A Game Theory Approach for Negotiating in Data Marketplaces

SOULMAZ GHEISARI*, JAIME OSVALDO SALAS, LUIS-DANIEL IBÁÑEZ, and GEORGE KONSTANTINIDIS, School of Electronics and Computer Science, University of Southampton, UK

We introduce a game-theoretic framework, together with a distributed algorithm, for automating multi-issue data-sharing negotiations. The framework models interactions between data providers and data consumers as a bargaining problem and defines a number of bargaining actions, negotiation parameter categories (such as numerical, hierarchical, or query-based) and associated classes of utility functions for negotiating on these parameters. We instantiate the framework for data-sharing in data marketplaces. For this real-world use case, we provide a set of concrete negotiation parameters such as properties of the dataset (e.g., format, price, or date), views or query-defined subsets, and even governance-related ones such as negotiating on the privacy policy or the terms-of-use of the dataset. We propose two heuristics to guide agents' behaviours, greedy and concession-based. For our framework and heuristics, we provide theoretical guarantees as well as an implementation evaluated on real-world datasets from bank marketing, gym workouts, and sensing applications. Compared to traditional negotiation approaches, such as monotonic concessions and take-it-or-leave-it protocols, our approach achieves higher agreement rates and utilities, while up to 49% improves utility balance, measured as fairness.

CCS Concepts: • **Theory of computation → Self-organization**; **Algorithmic game theory and mechanism design**; • **Information systems → Data exchange**; Web Ontology Language (OWL); • **Applied computing → Electronic data interchange**; • **Computing methodologies →** *Heuristic function construction*; *Multi-agent systems.*

Additional Key Words and Phrases: Agent, Bargaining Problem, Concession, Data Consumer, Data Provider, Data-Sharing, Game Theory, Greedy, Multi-Issue Negotiation

ACM Reference Format:
Soulmaz Gheisari, Jaime Osvaldo Salas, Luis-Daniel Ibáñez, and George Konstantinidis. 2026. A Game Theory Approach for Negotiating in Data Marketplaces. *Proc. ACM Manag. Data* 4, 1 (SIGMOD), Article 2 (February 2026), 27 pages. https://doi.org/10.1145/3786616

1 Introduction

In this paper, we address the problem of understanding, modelling, and automating multi-issue data-sharing negotiations, which are critical for enabling scalable, privacy-aware, and regulation-compliant data-sharing [15, 24, 32, 48].

Consider the example of a data-driven marketing campaign, where an ad-tech buyer requests data from a bank-marketing dataset for people who are between 30 and 45 years old (age $\in [30, 45]$) and who have technical or managerial jobs (job $\in \{\text{technician}, \text{admin}\}$), offering payment for that filtered view. However, the provider prefers to sell broader —ideally, full-table— access at a higher

*Corresponding author.

Authors' Contact Information: Soulmaz Gheisari, s.gheisari@soton.ac.uk; Jaime Osvaldo Salas, J.O.Salas@soton.ac.uk; Luis-Daniel Ibáñez, L.D.Ibanez@soton.ac.uk; George Konstantinidis, G.Konstantinidis@soton.ac.uk, School of Electronics and Computer Science, University of Southampton, Southampton, UK.

ACM 2836-6573/2026/2-ART2
https://doi.org/10.1145/3786616

price. The buyer favours *Advertising* (and can also accept *Direct Marketing*), but the provider limits the purpose to *Personalised Advertising*, caps the campaign duration at 90 days, and prohibits third-party sharing beyond the buyer's *company* role. Negotiations such as this one are typically manual, involving complex trade-offs over heterogeneous parameters, and rely on time-consuming, costly, ad-hoc, and error-prone processes which often lack standardisation.

We present a game-theoretic framework to model multi-issue negotiation in data-sharing and an instantiation, including an implementation of this framework to support semi- and/or fully-automated negotiation in data marketplaces. To the best of our knowledge, ours is the first game-theoretic framework in data-sharing negotiation that goes beyond a single negotiated parameter (usually just price). Indeed, we identify the need to support heterogeneous, multi-issue parameters, such as numerical, hierarchical, ontology-based, nominal, and query-based attributes. Negotiating these requires principled comparators or valuators for each parameter type (e.g., comparing two queries or two data processing purposes) and consistent aggregation into utilities [24, 39, 47].

To achieve this, we model data-sharing negotiations as a two-player bargaining problem with a distributed and sequential protocol. The negotiation setting involves a data provider (e.g., the bank-marketing dataset owner) and a data consumer (e.g., the ad-tech buyer) who take alternating turns in proposing and responding to multi-issue proposals over the strategic parameters [17, 75].

Each player tries to maximise their *payoff*, via a payoff function. At each turn, a player can evaluate the opponent's proposal (using a local *utility* function) and may either accept the other's last proposal, generate a new one (using a *proposal generation* function), reject the proposal as infeasible, or exit the negotiation. The game continues until both players accept a proposal (yielding an agreement), or until one party exits or no feasible proposal remains. If not terminated pre-maturely, the negotiation is structured to converge toward the Nash Bargaining Solution [23, 40, 61], which maximises the product of each player's payoff gains and guarantees Pareto optimality.

Our framework is abstract yet instantiable; it provides a reusable scaffold for practitioners to instantiate locally with domain-specific types, payoff, proposal and utility functions, without relying on central intermediators or rebuilding the protocol equilibrium and logic each time [17, 75]. We instantiate the framework for the domain of data marketplaces. Negotiated parameters in our instantiation include, beyond price, dataset/query views, a range of parameters on data usage policies such as usage control terms (purpose, action on data, duration, etc.), and even non-functional aspects (e.g., environmental footprint of dataset); these different types come from established protocols and vocabularies (e.g., IDSA, ODRL, DPV) and regulatory contexts (e.g., GDPR) [1, 19, 64, 72, 73].

Furthermore, we suggest and compare two heuristics to generate a proposal over the strategic parameters: greedy and concession. The greedy heuristic selects the closest acceptable proposal to the player's previous proposal, minimising internal deviation. The concession heuristic, in contrast, generates proposals closer to the opponent's best previous proposal, encouraging faster agreement through incremental compromise.

To validate our approach, we implemented the system and performed an empirical evaluation on real datasets. The results demonstrate that the system converges efficiently under both greedy and concession strategies, with the latter reaching agreements in fewer rounds on average. We demonstrate the superiority of our framework over traditional protocols like take-it-or-leave-it and monotonic concession [22, 50], achieving significantly higher agreement rates and utility gains. Additionally, we compare our framework against the centralised Nash Bargaining Solution Oracle (where computable) and present a Proof-of-Concept (PoC) to demonstrate the end-to-end feasibility of our approach.

In summary, our contributions are: (1) a formal, game-theoretic framework for automated, multi-issue data-sharing negotiation, uniquely supporting heterogeneous parameters like queries and hierarchical concepts; (2) a concrete instantiation for data marketplaces, including two novel proposal generation heuristics; (3) parameters and protocols aligned with standards (IDSA, ODRL, DPV) that practitioners can use in instantiations for their specific domain; (4) a theoretical analysis and proofs on convergence and computational complexity of our algorithms; and (5) a comprehensive experimental evaluation and PoC that demonstrates the feasibility and superiority of our system over established baselines.

2 Game Theory Background

In Game Theory, an *n-player game* [49], consists of a set of n players, strategies (or actions) for players, a set of strategy profiles (states in the game), and a payoff function for each player that evaluates the current state of the game.

Definition 2.1. An n-player game is a triple (P, Σ, Π), where:

- $P = \{p_1, p_2, \ldots, p_n\}$ is a set of n players.
- $\Sigma = \Sigma_1 \times \Sigma_2 \times \cdots \times \Sigma_n$ is a set of *strategy profiles*, where Σ_i is a set of strategies for the player p_i. A strategy profile is a vector $\bar{\sigma} = (\sigma_1, \sigma_2, \ldots, \sigma_n)$, where σ_i is a strategy (action) of Σ_i $(i = 1, 2, \ldots, n)$.
- $\Pi = (\pi_1, \pi_2, \ldots, \pi_n)$ is a vector of payoff functions. Where $\pi_i : \Sigma \longrightarrow \mathbb{R}$ is a real-valued payoff function for player p_i $(i = 1, 2, \ldots, n)$. The image of a payoff function in $\mathbb{R}$ is also known as a *payoff value.*

In general, players follow strategies that maximise their payoff in response to other players' strategies. A stable solution, where no player has an incentive to deviate from the current situation, is referred to as a *Nash equilibrium* (NE)[45]. To define NE, we first need to define what the *Best Response* to a strategy profile is [49]:

Definition 2.2. Let $\bar{\sigma}_{-i} = (\sigma_1, \sigma_2, \ldots, \sigma_{i-1}, \sigma_{i+1}, \ldots, \sigma_n)$ be a strategy profile without player p_i's strategy; the full strategy profile is written as $\bar{\sigma} = (\sigma_i, \bar{\sigma}_{-i})$. Player p_i's *best response* to $\bar{\sigma}_{-i}$ is a strategy $\sigma_i^* \in \Sigma_i$ such that: for all $\sigma_i \in \Sigma_i$, $\pi_i(\sigma_i^*, \bar{\sigma}_{-i}) \geq \pi_i(\sigma_i, \bar{\sigma}_{-i})$.

Definition 2.3. A strategy profile $\bar{\sigma}^* = (\sigma_1^*, \ldots, \sigma_n^*) \in \Sigma$ is a *Nash Equilibrium*, if for all i, σ_i^* is the best response to $\bar{\sigma}^*_{-i}$ for player p_i.

