \ldots, x_n, \neg x_1, \ldots, \neg x_n\}; \{C_1, \ldots, C_n\} \big\rangle.$$

It remains to define the utility functions u_{lits} and u_{clse} for each agent. If we consider any subset S of $\mathcal{R}$, then this consists of a subset of $\{x_1, \ldots, x_n, \neg x_1, \ldots, \neg x_n\}$ (literals) together with a subset of $\{C_1, \ldots, C_n\}$ (clauses). For a given allocation we use Y_{lits} to denote the subset of literals held by A_{lits}. Similarly Y_{clse}, C_{lits}, C_{clse} will describe respectively: the set of literals held by A_{clse}, of clauses held by A_{lits} and clauses held by A_{clse}. The idea underlying the construction of these is that moving literals from A_{clse} to A_{lits} by $C(n)$-contracts, will *only* be IR if at some stage those literals held by A_{lits} define a satisfying instantiation of $\Phi(X_n)$ (by choosing values for the variables which make the corresponding literals true).

$$u_{lits}(Y_{lits} \cup C_{lits}) = \begin{cases} 0 & \text{if } |Y_{lits}| < n \text{ and } Y_{lits} \text{ is } \textit{not} \text{ useful for } \bigwedge_{C_j \in C_{clse}} C_j, \\ 0 & \text{if } |Y_{lits}| = n \text{ and } Y_{lits} \text{ is } \textit{not} \text{ useful for } \Phi(X_n), \\ 0 & \text{if } |Y_{lits}| > n \text{ and } C_{lits} \neq \emptyset, \\ |Y_{lits}| & \text{otherwise,} \end{cases}$$

$$u_{clse}(Y_{clse} \cup C_{clse}) = \begin{cases} 0 & \text{if } |Y_{lits}| < n \text{ and } Y_{lits} \text{ is } \textit{not} \text{ useful} \\ & \quad \text{for } \bigwedge_{C_j \in C_{clse}} C_j, \\ 0 & \text{if } |Y_{lits}| = n \text{ and } Y_{lits} \text{ is } \textit{not} \text{ useful for } \Phi(X_n), \\ 0 & \text{if } |Y_{lits}| > n \text{ and } C_{lits} \neq \emptyset, \\ |C_{clse}| & \text{otherwise.} \end{cases}$$

We note that $|\mathcal{R}| = 3n$ so our bound on cluster size allows at most n elements from $\mathcal{R}$ to feature in a single deal.

We claim that $\Phi(X_n)$ is satisfiable if and only if there is an IR $C(n)$-contract path realising the deal $\langle P^s, P^t \rangle$.

First suppose that $\Phi(X_n)$ is satisfiable and let $\langle y_1, \ldots, y_n \rangle$ be a set of literals the instantiation of each to true satisfies $\Phi(X_n)$. Consider the sequence of O-contracts, $\langle \delta_1, \ldots, \delta_r \rangle$ in which $\delta_i = \langle P^{(i-1)}, P^{(i)} \rangle$ and $P^{(0)} = P^s$, $P^{(r)} = P^t$, resulting from the algorithm below.

(1) $i := 1$; $j := 1$.
(2) $P^{(j)}$ is formed by moving the literal y_i from Y_{clse} (in $P^{(j-1)}$) to Y_{lits}.

(3) $j := j + 1$;
  (3.1) Let $\{D_1, \ldots, D_p\}$ be the clauses currently in C_{lits} in which y_i occurs.
  (3.2) The next p O-contracts move each $D \in \{D_1, \ldots, D_p\}$ from C_{lits} to C_{clse}.
  (3.3) $j := j + p$; $i := i + 1$;
(4) If $i \leqslant n$ repeat from step (2).
(5) The final n O-contracts transfer each literal $\neg y_i$ from Y_{clse} to Y_{lits}.

To see that this procedure constructs an IR O-contract path realising $\langle P^s, P^t\rangle$ it suffices to note that in the allocation $P^{(j)}$,

$$u_{lits}(Y_{lits}^{(j)} \cup C_{lits}^{(j)}) = |Y_{lits}^{(j)}|,$$
$$u_{clse}(Y_{clse}^{(j)} \cup C_{clse}^{(j)}) = |C_{clse}^{(j)}|.$$

Furthermore with each deal either the number of literals in Y_{lits} increases by exactly one or the number of clauses in C_{clse} increases by exactly one.

Thus, if $\Phi(X_n)$ is satisfiable then this instance T_Φ of IR^n is accepted.

For the converse implication, suppose Δ is a IR $C(n)$-contract path realising the deal $\langle P^s, P^t\rangle$: $\Delta = \langle \delta_1, \delta_2, \ldots, \delta_i, \ldots, \delta_r\rangle$ with $\delta_i = \langle P^{(i-1)}, P^{(i)}\rangle$, $P^{(0)} = P^s$, $P^{(r)} = P^t$, and $P^{(i)} = \langle Y_{lits}^{(i)} \cup C_{lits}^{(i)}, Y_{clse}^{(i)} \cup C_{clse}^{(i)}\rangle$.

Noting that $\sigma_u(P^s) = 0$, consider the first deal $\delta_i = \langle P^{(i-1)}, P^{(i)}\rangle$ in Δ for which the following are true: $C_{lits}^{(i-1)} \neq \emptyset$ and $C_{lits}^{(i)} = \emptyset$. Certainly there must be such a deal since the first condition is true of P^s while the second holds for P^t. Consider the various possibilities:

(a) $|Y_{lits}^{(i-1)}| > n$.
If such a case were to occur then $u_{lits}(Y_{lits}^{(i-1)} \cup C_{lits}^{(i-1)}) = 0$ and $u_{clse}(Y_{clse}^{(i-1)} \cup C_{clse}^{(i-1)}) = 0$: in $P^{(i-1)}$, A_{lits} holds a non-empty set to clauses together with more than n literals. This contradicts the assumption that Δ is IR since it leads to $\sigma_u(P^{(0)}) = \sigma_u(P^{(i-1)} = 0$. We note that we cannot have $i = 1$ because of the premise $|Y_{lits}^{(i-1)}| > n$.

(b) $|Y_{lits}^{(i-1)}| \leqslant n$.
Since δ_i is a transfer of resources from A_{lits} to A_{clse}, we have $Y_{lits}^{(i)} \subseteq Y_{lits}^{(i-1)}$: if the set $Y_{lits}^{(i-1)}$ is *not* useful for $\Phi(X_n)$ then this would give $\sigma_u(P^{(i)}) = \sigma_u(P^s)$ (since both contributing utilities would be 0). This contradicts the assumption that Δ is IR, hence in this case $Y_{lits}^{(i-1)}$ must be useful and thus Φ is satisfiable. □

In our final result we show that the bound on cluster size may be increased to $m/2$. The argument used in the proof differs in one significant aspect from those presented in Theorem 12 and Theorem 14: it does not allow a lower bound on the complexity of $\text{IR}^{m/2-d}$ ($d > 0$) to be deduced.

Proof of Theorem 15. We again use a reduction from 3-SAT, but without the restrictions on the number of clauses in instances employed in Theorem 14. Given $\Phi(X_n)$ an instance of 3-SAT, the instance T_Φ of $\text{IR}^{m/2}$ has,

$$\mathcal{A} = \{A_1, A_2\},$$
$$\mathcal{R} = \{x_1, \ldots, x_n, \neg x_1, \ldots, \neg x_n\},$$
$$P^s = \big\langle \emptyset; \{x_1, \ldots, x_n, \neg x_1, \ldots, \neg x_n\}\big\rangle,$$
$$P^t = \big\langle \{x_1, \ldots, x_n, \neg x_1, \ldots, \neg x_n\}; \emptyset\big\rangle.$$

The utility functions, $\langle u_1, u_2\rangle$ being

$$u_1(S) = \begin{cases} 0 & \text{if } |S| < n, \\ 0 & \text{if } |S| = n \text{ and } S \text{ is not useful for } \Phi(X_n), \\ n & \text{if } |S| = n \text{ and } S \text{ is useful for } \Phi(X_n), \\ |S| & \text{if } |S| > n, \end{cases}$$
$$u_2(S) = 0.$$

Noting that $|\mathcal{R}| = 2n$, we claim that $\Phi(X_n)$ is satisfiable if and only if there is an IR $C(n)$-contract path realising $\langle P^s, P^t\rangle$, i.e., T_Φ is a positive instance of IR^n.