A Nash Equilibrium represents a state of individual strategic stability – no player can improve their outcome by unilaterally changing their strategy, assuming the others' strategies remain fixed. However, while NE helps identify stable decisions, it does not guarantee group efficiency. A *Pareto Optimal* [53] (PO) state is an NE where no player's situation can be improved without making another player worse off –in other words– guaranteeing social efficiency.

Definition 2.4. A strategy profile $\bar{\sigma}^* \in \Sigma$ is considered *Pareto Optimal* if there is no strategy profile $\bar{\sigma} \in \Sigma$ such that for all players i, $\pi_i(\bar{\sigma}) \geq \pi_i(\bar{\sigma}^*)$, there exists a player j such that $\pi_j(\bar{\sigma}) > \pi_j(\bar{\sigma}^*)$.

The *Pareto frontier* is the set of profiles that are Pareto Optimal.

In this paper we model the problem of data-sharing as a *two-player bargaining problem* [61] wherein two individuals have the opportunity to collaborate for mutual benefit in more than one way, such as through negotiation. No action taken by one individual without the consent of the other can affect the well-being of the other. The bargaining problem is also called a bilateral bargain or bilateral monopoly.

Definition 2.5. A *Two-Player Bargaining Game* is a quintuple $\mathcal{G} = (P, \Sigma, \Pi, \Theta, Q)$, where:

- (P, Σ, Π) is an n-player game with $n = 2$, and $P = \{p_1, p_2\}$.

- $\Theta = (\theta_1, \theta_2)$ is a pair of reservation values θ_1 and θ_2 for the players p_1 and p_2, respectively, where θ_i is the lowest acceptable payoff value for player i.
- $Q = (q_1, q_2)$ is a pair of payoff values $q_i \in \mathbb{R}$, called the *status quo values*, that the players p_1 and p_2 would receive, respectively, if they didn't bargain with the other player [27]. Status quo represents each player's fallback position, which could be a disagreement outcome (e.g., no deal) or a predefined baseline payoff.

A bargaining game can have multiple NE, and some that are PO depending on the players' strategies. However, the *Nash Bargaining Solution* (NBS) —a concept from cooperative game theory [49] — is a unique NE and PO solution for a given bargaining problem and is commonly considered as a fair solution. Every bargaining game that satisfies Nash axioms [61] has exactly one Nash bargaining solution on the PO frontier (the NBS). Solutions that are within the reservation bounds are called feasible and defined as:

Definition 2.6. A strategy pair $\bar{\sigma} = (\sigma_1, \sigma_2)$ is a *Feasible Solution* (FS) to a bargaining game, if: $\pi_1(\bar{\sigma}) \geq \theta_1$ and $\pi_2(\bar{\sigma}) \geq \theta_2$

Definition 2.7. A strategy pair $\bar{\sigma}^* = (\sigma_1^*, \sigma_2^*)$ is a *Nash Bargaining Solution* (NBS) or *fair solution*, if:

$$(\pi_1(\bar{\sigma}^*), \pi_2(\bar{\sigma}^*)) = \underset{(\bar{\sigma}) \in FS}{\arg\max}\ (\pi_1(\bar{\sigma}) - q_1)(\pi_2(\bar{\sigma}) - q_2),$$

where $\bar{\sigma}$ is a FS in the set of all feasible solutions, q_i is the status quo value of player p_i and $\pi_1(\bar{\sigma}) \geq q_1,$ and $\pi_2(\bar{\sigma}) \geq q_2$ [61].

Bargaining games that model real-life negotiations, such as the data negotiations that we address in this paper, are *sequential* and *distributed*, with each player independently taking turns to suggest a solution. At the same time, players have *incomplete information* [37]; not all game parameters of one player (e.g., costs, values, or constraints) are known to the other.
For example, the ad-tech buyer knows their maximum willingness to pay, and a bank-marketing data provider knows their minimum acceptable price, but neither knows the other's value. Therefore, each chooses a strategy (e.g., what to propose or ask) based on their beliefs about the other side.

To capture the alternating, distributed process of bargaining, we provide Algorithm 1 as a skeleton where both players iteratively exchange and revise proposals by calling function Play in line 1.

Algorithm 1 Distributed Sequential Bargaining

1: **function** Play($i, t, \sigma^{t-1}, \sigma^{t-2}$, maxProd, $\mathcal{G}$)
2: $\quad$ $j \leftarrow (2 - i) + 1$ // j is the other player
3: $\quad$ **if** $t == 0$ **then**
4: $\quad\quad$ $\sigma^t \leftarrow$ choose an action in Σ_1
5: $\quad$ **else**
6: $\quad\quad$ $\sigma^t \leftarrow$ BestResponse($i, \sigma^{t-1}, \mathcal{G}$)
7: $\quad\quad$ **if** $\sigma^t == \sigma^{t-2}$ **then**
8: $\quad\quad\quad$ **return** ((σ^t, σ^{t-1}), maxProd)
9: $\quad\quad$ **else**
10: $\quad\quad\quad$ $\upsilon \leftarrow \pi_1(\sigma^t, \sigma^{t-1})$
11: $\quad\quad\quad$ $\nu \leftarrow \pi_2(\sigma^t, \sigma^{t-1})$
12: $\quad\quad\quad$ **if** $(\upsilon - q_1)(\nu - q_2) >$ maxProd **then**
13: $\quad\quad\quad\quad$ maxProd $\leftarrow (\upsilon - q_1)(\nu - q_2)$
14: $\quad$ **return** Play($j, t + 1, \sigma^t, \sigma^{t-1}$, maxProd, $\mathcal{G}$)

The algorithm takes as inputs (line 1): the player identifier i (1 or 2), the current turn t (player p_1 plays at even t, and p_2 at odd t), the incoming action from the opponent σ^{t-1}, the player's own previous action σ^{t-2} (played two rounds earlier), the best joint payoff product so far, `maxProd` (the *Nash product* computed toward the NBS of Def. 2.7), and the remaining game parameters $\mathcal{G}$.

The game starts with $\textsc{Play}(1, 0, \texttt{NULL}, \texttt{NULL}, 0, \mathcal{G})$, so player p_1 moves first by choosing an initial action (line 4). Thereafter, at each turn $t > 0$, the active player p_i plays a best response to the opponent's last action (line 6). If this best response coincides with the action played two rounds earlier (line 7), the procedure halts and returns the current pair of actions (line 8); otherwise, it updates the recorded Nash product (lines 10–13) and invokes the other player j (line 14). The BestResponse subroutine in Alg. 2 implements a simple maximisation over Σ_i against the opponent's last action.

Algorithm 2 Compute Next Best Strategy

function BestResponse($i, \sigma^{t-1}, \mathcal{G}$)
    $\sigma^* \leftarrow$ an action in Σ_i
    for all $\sigma \in \Sigma_i$ **do**
        if $\pi_i(\sigma, \sigma^{t-1}) > \pi_i(\sigma^*, \sigma^{t-1})$ **then**
            $\sigma^* \leftarrow \sigma$
    return σ^*

Note that Algorithms 1 and 2 are oversimplified in order to serve as a basis for further development. First, although we give a single distributed algorithm in Algorithm 1, each player could choose to have its own implementation; instead of a recursive Play function we could have a function $\textsc{Play}_1$ calling in line 14 a function $\textsc{Play}_2$ and vice versa. That could allow different implementations of Algorithm 2 per $\textsc{Play}_i$, and also sharing in line 14 only the parts of the game $\mathcal{G}$ that are necessary for the other player (since our games are of incomplete information). Moreover, in order for Algorithm 1 to converge to an NBS, its implementation needs to guarantee certain regularity conditions. If each player's strategy set Σ_i is finite (or compact) and payoffs π_i are continuous, and if each best-response step never decreases the joint Nash product $(v - q_1)(v - q_2)$, then the sequence of plays makes monotonic progress and eventually stabilises at a pair of mutual best responses, that is, an NE. Moreover, if the Nash product is strictly quasi-concave on the feasible set (e.g., when at least one player's utility is strictly concave), then the NBS is unique [28, 53, 61, 63]. Without these conditions, which we enforce subsequently, the dynamics may oscillate or cycle, and convergence is not guaranteed.

For brevity, in the next section we expand on the skeleton of Algorithm 1 by working on a single recursive function that can be easily implemented as two separate and mutually recursive functions (which we do implement in Section 5).

3 Negotiation Framework for Data-Sharing

In this section, we present an abstract framework for data-sharing negotiations, building on Definition 2.5 and Algorithms 1 and 2.

In the context of data-sharing, negotiations revolve around a set of *strategy parameters* $X = \{X_1, X_2, ..., X_n\}$, such as dataset view, price, purpose of use, and duration. In our running example (ad-tech buyer and bank-marketing data provider) the strategy parameters are $X = \{X_1 : \mathit{Dataset},\ X_2 : \mathit{Query},\ X_3 : \mathit{Purpose},\ X_4 : \mathit{Third\text{-}Party},\ X_5 : \mathit{Duration},\ X_6 : \mathit{Price}\}$. At each turn, a player proposes concrete values for these parameters, (*a proposal*).

While Section 4 presents a concrete instantiation of a negotiation game tailored for data marketplaces, including a set of concrete strategy parameters, this section focuses on the underlying framework. We define a value domain V_i for each parameter X_i and for each player p_i we introduce abstract mechanisms such as: a proposal *generation* function f_i, a proposal *selection* function ϕ_i to choose from existing or generated proposals, a proposal *utility evaluation* function u_i (also called *local* utility function), and a *utility distance* function Δ to compare the utilities of two consecutive proposals (i.e., by $\Delta(u_i(\vec{\mathbf{v}}^{t-1}), u_i(\vec{\mathbf{v}}^{t}))$). We also extend the payoff functions and reservation bounds to incorporate proposal values for strategy parameters. Status quo values are default payoff values.