Suppose that $\Phi(X_n)$ is satisfiable. Let $\{y_1, y_2, \ldots, y_n\}$ be a set of n literals the instantiation of each to true will satisfy $\Phi(X_n)$. Consider the sequence of $C(n)$-contracts, $\langle\delta_1, \delta_2\rangle$ below in which Y_j^i is the subset of $\mathcal{R}$ held by A_j after δ_i.

i	Y_1^i	Y_2^i	$u_1(Y_1^i)$	$u_2(Y_2^i)$
0	$\emptyset$	$\{y_1, \ldots, y_n, \neg y_1, \ldots, \neg y_n\}$	0	0
1	$\{y_1, \ldots, y_n\}$	$\{\neg y_1, \ldots, \neg y_n\}$	n	0
2	$\{y_1, \ldots, y_n, \neg y_1, \ldots, \neg y_n\}$	$\emptyset$	$2n$	0

This sequence is IR and realises the deal $\langle P^s, P^t\rangle$ as required.

Conversely, suppose that Δ is a IR $C(n)$-contract path realising the deal $\langle P^s, P^t\rangle$: $\Delta = \langle\delta_1, \delta_2, \ldots, \delta_i, \ldots, \delta_r\rangle$ with $\delta_i = \langle P^{(i-1)}, P^{(i)}\rangle$, $P^{(0)} = P^s$, $P^{(r)} = P^t$. Noting that $\sigma_u(P^{(0)}) = 0$, in order for δ_1 to be IR, we must have $\sigma_u(P^{(1)}) > 0$. This, however, can only happen if $|Y_1^1| \geqslant n$, and since δ_1 is a $C(n)$-contract, it therefore follows that $|Y_1^1| = n$. Such an allocation to A_1, however, will only yield $u_1(Y_1^1) > 0$ if the set Y_1^1 is useful for $\Phi(X_n)$, i.e., if $\Phi(X_n)$ is satisfiable. □

4. Further work and development

Our results presented over Theorems 12–15 above, have been concentrated on *lower bounds* on computational complexity. In total for a range of values of cluster size, the problem of deciding whether a particular resource allocation setting admits a rational $C(k)$-contract path between two specified allocations appears unlikely to admits a feasible algorithmic solution, even if the settings of interest comprise only two agents.

In this section we briefly consider approaches and open problems directed towards more positive results. Our review comprises two subsections, the first of which deals with a somewhat abstruse technical point alluded to earlier; the second outlining algorithmic approaches that might be used in tackling formulations of IRO as an 'optimisation' problem. Readers who are more interested in the algorithmic aspects may wish to proceed directly to the second subsection.

4.1. Upper bounds on IRO

We first consider the issue raised earlier, namely whether $\text{IR}^k \in \text{NP}$. The results of [4, Theorems 3, 4], whereby positive instances of IRO in two agent settings are constructed in which the unique witnessing IR O-contract path has length exponential in m, may appear to disqualify the obvious 'guess and verify' algorithm from being realisable in NP. This reasoning, however, does not take into account the fact that an instance of IRO contains not only the elements $\langle \mathcal{A}, \mathcal{R}, P^s, P^t \rangle$ but also an encoding of the collection of utility functions $\mathcal{U}$. While the constructions from [4] are exponentially long in terms of the former, it is far from clear whether these paths are also exponential in the length of an optimal straight-line programs for $\mathcal{U}$. It is this issue that raises the principal difficulty in inferring that the obvious algorithm cannot be realised in NP as a consequence of [4]. The concerns of [4] are in establishing 'extremal' properties, thus the utility functions constructed to these ends are highly artificial in nature: in particular, the question of optimal straight-line programs is not addressed (since this is not relevant in the context). In total, the following question is unresolved:

Question 1. Is there a polynomial-bound, $q()$ with which: if $T = \langle \mathcal{A}, \mathcal{R}, \mathcal{U}, P^s, P^t \rangle$ is a *positive* instance of IRO encoded, using the approach described above, in $|T|$ bits, then there is *always* some IR O-contract path realising $\langle P^s, P^t \rangle$ whose length is at most $q(|T|)$?

A *negative* answer would indicate that the obvious algorithm could not be implemented in NP: a result that would *not* rule out the possibility of $\text{IRO} \in \text{NP}$, but it would indicate that such an upper bound requires a structure *other than a witnessing contract-path* to serve as the polynomial-length certificate.

A *positive* answer to Question 1 is likely to be *extremely hard* to obtain: although we have remarked on the 'artificial' nature of the utility functions in [4] these are, nonetheless, well-defined. In consequence, a positive answer would imply that any straight-line program realising these functions has exponential length: to date the largest lower bound proved for a n-argument function within this model is $3n$ given in [1], [3, pp. 91–99].

4.2. Formulating IRO as an optimisation problem

We have considered properties of $C(k)$-contract paths from the perspective of deciding if paths meeting particular criteria *exist*: in these terms our results indicate that feasible algorithms are unlikely to be found. One possibility is to identify 'special cases' which admit tractable decision processes, e.g., recent work reported in [6] considers a class of resource allocation settings motivated from a 'task allocation' context: the resource set is viewed as a set of m locations, $\mathcal{C}$ with $d_{i,j}$ describing the 'cost' of moving between c_i and c_j; the utility that each agent assigns to any subset S of C is the total cost of a minimal spanning tree of S. There are also a number of related problems for which possible approximation techniques may be constructed. We consider one such problems in this section and outline a 'greedy' approach for it.

We begin by observing that if P^s and P^t are distinct allocations with $\sigma_u(P^t) > \sigma_u(P^s)$ then the length of any O-contract (whether or not such is individually rational) is at least

$$Diff(P^s, P^t) = \sum_{A_i \in \mathcal{A}} \left|\{r \in \mathcal{R}\colon\ r \in P_i^s \text{ and } r \notin P_i^t\}\right|.$$

That is, the total number of resources in $\mathcal{R}$ which have to reallocated from their original owner in P^s to a new owner in P^t. Recognising that it may not be possible to identify an IR O-contract path of length $Diff(P^s, P^t)$ to realise $\langle P^s, P^t\rangle$ motivates the problem of finding an O-contract path that achieves this minimal length *and* has the fewest number of *irrational* deals among such paths. More formally,

Definition 16. The problem *Minimal Irrationality* (MI) takes as an instance a resource allocation setting $\langle \mathcal{A}, \mathcal{R}, \mathcal{U}\rangle$ and allocations P^s, P^t of $\mathcal{R}$ to $\mathcal{A}$. The value returned by $\text{MI}(\mathcal{A}, \mathcal{R}, \mathcal{U}, P^s, P^t)$ is

$$\min\bigl\{k\colon\ \exists \text{ an } O\text{-contract path, } \Delta = \langle \delta_1, \ldots, \delta_r\rangle, \text{ of length } Diff(P^s, P^t) \\ \text{realising } \langle P^s, P^t\rangle \text{ and on which there are at most } k \text{ deals, } \delta_i, \\ \text{that are } \textit{not} \text{ individually rational}\bigr\}.$$

It is, of course, an immediate consequence of Theorem 12 and Corollary 13 that the *decision problem* form of MI (in which the upper bound on the number of permitted irrational deals, k, occurs as part of an instance) is NP-complete: use the bound $k = 0$ and the reduction of Corollary 13 noting that if the deal $\langle P^s, P^t\rangle$ can be realised by an IR O-contract path of length $Diff(P^s, P^t)$ if and only if the CNF from which the instance is formed is satisfiable.