Definition 3.1. (Data-Sharing Negotiation Game) A data-sharing negotiation game is a tuple $\mathcal{G} = (P, \Sigma, \Pi, \Theta, Q, X, V, \Phi, F, U, \Delta)$, where:

- $(P, \Sigma, \Pi, \Theta, Q)$ is a sequential 2-player bargaining game with:
  - player p_1 being the data consumer and p_2 the data provider,
  - Σ provides available action set for players.
  - the payoff functions are extended to receive three proposal tuples (the previous two and the new one) such that $\pi_i : \Sigma \times \mathcal{V}^3 \rightarrow \mathbb{R}$, where $\mathcal{V}$ is the domain of proposals. This is a protocol-level payoff that makes the payoff value depend on the cross-product of the chosen action and the proposals, even though such action–proposal combinations are not explicitly represented in the action set.
  - Θ and Q are generalised reservation bounds of the strategy parameters and utility value, and status quo values, respectively.
- $X = (X_1, X_2, \ldots, X_n)$ is a vector of $n \in \mathbb{N}$ attributes called *strategy parameters*; these are the subjects of the negotiation.
- $V = (V_1, V_2, \ldots, V_n)$ is a vector of *domains*; each domain V_i is a set of values for the corresponding parameter X_i, such that the domain of proposals $\mathcal{V}$ is $V_1 \times V_2 \times \cdots \times V_n$.
- $\Phi = (\phi_1, \phi_2)$ is a pair of proposal selection functions where each function takes as input a strategy and two last proposals (the opponent's recent proposal and the player's last proposal); output is a proposal, i.e., $\phi_i : \Sigma \times \mathcal{V}^2 \rightarrow \mathcal{V}$.
- $F = (f_1, f_2)$ is a pair of proposal generation functions where each f_i is used internally in an ϕ_i to generate a new proposal given two previous ones: $f_i : \mathcal{V}^2 \rightarrow \mathcal{V}$.
- $U = (u_1, u_2)$ is a pair of local utility functions players apply to calculate the utility of a proposal locally.
- Δ is a proposal distance function: $\Delta : \mathbb{R}^m \times \mathbb{R}^m \rightarrow \mathbb{R}$ that will help us compare proposals or their utilities.

We model data-sharing negotiation as a 2-player sequential bargaining game, where a *data consumer* is negotiating to obtain data from a *data provider*. Both players apply the same strategy set include four available actions, which may go hand in hand with proposals:

- G: **Generate** a proposal for the strategy parameters using proposal selection function ϕ_i.
- A: **Accept** the last proposal received at $t-1$.
- R: **Reject** the proposal received at $t-1$ because it is outside the player's reservation values.
- E: **Exit** the negotiation. This is a special action that might not lead to an NBS but allows a player to arbitrarily terminate the negotiation.

The rest of the section details each dimension of Def. 3.1 and algorithms for our framework. In Section 3.1 we discuss types of strategy parameters, proposal domains and how we can evaluate them through the proposal utility functions u_i, and compare them through the proposal distance function Δ. In Section 3.2 we discuss the proposal generation and selection functions and define our setting's payoff functions. In Section 3.3 we explore reservation values and status quo.

Note that, as pointed out in Section 2, we again use a single recursive algorithm to present the behaviour of both players, with each recursive step representing the execution of a different player;

we use a single proposal selection function ϕ and a single proposal generation f. In reality, players could choose to use different functions and even play irrationally [20]. If the functions obey some general principles, such as in the case of the concrete functions we give in Section 4, and if the players do not terminate the game irrationally, games in our framework converge to (an agreed vector of values $\vec{\mathbf{v}}$ for the strategy parameters that is) an NBS.

3.1 Strategy Parameters

In a data-sharing negotiation, players negotiate over the strategy parameters. These parameters often exhibit distinct characteristics; here we distinguish four main categories: nominal, query-based, hierarchical and numerical, and discuss options for the utility evaluation function u_i and the proposal distance Δ. More types of parameters can be easily added in our framework.

Nominal parameters are attributes that represent categories without any inherent order or quantitative value, such as natural language text (e.g., a part of the negotiated agreement's terms and conditions). The utility value of a nominal parameters depends on the application and could even be human-produced: e.g., a lawyer reads proposed terms and conditions and produces a utility value. In the bank-marketing example, provider may prioritise that their brand must be mentioned in the advertisements; and the ad-tech company's lawyer might value this term highly.

A query-based parameter defines subsets of datasets or query answers. In data-sharing, actors might negotiate over query answers that return specific records or attributes (e.g., some views over the data might be private).

Evaluating the utility of a proposed query as a numerical value can be done in many different ways, for example one could price the query [47], or use information-theoretic notions such as how much information a query discloses [55]. Comparing different proposed queries in negotiation can also be done directly, without relying on utility. Notions such as query containment, equivalence [3], query determinacy [60] or query overlap [46] can also be used. For example, if a requested query violates any of the provider's prohibitions, the provider can compute a new counter-query that excludes all prohibited data by computing the overlap between the requested query and the prohibited data. The consumer can then decide to accept or reject the provider's counter-query.

Ontology-based and Hierarchical parameters are those whose possible values are organised in tree-like structures, e.g., taxonomies or ontologies, and represent relationships between concepts with varying levels of specificity. We use this type to specify dataset policies.

Policy languages such as the Open Digital Rights Language (ODRL) [19] in combination with ontologies, such as the Data Privacy Vocabulary (DPV) [64], are used to express fine-grained preferences, policies and rules. Users can define rules based on *deontic logic* concepts, namely *permissions*, *prohibitions* and *obligations* with components such as the *actions* they refer to, *assets* that are the targets of rules, *purposes* for which the rules are enforecd, *actors*, who are the assignees or assigners of the rules, etc [66]. More detailed attributes of these components can be specified in the ontology vocabularies as properties, and policies can detail restrictions on their values. In the general case, a data consumer will want the most permissive policy rules for the smallest price, whereas the data provider will want to cost data usage as much as possible.

One can combine policies and query/view-based data access. For example, the provider might define which views, tables and attributes in their dataset are subject to particular data and usage rules. A simple such ODRL permission can be for the action "execute query" with a property "query" whose value is the string representation of the query (in a standard query language). Conversely, the provider can also define prohibitions in the same manner. Assuming the requested query doesn't violate the provider's access control policy, the provider should then calculate the price of the requested dataset and query pair (using their own pricing function), and add said value to the price parameter of the negotiation.

Although orthogonal, in our framework we translate ontology requirements to numerical values. The numerical utility value of a hierarchical parameter will depend on the position of the chosen value in the hierarchy. Considering the number of edges from the root of the hierarchical tree T to the chosen value *node* as $depth(node, T)$, and $height(T)$ as the depth of the deepest node in the hierarchy, an example utility value will be:

$$utility(node) = \begin{cases} \frac{depth(node,T)}{depth(n,T)} & \text{for data providers;} \\ \frac{height(T)-depth(n,T)}{depth(n,T)} & \text{for data consumers.} \end{cases} \tag{1}$$

In the example, the ad-tech buyer sets their purpose as *Advertising*; considering the DPV purpose hierarchy[1], we have $depth(Advertising, \texttt{DPV\_purpose}) = 2$ (as a child node of *Marketing*) and we have $height(\texttt{DPV\_purpose}) = 4$. Thus, the utility value for both the data provider and the buyer is $\frac{1}{2}$. Distance functions for hierarchical values can simply compare their utility values or involve a more sophisticated algorithm that compares ontologies [29, 66].

Numerical parameters Numerical parameters are simply represented by numbers; price is a common example in bargaining problems. Utilities and distances are trivially defined, e.g, a utility value could be the normalised value of the parameter itself and the distance a simple difference. For example, $price = £50$ in the ad-tech buyer initial proposal is a numerical parameter.

Once the strategy parameters are defined, parties generate and exchange proposals based on their chosen strategies. Next, we introduce the proposal selection and generation functions and define how these are used in the payoff function of the data-sharing negotiation game.

3.2 Proposal Selection and Payoff Functions

At each turn of the game, a player may formulate a proposal $\vec{\mathbf{v}}^t$ and suggest it to the opponent. To make proposals towards an NE (and an NBS) solution, a player needs to consider their previous proposal (denoted $\vec{\mathbf{v}}^{t-2}, t \geq 2$) sent at $t-2$ and compute the new proposal $\vec{\mathbf{v}}^t$ that may have a lower local utility value than $\vec{\mathbf{v}}^{t-2}$, $u_i(\vec{\mathbf{v}}^t) \leq u_i(\vec{\mathbf{v}}^{t-2})$, while respecting the reservation bounds. Optionally, proposal generation can also take into account the opponent's recent proposal (denoted $\vec{\mathbf{v}}^{t-1}$). This is the reason that the general form of the proposal generation function is $f : \mathcal{V} \times \mathcal{V} \to \mathcal{V}$ (see Def. 3.1). In addition, efficient implementations of f should suggest proposals within the boundaries of the reservation values Θ_i which reflect the player preferences (we define reservation values customised for data-sharing scenarios in Section 3.3). Note that the feasible space of proposals in multi-issue negotiation is high-dimensional and expands with the number of strategy parameters [16], and thus, one can employ heuristic methods to generate proposals efficiently. In Section 4, we present two proposal generation heuristics that take a different number of past proposals into account.

In the beginning of the game, there are no past proposals to compare to, so the generation function computes the values for the proposal by taking the players' corresponding reservation values. For the consumer, these values will be the upper threshold of their reservation values (other than price), and the opposite for providers.

At $t \geq 2$, the proposal selection function takes the action of the player and the last two proposals $\vec{\mathbf{v}}^{t-1}, \vec{\mathbf{v}}^{t-2}$ and, if the action is G (Generate), calls f and generates a new proposal, or if the action is A (Accept), returns the accepted proposal $\vec{\mathbf{v}}^{t-1}$. When the player's action is R (Reject), because the opponent's proposal is outside the player's reservation values, they either generate a new proposal, as G (if possible) or return their own previous proposal $\vec{\mathbf{v}}^{t-2}$. Ultimately, if the player is choosing E (Exit) there is no need to call ϕ or f. For ease of notation, we will define the set $\vec{\mathbf{V}}^t = \{\vec{\mathbf{v}}^i \mid i \in [0, t]\}$ that contains needed previous proposals.

[1]https://github.com/w3c/dpv/blob/master/diagrams/purpose-2.png

$$\phi(\sigma^t, \vec{\mathbf{V}}^{t-1}) = \begin{cases} f(\vec{\mathbf{v}}^{t-1}, \vec{\mathbf{v}}^{t-2}) & \text{if } \sigma^t = G, \\ \vec{\mathbf{v}}^{t-1} & \text{if } \sigma^t = A; \\ \vec{\mathbf{v}}^{t-2} & \text{if } \sigma^t = R; \\ \texttt{NULL} & \text{if } \sigma^t = E; \end{cases} \tag{2}$$

In order to select a strategy at t, the player must evaluate the payoff of the strategy profile at t, $\bar{\sigma}^t = (\sigma^t, \sigma^{t-1})$, but taking the generated proposal $\vec{\mathbf{v}}^t$ and the previous two proposals, $\vec{\mathbf{v}}^{t-1}, \vec{\mathbf{v}}^{t-2}$, into consideration. The extended payoff function also considers the proposal utility and distance functions in order to compute the action with the highest payoff, and is defined as follows:

$$\pi_i(\sigma^t, \sigma^{t-1}, \vec{\mathbf{V}}^t) = \begin{cases} 1 & \text{if } (\sigma^t, \sigma^{t-1}) = (A, A) \\ 1 & \text{if } (\sigma^t, \sigma^{t-1}) = (E, E) \\ 1 - \Delta(u_i(\vec{\mathbf{v}}^{t-1}), u_i(\vec{\mathbf{v}}^t)) & \text{if } (\sigma^t, \sigma^{t-1}) = (G, G) \\ 1 - \Delta(u_i(\vec{\mathbf{v}}^{t-2}), u_i(\vec{\mathbf{v}}^{t-1})) & \text{if } (\sigma^t, \sigma^{t-1}) = (A, G/R) \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

When the opponent plays an Accept or Exit action, Equation 3 incentivises the player to mirror these actions (the first two cases). If the opponent either generates a proposal (G) or rejects the player's proposal (R), then the payoff corresponds to the best outcome between accepting the opponent's proposal or generating a new one in response (the next two cases). In all other cases, the payoff is 0, as they represent invalid combinations of strategies.

Considering the proposal (or local) utility function $u_i(\vec{\mathbf{v}})$, several functions are applicable. In multi-objective decision and game-theoretic settings, an *additive* utility function is appropriate when the objectives are largely *substitutable*[2] and trade-offs between them are approximately linear [8, 42, 43]. While our framework adopts a linear additive utility function within each agent for tractability and to ensure convexity of acceptable proposal sets, other formulations—such as multiplicative [41], quadratic [54], or non-additive (Choquet) utilities [8, 34]—can capture richer inter-issue interactions. For example, a *multiplicative* utility function is preferred when objectives are complementary or interact in a "weakest-link" manner, when proportional effects or scale invariance are desired [6, 41, 61, 70]. In addition, for the Δ functions we can use different options, e.g., Euclidean distance or Manhattan (L1) distance. In Section 4 we employ a simple Euclidean distance Δ function and a linear additive utility function.

Before integrating the proposal selection and payoff functions into our algorithm in Section 3.4, we discuss reservation values.

3.3 Reservation Values and Status Quo

In our framework we abuse the notion of a reservation bound, which is simply a lowest allowed payoff value per Definition 2.5, and define bounds for the values of the strategy parameters themselves (i.e., defining whether proposal $\vec{\mathbf{v}}$ is within the reservation bounds), as well as a lowest acceptable value for the proposal utility. In effect, a bound will be a pair $\Theta_i = (\{\theta^i_j\}^n_{j=1}, \theta^i_{ur})$; where $\{\theta^i_j\}^n_{j=1}$ is the per-issue feasibility region, such that each $\theta^i_j = <\theta^{min}_j, \theta^{max}_j>$ is the acceptable value range for strategy parameter X_j for player p_i and θ^i_{ur} is the utility reservation threshold for p_i, comparable with $u_i(\vec{\mathbf{v}})$. In fact, θ^i_{ur} is the lowest acceptable payoff value for player i. Depending on the type of the strategy parameter, θ^i_j is in one of the following categories:

[2]If one objective becomes worse, another can improve sufficiently to keep overall satisfaction constant.

- If X_j is a nominal parameter, reservation bounds can come from user defined-functions that analyse the value of the parameter in a proposal or from extra-algorithmic processes.
- If X_j is a query-based parameter, reservation bounds can come from subset relations of the query answers, containment relationships of queries, or numerical prices that represent the answer's information load, or monetary value. For example, the bank-marketing provider may simply set their reservation bounds for $X_j : Query$ to (SELECT * FROM D) demonstrating their willingness to share the whole table.
- If X_j is a hierarchical parameter, $\theta_j^{\min}$ can be a value at the lower levels of the hierarchy, and $\theta_j^{\max}$ at the higher levels. For example, the bank-marketing provider sets the bounds of $X_j : Purpose$ as $\theta_j^{\max} = \mathit{PersonalisedAdvertising}$, meaning that they will accept purposes under it in the respective hierarchy, such as *TargetedAdvertising* and they will reject *Advertising* which is higher than $\theta_j^{\max}$ in the hierarchy.
- If X_j is a numerical parameter, the bounds are self-explanatory. For example, an ad-tech buyer may define their reservation bounds for $X_j : Price$ as $[0, 90]$.

Note that there must exist an overlap between the players' reservation values, otherwise negotiation is not possible.

Regarding status quo, we assume there is no prior agreement and a disagreement yields no local utility for either player and $Q = (q_1, q_2) = (0, 0)$. While this assumption may incentivise players to reach an agreement, our framework is flexible and can incorporate payoff values from previous agreements as the players' status quo values.

In our example, we instantiate Definition 3.1 with the ad-tech buyer (consumer p_1) and the bank-marketing provider (p_2). The strategy parameters are $X = \{X_1 : Query,\ X_2 : Purpose,\ X_3 :$ Third-Party, $X_4 : Duration,\ X_5 : Price\}$ with domains $V_1, \ldots, V_5$ as defined earlier. A proposal in round t is of the form $\vec{\mathbf{v}}^t = (v_1^t, \ldots, v_5^t) \in V_1 \times \cdots \times V_5$. The utilities $U = (u_1, u_2)$ are linear additive over normalised per-attribute scores (query value, purpose depth, etc.), with a scalar distance $\Delta(a, b) = |a - b|$ to measure movement across rounds. Each side enforces reservation bounds Θ_i (per-attribute ranges plus a minimum utility). At $t = 0$ the buyer starts the negotiation with $\vec{\mathbf{v}}^0$ =((SELECT * FROM D WHERE 30 $\leq$ *age* $\leq$ 45 AND *job* $\in \{technician, admin\}$), *Advertising*, *Company*, 90d, £50). If the provider does not accept $\vec{\mathbf{v}}^0$ and chooses G, they generate a counter-offer; e.g., $\vec{\mathbf{v}}^1 = (v_1^0, \mathit{PersonalisedAdvertising}, v_3^0,$ 60d, £70), staying within Θ_2. The buyer then chooses A or G by maximising π_1; under the proposal generation step, f_1 nudges $\vec{\mathbf{v}}^0$ toward $\vec{\mathbf{v}}^1$ (e.g., shorter duration or higher price) while keeping $u_1 \geq \theta_{ur}^1$. This back-and-forth continues until both accept (A, A) or one exits.

3.4 Data-Sharing Negotiation Algorithm

We extend Algorithm 1 into Algorithm 3 which now contains the set of past proposals $\vec{\mathbf{V}}^{t-1}$ as an additional input, and one additional output, $\vec{\mathbf{v}}^t$, which is the converged agreed proposal of the NBS of the game. We also extend Algorithm 2 with the same two additional inputs and output resulting in Algorithm 4. Similarly to Algorithm 1, the first call of the game is Play(1, 0, NULL, NULL, 0, $\emptyset$, $\mathcal{G}$); these are the same values that initialise Algorithm 1 and where the new inputs $\vec{\mathbf{v}}^{t-1}$ and $\vec{\mathbf{v}}^{t-2}$ are both undefined. Unlike Algorithm 1, and closer to a real setting, any player can choose at any turn to terminate the game by choosing strategy E (lines 3-4). In the next turn the opponent will reply to this with an E themselves and then terminate the game by returning NULL values (lines 9-10).