Suppose we regard MI as a (partial) function[6] whose domain comprises resource allocation settings $T = \langle \mathcal{A}, \mathcal{R}, \mathcal{U}\rangle$ and pairs of allocations $\langle P^s, P^t\rangle$ as given in Definition 16, and whose range is $\mathbb{N}$. We may re-interpret the result of [13] given in Fact 7 as indicating: $\text{MI}(T, \langle P^s, P^t\rangle) \leqslant Diff(P^s, P^t)$, i.e., there is always some O-contract path of length $Diff(P^s, P^t)$ available; and, there are instances for which $\text{MI}(T, \langle P^s, P^t\rangle) > 0$, i.e., there deals which cannot be realised by any IR O-contract path. In total, [13] gives

$$\forall \langle T, P^s, P^t\rangle\colon \text{MI}\bigl(T, \langle P^s, P^t\rangle\bigr) \leqslant Diff(P^s, P^t),$$
$$\exists \langle T, P^s, P^t\rangle\colon \text{MI}\bigl(T, \langle P^s, P^t\rangle\bigr) \geqslant 1.$$

It is a trivial matter to obtain exact bounds improving these to

$$\forall \langle T, P^s, P^t\rangle\colon \text{MI}\bigl(T, \langle P^s, P^t\rangle\bigr) \leqslant Diff(P^s, P^t) - 1,$$
$$\exists \langle T, P^s, P^t\rangle\colon \text{MI}\bigl(T, \langle P^s, P^t\rangle\bigr) \geqslant Diff(P^s, P^t) - 1.$$

For the upper bound simply note that since $\sigma_u(P^t) > \sigma(P^s)$ there must be *at least one* IR O-contract on any O-contract path of minimal length realising $\langle P^s, P^t\rangle$. For the lower

[6] 'Partial' since it is convenient to regard its value as undefined when $\sigma_u(P^t) \leqslant \sigma_u(P^s)$.

bound, use any $\langle T, P^s, P^t\rangle$ under which $\sigma_u(P^t) = 1$ and $\sigma_u(P) = 0$ for all allocations P differing from P^t.

While the behaviour of $\textsc{mi}(T, \langle P^s, P^t\rangle)$ from a general perspective is of some interest, e.g., studies of its value 'on average', such investigations are outside the scope of this note. Our main interest here will be to outline a heuristic aimed at constructing O-contract paths which attain the optimal value.

To simplify the presentation we shall assume that exactly two agents are involved, noting that the development to more than two is straightforward. We present the algorithm and then discuss the thinking underpinning it

Input: $\langle\{A_1, A_2\}, \mathcal{R}, \{u_1, u_2\}, P^s, P^t\rangle$
returns O-contract path of length $\mathit{Diff}(P^s, P^t)$ realising $\langle P^s, P^t\rangle$
$Q := P^s$; $i := 1$;
while $Q \neq P^t$ **loop**
 Choose $p \in Q_1 \setminus P^t_1 \cup Q_2 \setminus P^t_2$ such that the allocation V formed by moving p from A_1 to A_2 (if $p \in Q_1$) or from A_2 to A_1 (if $p \in Q_2$) has the following properties:
  P1 $\sigma_u(V) > \sigma_u(Q)$.
  P2 $\sigma_u(V) - \sigma_u(Q)$ is *minimal* among possible choices that satisfy P1.
  P3 If no choice of $p \in Q_1 \setminus P^t_1 \cup Q_2 \setminus P^t_2$ that satisfies P1 is possible, i.e., $\forall V \sigma_u(V) \leqslant \sigma_u(Q)$ then choose any V for which the value $\sigma_u(Q) - \sigma_u(V)$ is *maximised*.
 $\delta_i := \langle Q, V\rangle$;
 output δ_i;
 $Q := V$; $i := i + 1$;
end loop

It is not difficult to see that the sequence, $\langle\delta_1, \ldots, \delta_r\rangle$, that is output by this algorithm describes an O-contract path of length $r = \mathit{Diff}(P^s, P^t)$: some deal is chosen via (P1–P3); this deal *is* an O-contract; and, since the choice made is in terms of the current allocation (Q) with respect to the final allocation (P^t), it follows that $r = \mathit{Diff}(P^s, P^t)$.

The motivation for the algorithm is the following: given that $\sigma_u(P^t) > \sigma_u(P^s)$ and that the O-contract path to be formed must have minimal length, i.e., $\mathit{Diff}(P^s, P^t)$, the aim is to implement as many 'small increases' in σ_u within a minimal length path. Of course it may happen that a point, Q, is reached where *every* successor O-contract will result in σ_u not being increased. Rather than attempt to minimise any loss, the algorithm does the opposite: P3 implements the deal which *maximises* the loss of welfare. The idea being that the remaining O-contracts (particularly as the subsequent increments in σ_u are kept minimal) will be 'more likely' to be IR as a result.