At the first turn of the game ($t = 0$), and if not exiting, p_1 now must choose the action G, and generate a proposal for the strategy parameters (line 6); the first proposal will be generated to fully satisfy the user's preferences by assigning the upper bounds of the reservation values to each parameter. As before, in intermediate game states, the player calls the BestResponse function at

Algorithm 3 Distributed Bargaining for Data-Sharing Negotiation

1: **function** PLAY($i, t, \sigma^{t-1}, \sigma^{t-2}, \texttt{maxProd}, \vec{\mathbf{V}}^{t-1}, \mathcal{G}$)
2: $j \leftarrow (2 - i) + 1$ ▷ j is the other player
3: **if** Player chooses to terminate **then**
4: **return** PLAY($j, t+1, E, \sigma^{t-1}, 0, \emptyset, \mathcal{G}$)
5: **if** $t < 2$ **then** ▷ First play of either player.
6: $\vec{\mathbf{v}}^t \leftarrow \phi(G, \emptyset)$
7: **else**
8: $(\sigma^t, \vec{\mathbf{v}}^t) \leftarrow$ BESTRESPONSE($i, \sigma^{t-1}, \vec{\mathbf{V}}^{t-1}, \mathcal{G}$)
9: **if** $(\sigma^t, \sigma^{t-1}) == (E, E)$ **then**
10: **return** $((E, E), 0, \texttt{NULL})$
11: **if** $\vec{\mathbf{v}}^t == \vec{\mathbf{v}}^{t-1}$ **then**
12: **return** $((A, A), \texttt{maxProd}, \vec{\mathbf{v}}^t)$
13: **else**
14: $\upsilon \leftarrow \pi_1(\sigma^t, \sigma^{t-1}, \vec{\mathbf{V}}^t)$
15: $\nu \leftarrow \pi_2(\sigma^t, \sigma^{t-1}, \vec{\mathbf{V}}^t)$
16: **if** $(\upsilon - q_1) \cdot (\nu - q_2) > \texttt{maxProd}$ **then**
17: $\texttt{maxProd} \leftarrow (\upsilon - q_1) \cdot (\nu - q_2)$
18: $\vec{\mathbf{V}}^t \leftarrow \vec{\mathbf{V}}^{t-1} \cup \{\vec{\mathbf{v}}^t\}$
19: **return** PLAY($j, t+1, \sigma^t, \sigma^{t-1}, \texttt{maxProd}, \vec{\mathbf{v}}^t, \mathcal{G}$)

Algorithm 4 Best response for Data-Sharing Negotiation

1: **function** BESTRESPONSE($i, \sigma^{t-1}, \vec{\mathbf{V}}^{t-1}, \mathcal{G}$)
2: **if** $\sigma^{t-1} == R$ **then**
3: **if** $\vec{\mathbf{v}}^{t-1} \in \Theta_i$ **then**
4: **return** $(A, \vec{\mathbf{v}}^{t-1})$
5: **else**
6: **return** $(E, \texttt{NULL})$
7: **if** $\vec{\mathbf{v}}^{t-1} \notin \Theta_i$ and $\vec{\mathbf{v}}^{t-1} == \vec{\mathbf{v}}^{t-3}$ **then**
8: **return** $(R, \vec{\mathbf{v}}^{t-2})$
9: $\sigma^* \leftarrow$ an action in $\Sigma_i = \{A, E, G\}$
10: $\vec{\mathbf{v}}^{*t} \leftarrow \phi(\sigma^*, \vec{\mathbf{V}}^{t-1})$
11: **for all** $\sigma \in \{A, E, G\}$ **do**
12: $\vec{\mathbf{V}}^{*t} \leftarrow \vec{\mathbf{V}}^{t-1} \cup \{\vec{\mathbf{v}}^{*t}\}$
13: $\vec{\mathbf{v}}^t \leftarrow \phi(\sigma, \vec{\mathbf{V}}^{t-1})$
14: $\vec{\mathbf{V}}^t \leftarrow \vec{\mathbf{V}}^{t-1} \cup \{\vec{\mathbf{v}}^t\}$
15: **if** $\pi_i(\sigma, \sigma^{t-1}, \vec{\mathbf{V}}^t) > \pi_i(\sigma^*, \sigma^{t-1}, \vec{\mathbf{V}}^{*t})$ **then**
16: $\sigma^* \leftarrow \sigma$
17: $\vec{\mathbf{v}}^{*t} \leftarrow \vec{\mathbf{v}}^t$
18: **return** $(\sigma^*, \vec{\mathbf{v}}^{*t})$

line 8 (now implemented by Algorithm 4). This function will return the same proposal that got in its input whenever there is an agreement and the two players play A, at which point the game will terminate with an NBS (lines 11-12). In the remainder of the cases, the algorithm proceeds as previously, maintaining the maximum Nash product (lines 14-17) and calling the next turn (line 19).

Notice that the first action of Algorithm 4 is to check if the opponent rejected the proposal (line 2). If the opponent's proposal is within the player's reservation values, they accept it (lines 3-4). Otherwise, the player chooses E and terminates the game (Line 6). On the other hand, if the opponent's proposal is outside the player's reservation values, and if the opponent's previous two proposals at $t-1$ and $t-3$ are identical (line 7) the player will fall back to suggesting the previous proposal they had sent at $t-2$ (line 8). This is because, despite a prior rejection of $\vec{\mathbf{v}}^{t-3}$ by the player (by generating a new one), the opponent has sent the same proposal.

In any other case, the algorithm proceeds by choosing the best strategy and proposal (towards the NBS) in lines 11-17.

One-round walk-through on the running example. Assume the buyer's initial proposal is $\vec{\mathbf{v}}^0$, and the provider replies with $\vec{\mathbf{v}}^1$. On the buyer's next turn:

$$\begin{aligned}
\textbf{A:}\ & \sigma_1^t = A \ \Rightarrow\ \vec{\mathbf{v}}^t = \vec{\mathbf{v}}^{t-1} = \vec{\mathbf{v}}^1 \ \ \text{(agreement: } (A, A)).\\
\textbf{G:}\ & \sigma_1^t = G \ \Rightarrow\ \vec{\mathbf{v}}^t = f_1(\vec{\mathbf{v}}^{t-1}, \vec{\mathbf{v}}^{t-2}) = f_1(\vec{\mathbf{v}}^1, \vec{\mathbf{v}}^0) \ \ \text{(e.g., } £50 \rightarrow £60).\\
\textbf{R/E:}\ & \sigma_1^t = R \Rightarrow\ \vec{\mathbf{v}}^t = \vec{\mathbf{v}}^{t-2} = \vec{\mathbf{v}}^0; \quad \sigma_1^t{=}E \Rightarrow \vec{\mathbf{v}}^t = \texttt{NULL} \ \ \text{(terminate)}.
\end{aligned}$$

4 Negotiation in Data Marketplaces

In this section we instantiate our framework for the case of negotiating on Data Marketplaces (DMs) [68]. Data providers who are willing to share their datasets submit relevant dataset specifications to DMs. Data consumers explore data marketplaces to find datasets of interest. Upon discovering a relevant dataset, they may initiate negotiations with data providers or intermediaries to gain access. To trigger the negotiation process, a data consumer needs to specify their desired dataset as a query-based strategy parameter X_i, their reservation values Θ, and a preferred proposal utility function u; since we implement a linear additive utility function (Equation 4), the players need only to specify the weights of the function. In addition to these inputs our system provides a proposal generation and selection function, and a payoff function as defined in Section 3.2. We also consider the default status quo values as 0. Based on these inputs and our system's offerings, consumers' agents can initiate and conduct negotiations with data providers' agents.

4.1 Framework Instantiation for DMs

Our system uses the ODRL language [19] to support policies for data usage. These policies support a number of features that we make available as **strategy parameters** X:

- **Dataset:** The primary subject of the negotiation, represented by a URL.
- **Query**: A dataset can be shared in its entirety or through a query-defined subset. We use conjunctive queries for this (these are Select-Project-Equijoin queries in SQL [3]).
- **Data Usage Rules:** ODRL can specify rules on how the shared data can be used. These have the following elements that agents can negotiate:
  - **Type:** The rule specifies a *permission*, a *prohibition*, or an *obligation.*
  - **Action:** Specific operation or data processing action permitted, prohibited or obliged. This is a hierarchy parameter chosen from either the ODRL vocabulary of actions or the DPV processing operations [72].
  - **Actor:** The role or entity that the action refers to (e.g., Company). This is a hierarchy parameter chosen from the DPV vocabulary of actors.
  - **Third-Party Sharing:** The actors or entities with whom the data consumer may further share the dataset. Hierarchical parameter as above.

  - **Duration:** Period for which the data will be shared. Numerical parameter.
  - **Purpose:** The intended purpose for data usage; hierarchical parameter chosen from the DPV purposes.
- **Dataset Price:** The monetary value for data access.
- **Environmental Impact:** The environmental consequences of data-sharing and subsequent use. Numerical value denoting the carbon footprint of the dataset or the operation.
- **Terms and Conditions:** Nominal parameter containing text around the terms and conditions of data usage.

After the data consumer's agent generates the initial proposal it starts the game $\mathcal{G}$, the data provider—if willing to participate—invokes its corresponding agent to enter the game. Upon receiving a proposal, each agent compares it against its **reservation values** to identify any mismatches. if both data consumers and providers express their reservation values using the ODRL format, mismatch detection reduces to a rule comparison [69] which our system implements.

Then, agents compute the local utility of the proposals. We offer a linear additive utility function, which is a common assumption in games with multiple objectives [42]. We call it the **proposal utility function** $u_i(\vec{\mathbf{v}})$. The value $norm(v_j)$ denotes the normalised form of a v_j in proposal of values $\vec{\mathbf{v}}$, where $norm(\cdot)$ is a domain-specific normalisation function. To reflect the relative importance—or priority—of each parameter, agents define a weight vector $W_i = (w_1, \ldots, w_n)$, where w_j is the weight assigned to parameter X_j, such that $\sum_{j=1}^{n} w_j = 1$. Agents compute the proposal utility value using Equation 4.

$$u_i(\vec{\mathbf{v}}) = \sum_{j=1}^{n} w_j \cdot norm(v_j) \tag{4}$$

The agents compute the payoff value (Equation 3) using the utility of the proposals and a simple Euclidean proposal distance Δ function. The agents, then choose their next action based on the payoff value. In a case that an agent selects G to generate a new proposal, we introduce the **proposal generation function** f for DMs, consists of the following steps:

(1) **Target Parameters Value Adjustment:** For each strategy parameter X_j, a player computes a desired value v_j^{target}. While this value should be less desirable than the one in the agent's previous proposal, it moves the agent closer to an agreement and remains sufficiently acceptable. The agent uses their reservation values $(\theta_j^{min}, \theta_j^{max})$, the previous proposal at $t-2$ (assuming that the player received the opponent's proposal at $t-1$) and a time-dependent concession function[3] $c(t)$ with a concession rate β that defines the speed of concession:

$$v_j^{target} = \begin{cases} \max(\theta_j^{\min}, v_j^{t-2} - c(t)) & \text{if } p_i \text{ is a data provider,} \\ \min(\theta_j^{\max}, v_j^{t-2} + c(t)) & \text{if } p_i \text{ is a data consumer,} \end{cases}$$

where: $c(t) = (\theta_j^{\max} - \theta_j^{\min}) \cdot w_j \cdot \left(\frac{t}{t_{\max}}\right)^{\beta}$ We invert price into a benefit-aligned variable [42] so that it is consistent with the direction of preference of the other strategy parameters. Computing the desired values of all strategy parameters, the agent calculates the joint utility value using Equation 4 as u_i^{target}, where u_i^{target} is the agent's desired utility value of their next proposal.

(2) **Indifference Surface Computation:** The agent then identifies a set $\Psi_i^t = \{\vec{\mathbf{v}} \in V_1 \times \cdots \times V_n \mid u_i(\vec{\mathbf{v}}) = u_i^{\text{target}} \text{ and } \vec{\mathbf{v}} \text{ satisfies } \Theta_i\}$ as proposals that yield the desired utility value $u_i^{\text{target}} \geq \theta_{ur}^i$

[3] A concession strategy is a negotiation tactic where an agent gradually relaxes its demands or preferences over time, in hopes of reaching an agreement. It reflects the idea that the longer the negotiation continues, the more willing the agent is to compromise.

and satisfy per-issue reservation boundaries, $\{\theta_j^i\}_{j=1}^n$. Ψ_i^t is the indifference surface of the desired utility value.

(3) **Proposal Selection Heuristic:** Then, the agent must select one proposal from Ψ_i^t to send to the opponent. We suggest the following heuristics:
- **Greedy:** In a greedy approach, the agent selects a proposal $\vec{\mathbf{v}}$ with the minimum Euclidean distance from their own previous proposal $\vec{\mathbf{v}}^{t-2}$ as: $\vec{\mathbf{v}}^t = \arg\min_{\vec{\mathbf{v}}\in\Psi_i^t} \|\vec{\mathbf{v}} - \vec{\mathbf{v}}^{t-2}\|_2$.
- **Concession:** In the concession approach, the player selects a proposal with the minimum Euclidean distance from the opponent's proposal $\vec{\mathbf{v}}^{t-1}$ as: $\vec{\mathbf{v}}^t = \arg\min_{\vec{\mathbf{v}}\in\Psi_i^t} \|\vec{\mathbf{v}} - \vec{\mathbf{v}}^{t-1}\|_2$.

Function f thus generates a proposal that brings the agents closer to an agreement and aligns with their desired proposal utility while incorporating either self-oriented (greedy) or opponent-oriented (concession) preferences. The selection heuristic may be replaced by alternative similarity functions fuzzy similarity, as proposed in [65], depending on the negotiation context.

4.2 Convergence and Complexity

We analyse the convergence and computational efficiency of our bilateral negotiation process. Each agent's utility $u_i(\vec{\mathbf{v}})$ is continuous and concave, with feasible regions that are closed, convex, and overlapping. At each round, an agent updates its proposal by moving towards the opponent's last offer within its own acceptable set. This *non-expansive* [4] update rule ensures that inter-proposal distances shrink monotonically, making the alternating sequence of offers Cauchy [5] and hence convergent. Under compactness and continuity assumptions, the sequence converges to a stable agreement $\vec{\mathbf{v}}^\star$ that is individually rational and Pareto-efficient. Moreover, because the Nash product $NP(\vec{\mathbf{V}}^t) = (v - q_1)\,(\nu - q_2)$; where $v \leftarrow \pi_1(\sigma^t, \sigma^{t-1}, \vec{\mathbf{V}}^t), \nu \leftarrow \pi_2(\sigma^t, \sigma^{t-1}, \vec{\mathbf{V}}^t)$ is non-decreasing and bounded, the process converges to its maximum value. The limit $\vec{\mathbf{v}}^\star$ thus maximises the joint Nash product, coinciding with the *Nash Bargaining Solution (NBS)* [62]. When utilities are strictly concave (or the Nash product is strictly quasi-concave), the limit is unique [28, 53, 61, 63].

Computational complexity. Each negotiation round involves evaluating utilities, feasibility constraints, and generating the next proposal. With linear-additive utilities and closed-form updates, the per-round cost is near-linear in the number of attributes, $O(n)$, or up to $\tilde{O}(n^3)$ when using a generic quadratic-program (QP) solver [6]. The number of rounds required to reach an ε-accurate outcome depends on concession dynamics: logarithmic $O(\log(1/\varepsilon))$ under smooth, strongly concave utilities, and sublinear $O(1/\varepsilon)$ under standard diminishing-concession policies. Hence, the overall runtime is:

$$T(\varepsilon) = \begin{cases} \tilde{O}\big(n \log \frac{1}{\varepsilon}\big) & \text{(smooth case)}, \\ \tilde{O}\big(n\,\varepsilon^{-1}\big) & \text{(generic diminishing concessions)}. \end{cases}$$

Overall, the protocol achieves provably convergent, fair, and efficient outcomes with practical computational complexity for real-world multi-issue data negotiations.

5 Performance Evaluation

To evaluate the performance of our negotiation framework, we conduct experiments across multiple datasets and utility functions, and test robustness to weight perturbations and strategic misreporting.

[4] A map is non-expansive if it never makes two points farther apart; projection onto a convex set is non-expansive [7].

[5] A sequence is Cauchy if, as you go further and further along the sequence, all of its elements get arbitrarily close to each other.

[6] A generic QP solver is a general-purpose algorithm/library that solves quadratic programs—optimisation problems with a quadratic objective and linear constraints [12]. We use it when the step is more complex than one equality + bounds and we cannot exploit special structure. For example, to encode policy logic such as "if *purpose* = *PersonalisedAdvertising* then *third-party* = *Company* and *duration* ≤ 90," we introduce binary indicators and linear implications.

We report agreement rate, joint utilities, Nash product, Jain fairness, and time-to-agreement, with sensitivities to reservation overlap, conflict severity, and parameter count. Baselines include TILI [50], MCP [22], and an oracle centralised Nash Bargaining Solution [18, 44] (where computable). We also present a PoC study.

5.1 Experimental Setup and Parameter Settings

We implemented and executed all the algorithms in Python on a MacBook, which has a single CPU of Apple M1 and 8 GB of memory. Agents negotiate over sharing data.

They are rational and follow the alternating multi-issue proposal protocol. Strategy parameters for our use case are: dataset, query, data usage rules (such as type, action, actor third-party sharing, duration, and purpose), and dataset price (see Section 4.1).

We use three datasets to span sizes and distributions. **Bank Marketing** [59]: a medium-size tabular data associated with direct marketing campaigns (phone calls) of a Portuguese banking institution [7]. **RecGym** [11]: a larger, event-style records which is a collection of gym workouts designed for research and development in recommendation systems and fitness applications [8]. **PIRVision** [21]: an additional tabular dataset with different attribute distributions contains occupancy detection data collected from a sensing node in residential and office environments [9]. For each dataset we define negotiable attributes with the same DPV/ODRL vocabularies.

The agents follow the same set of strategies and proposal function introduced in Section 3 to participate in the negotiation. The agents can choose either a greedy approach or a concession approach (see Section 4.1), denoted GTG and GTC respectively, to generate a new proposal. All experiments use the closed-form projection onto the agent's indifference surface under one equality (target utility) plus box constraints (reservation bounds). This yields a near-linear per-round cost in the number of negotiated attributes n (i.e., $\tilde{O}(n)$). We do *not* cite a generic QP solver in any reported result. The common goal is to reach an agreement while each agent tries to improve its payoff value. To measure the payoff value (Equation 3) agents need to calculate the proposal utility of proposals; they use a *linear additive function* (Equation 4) as a baseline and a *multiplicative function*, $u_i(\vec{\mathbf{v}}) = \prod_j norm(v_j)^{\alpha_{ij}}$ (with attribute-wise normalisation $norm(\cdot)$ and $\sum_j \alpha_{ij} = 1$).

Each agent has a set of reservation values (Section 3.3) while they have no or incomplete information about their opponent's reservation values as well as the weights w_j in proposal utility function. The negotiation process terminates when either an agreement is reached, agents exit the game or the number of rounds exceeds a set threshold, to avoid be stuck in loops.