We outline this approach merely to indicate that there may be reasonable approximation techniques for the class of problems which have been our principal interest. We will not present a detailed analysis of this algorithm's performance: such studies—both experimental and analytic—of this method and several variations are the topic of continuing work.

5. Conclusion

We have considered a number of decision problems that naturally arise from the multiagent contract negotiation models promoted by (among others) [7,13]. In summary, if contracts are restricted to those in which a limited number of resources can be transferred from one agent to another and are required to be rational (in the sense of strictly improving overall worth of an allocation), then not only is it the case that a suitable contract-path to an optimal allocation may fail to exist (as already shown in [13]), but even *deciding* if a path from a given allocation to a specified more beneficial allocation is possible, is intractable. There are a number of directions in which the results above could be developed. The requirement for individuals deals in a contract-path to be IR could be relaxed so that a limited number of 'irrational' deals are permitted, provided that the allocation eventually reached improves upon the initial allocation. Alternatively, we could consider contracts in which deals permitting an *exchange* of resources between two agents are allowed—the so-called *swap* or S-contracts of [13]. We conjecture, however, that even these degrees of freedom will continue to yield decision questions that are intractable.

Acknowledgement

The work reported in this article was carried out under the support of EPSRC Grant GR/R60836/01.

References

[1] N. Blum, A Boolean function requiring $3n$ network size, Theoret. Comput. Sci. 28 (1984) 337–345.
[2] F. Dignum, M. Greaves (Eds.), Issues in Agent Communication, Lecture Notes in Comput. Sci., vol. 1916, Springer, Berlin, 2000.
[3] P.E. Dunne, The Complexity of Boolean Networks, Academic Press, New York, 1988.
[4] P.E. Dunne, Extremal behaviour in multiagent contract negotiation, J. AI Res. 23 (2005) 41–78.
[5] P.E. Dunne, A.M. Gibbons, M. Zito, Complexity-theoretic models of phase transitions in search problems, Theoret. Comput. Sci. 294 (2000) 243–263.
[6] P.E. Dunne, M.R. Laurence, M.J. Wooldridge, Tractability results for automatic contracting, in: Proc. ECAI'04, Valencia, 2004, pp. 1002–1003.
[7] U. Endriss, N. Maudet, F. Sadri, F. Toni, On optimal outcomes of negotiations over resources, in: Proc. Second Intnl. Joint Conf. on Autonomous Agents and Multiagent Systems (AAMAS-2003), 2003, pp. 177–184.
[8] M. Fischer, N.J. Pippenger, Relations among complexity measures, J. ACM 26 (1979) 361–381.
[9] S. Kraus, Strategic Negotiation in Multiagent Environments, MIT Press, Cambridge, MA, 2001.
[10] P. McBurney, S. Parsons, M. Wooldridge, Desiderata for argumentation protocols, in: Proc. First Intnl. Joint Conf. on Autonomous Agents and Multiagent Systems (AAMAS-2002), ACM Press, New York, 2002, pp. 402–409.
[11] D.E. Muller, F.P. Preparata, Bounds to complexities of networks for sorting and for switching, J. ACM 22 (1975) 195–201.
[12] J.S. Rosenschein, G. Zlotkin, Rules of Encounter, MIT Press, Cambridge, MA, 1994.
[13] T.W. Sandholm, Contract types for satisficing task allocation: I theoretical results, in: AAAI Spring Symposium: Satisficing Models, 1998.

[14] T.W. Sandholm, Distributed rational decision making, in: G. Weiß (Ed.), Multiagent Systems, MIT Press, Cambridge, MA, 1999, pp. 201–258.
[15] C.P. Schnorr, The network complexity and Turing machine complexity of finite functions, Acta Inform. 7 (1976) 95–107.
[16] R.G. Smith, The contract net protocol: high-level communication and control in a distributed problem solver, IEEE Trans. Comput. C-29 (12) (1980) 1104–1113.