5.2 Metrics and Sensitivity Factors

We report the following metrics:

Agreement rate. Let $\vec{\mathbf{V}}_{\text{agree}}$ be the set of proposals that were agreed on by both players at the end of a run, and let N be the total number of runs (100 in our experiments). We use Equation 5 to calculate the agreement rate of each experiment:

$$\text{Agreement Rate} = \frac{|\vec{\mathbf{V}}_{\text{agree}}|}{N} \times 100 \tag{5}$$

[7]https://archive.ics.uci.edu/dataset/222/bank+marketing
[8]https://archive.ics.uci.edu/dataset/1128/recgym:+gym+workouts+recognition+dataset+with+imu+and+capacitive+sensor-7
[9]https://archive.ics.uci.edu/dataset/1101/pirvision_fog_presence_detection

Utility. We compute average utility value over agreements for each player using Equation 6, where u_i is the local utility function of the player i.

$$\text{Average Utility} = \frac{\sum_{\vec{\mathbf{v}} \in \vec{\mathbf{V}}_{\text{agree}}} u(\vec{\mathbf{v}})}{|\vec{\mathbf{V}}_{\text{agree}}|} \tag{6}$$

Fairness. We measured fairness using *Jain's Fairness Index* [38], which captures the degree of balance in utility distribution. A perfectly fair outcome yields a value of 1, indicating that both parties received equal utility. Let $\vec{\mathbf{v}} \in \vec{\mathbf{V}}_{\text{agree}}$ represent the final proposal the players agreed upon. Then, Jain's Index is calculated as Equation 7:

$$\text{Fairness} = \frac{\left(u_1(\vec{\mathbf{v}}) + u_2(\vec{\mathbf{v}})\right)^2}{2\left(u_1(\vec{\mathbf{v}})^2 + u_2(\vec{\mathbf{v}})^2\right)} \tag{7}$$

Time-to-agreement. Average of the number of rounds needed to reach an agreement;
Oracle gap. We also report the difference to the centralised NBS in utilities [10].

We assess these metrics from different perspectives and sensitivity factors as below:

Overlap between reservation values. Represented by the percentage of overlap between preferences (from 10% to 90%). To compute the degree of overlaps between reservation values per-issue $\{\theta^i_j\}^n_{j=1}$ (see reservation values definition in Section 3.3), we use Jaccard similarity coefficient (multiplied by 100 for percentage);

$$overlap(\{\theta^1_j\}, \{\theta^2_j\}) = \frac{|\{\theta^1_j\} \cap \{\theta^2_j\}|}{|\{\theta_j\}^1 \cup \{\theta^2_j\}|} \times 100 \tag{8}$$

Conflict rate. We define conflict rate as the degree of disagreement between two proposals, quantified by the normalised distance between their values across all strategy parameters. Let $\vec{\mathbf{v}}_1 = (v_{1,1}, v_{1,2}, \ldots, v_{1,n})$ and $\vec{\mathbf{v}}_2 = (v_{2,1}, v_{2,2}, \ldots, v_{2,n})$ represent two proposals as vectors of strategy parameter values. We normalise each value $v_{i,j}$ to the interval $[0, 1]$ using the domain-specific normalisation function $norm(\cdot)$ such that $\tilde{v}_{i,j} = norm(v_{i,j})$, for $i \in \{1, 2\},\ j \in \{1, \ldots, n\}$. The normalised proposals are thus: $\tilde{\vec{\mathbf{v}}}_1 = (\tilde{v}_{1,1}, \tilde{v}_{1,2}, \ldots, \tilde{v}_{1,n})$, and $\tilde{\vec{\mathbf{v}}}_2 = (\tilde{v}_{2,1}, \tilde{v}_{2,2}, \ldots, \tilde{v}_{2,n})$. We define the conflict rate as the normalised Euclidean distance between the two proposals:

$$ConflictRate(\vec{\mathbf{v}}_1, \vec{\mathbf{v}}_2) = \frac{\left\|\tilde{\vec{\mathbf{v}}}_1 - \tilde{\vec{\mathbf{v}}}_2\right\|_2}{\sqrt{n}} = \frac{\sqrt{\sum_{j=1}^{n} (\tilde{v}_{1,j} - \tilde{v}_{2,j})^2}}{\sqrt{n}} \tag{9}$$

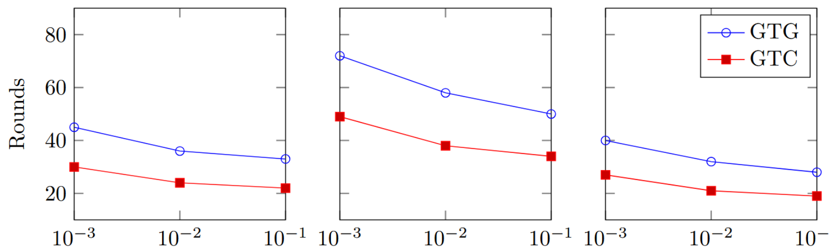
Since each strategy parameter X_j has an associated weight w_j, with $\sum_{j=1}^{n} w_j = 1$ in the proposal utility functions (see Equation 4), we use the **weighted conflict rate** too as:

$$WeightedConflictRate(\vec{\mathbf{v}}_1, \vec{\mathbf{v}}_2) = \sqrt{\sum_{j=1}^{n} w_j \cdot (\tilde{v}_{1,j} - \tilde{v}_{2,j})^2} \tag{10}$$

Weighted conflict rates range from 0 to 1, where 1 represents high conflict/misalignment between values.

Scalability. The number of negotiation strategy parameters/attributes n, ranging from 1 to 10.
Time. The number of iterations, ranging from 1 to 50.

[10]The centralised NBS is the outcome when a single planner/solver with full information computes the agreement directly—i.e., it solves the Nash-product maximisation once, rather than the agents reaching it through a distributed/alternating negotiation.



(a) Bank **(b) RecGym** **(c) PIRVision**

Fig. 1. Rounds to agreement vs. tolerance ε.

5.3 Findings

We conduct the experiments and present results in three categories: (A) convergence and computational efficiency; (B) agreement, utility, and fairness under key sensitivities; (C) ablations and (D) Oracle comparisons.

5.3.1 Convergence and Efficiency Analysis. We evaluate the two proposal generation heuristics *Greedy* (GTG) and *Concession* (GTC) on the datasets. Each scenario instantiates the same bilateral protocol with ODRL/DPV policy attributes and a query/view attribute.

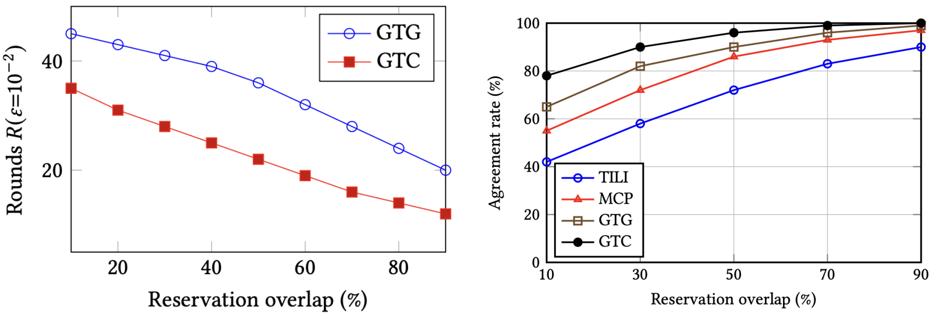
We declare ε-approximate convergence when either the Nash product change is small, $|\mathrm{NP}(\vec{\mathbf{V}}^t) - \mathrm{NP}(\vec{\mathbf{V}}^{t-1})| \leq \varepsilon$, or the proposal distance is small, $\|\vec{\mathbf{v}}^t - \vec{\mathbf{v}}^{t-1}\|_2 \leq \varepsilon$, with $\varepsilon \in \{10^{-1}, 10^{-2}, 10^{-3}\}$. We report rounds-to-agreement $R(\varepsilon)$, per-round runtime (proposal generation plus utility evaluation), and total time (wall-clock). Results are averaged over 30 runs per setting with 95% confidence intervals (Figure 1).

We then vary overlap (Θ_1, Θ_2) from 10% to 90% while holding other factors fixed. Both heuristics show monotone improvement in $R(\varepsilon)$ with more overlap; GTC benefits more at low–moderate overlap (10–50%), where targeted concessions accelerate alignment. At very high overlap ($\geq$ 80%), both terminate quickly and the gap narrows (Figure 2a).

Next, we study the impact of utility functions and informative views on the performance. We compare additive and multiplicative utilities on agreement rate and Nash product (as utility product) as reservation overlap increases. The curves are nearly identical on agreement rate and Nash product, and the multiplicative form yields a slight efficiency edge at higher overlap (e.g., median NP $\uparrow$ by $\approx$ 0.02 at 90% overlap), consistent with its complementary-attributes behavior (Figure 3a). We also add a small bonus for "more informative" queries, weighted by λ. Raising λ from 0 to 0.10 improves agreement and boosts Nash product ($6 - 9\%$). However, we see a mild tapering at $\lambda = 0.15$. So the information bonus is helpful, but not the main driver of outcomes (Figure 3b). Overall, GTC is faster in rounds than GTG across datasets and tolerances, and with a closed-form proposal generation step, per-round complexity is predictable—near-linear in the number of attributes.

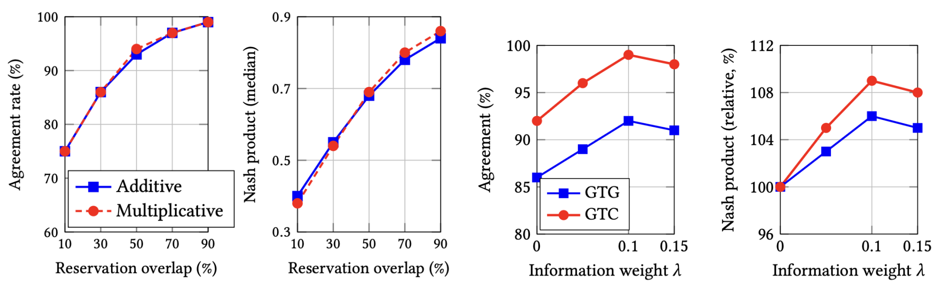
5.3.2 Agreement, utility, and fairness under sensitivities. We benchmark our framework (GTG/GTC) against *TILI* and *MCP* across sensitivity factors. We average results over 50 runs per setting, and report agreement rate, the Nash product, and Jain's fairness index.

Agreement rate. GTC consistently achieves the highest agreement rates across all factors. GTG is competitive when preferences are moderately aligned, but degrades earlier as conflict rises. MCP performs well with few attributes and high overlap, yet drops as dimensionality or conflict increases. TILI is strong only in favourable cases and otherwise lags due to its one-shot nature. Figure 2b



(a) (b)

Fig. 2. Metrics against reservation values overlap (a) rounds to agreement, (b) agreement rate.



(a) Utility family comparison (additive vs. multiplicative).

(b) Effect of the information-oriented query component.

Fig. 3. Agreement rates and Nash products.

shows agreement rises with overlap; GTC leads across the range (from 78% $\rightarrow$ 100%), GTG is next (65% $\rightarrow$ 99%), MCP follows (55% $\rightarrow$ 97%), and TILI trails (42% $\rightarrow$ 90%).

Utilities and Nash product. GTC yields the most balanced utilities and the largest Nash (utility) product across datasets. As hierarchy depth and query selectivity increase, utilities shift (reflecting policy alignment and information value), yet GTC maintains superior joint gains. GTG trails GTC but remains robust under moderate conflict. MCP's utilities decline with n and depth, while TILI systematically deviates toward the proposer. Fig. 4 shows that as conflict increases, utilities and NP fall at different rates; GTC keeps the strongest levels (u_1 : 0.78 $\rightarrow$ 0.42, u_2 : 0.80 $\rightarrow$ 0.52, NP : 0.62 $\rightarrow$ 0.22), GTG is second (u_1 : 0.70 $\rightarrow$ 0.28, u_2 : 0.74 $\rightarrow$ 0.40, NP : 0.52 $\rightarrow$ 0.11), MCP degrades more (u_1 : 0.72 $\rightarrow$ 0.30, u_2 : 0.76 $\rightarrow$ 0.48, NP : 0.55 $\rightarrow$ 0.14), and TILI loses NP fastest (u_1 : 0.82 $\rightarrow$ 0.05, u_2 : 0.88 $\rightarrow$ 0.75, NP : 0.72 $\rightarrow$ 0.04).

Fairness. Measured by Jain's index, GTC is the most stable, fairness remains near-ideal even as overlap shrinks, conflict rises, or n grows. GTG converges to high fairness with sufficient rounds but is more sensitive to conflict. MCP is fair when overlap is high and the attribute space is small; otherwise it drifts. TILI shows the largest fairness asymmetry, especially under harder scenarios. Figure 4 shows fairness vs. number of strategy parameters indicates GTC near-constant (from 1.00 to 0.97 at n=10), GTG high but gently declining (0.98 $\rightarrow$ 0.88), MCP moderate (0.96 $\rightarrow$ 0.80), and TILI dropping steeply (1.00 $\rightarrow$ 0.48).

Overall, GTC is the most robust across sensitivities—highest agreement, best-balanced utilities, strong Nash product, and stable fairness. GTG is a solid second choice in benign-to-moderate circumstances. MCP is competitive in low-dimensional, high-overlap settings. TILI is fast but fragile, with outcomes skewed toward the proposer and poor performance as difficulty increases.

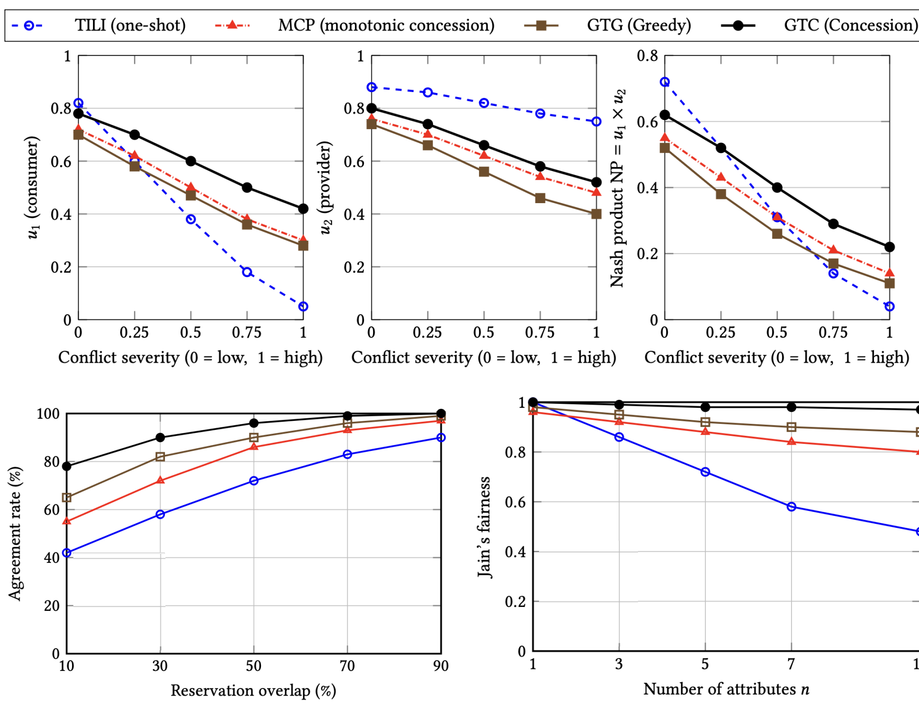
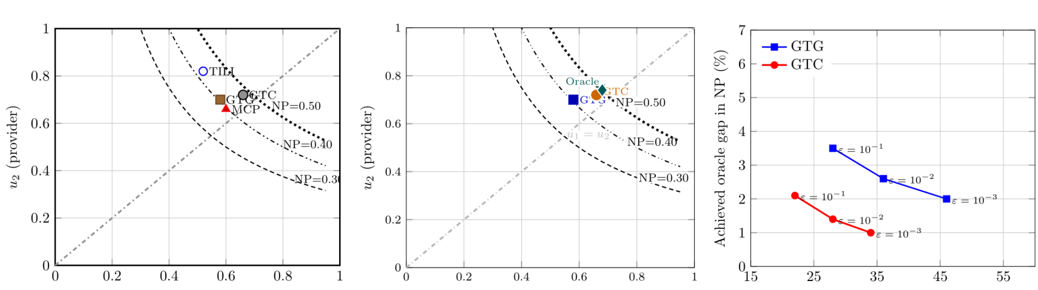


Fig. 4. Utilities and Nash product vs. conflict severity, Agreement rate vs. reservation overlap, and Fairness vs. dimensionality.

5.3.3 Ablations. We also perform ablation studies on GTC by selectively removing or simplifying components of the protocol to quantify their contribution. Concretely, we (i) remove the concession



(a) Utility balance vs. efficiency. (b) Outcome map on Bank. (c) Cost–accuracy trade-off.

Fig. 5. Utility balance between consumer and provider, comparison against oracle and convergence speed against oracle gap for both heuristics.

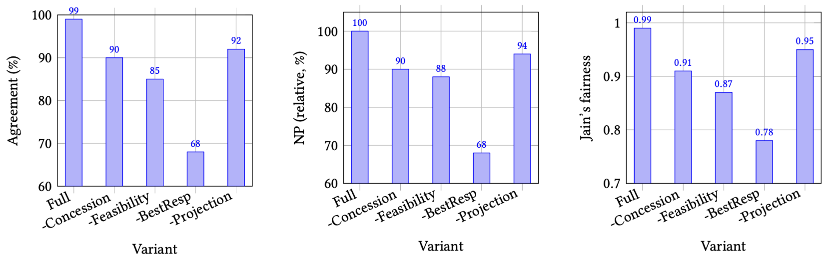


Fig. 6. Ablations on the GTC variant.

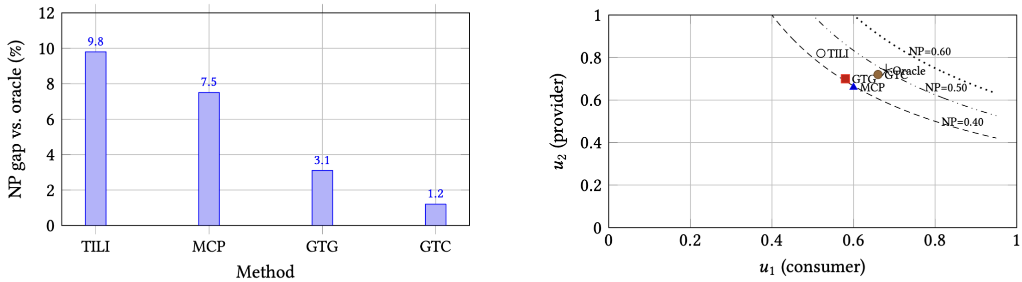
step in proposal generation function f, (ii) disable feasibility–guided proposal generation (no reservation bounds considers), (iii) replace the best–response with a naïve/random choice, and (iv) substitute the indifference–surface projection with a simple coordinate tweak. We run all settings and report the change in performance metrics. Large degradations upon removing a component indicate that component is critical; negligible changes suggest redundancy or over–engineering. Figure 6 show the causal role of the key components.

- Without concession agreement drops by 6–10pp; NP decreases $\approx$ 10%. This confirms targeted concession is the main accelerator of convergence.
- Infeasible proposals appear in 12–20% of rounds (then rejected), agreement falls 10–18pp, and fairness deteriorates (median −0.08). Considering reservation bounds is thus critical for stability and sample–efficiency of the search.
- No best–response causes Agreement rate collapses (often $<$ 70%), NP drops $>$ 30%, and variance across runs doubles.

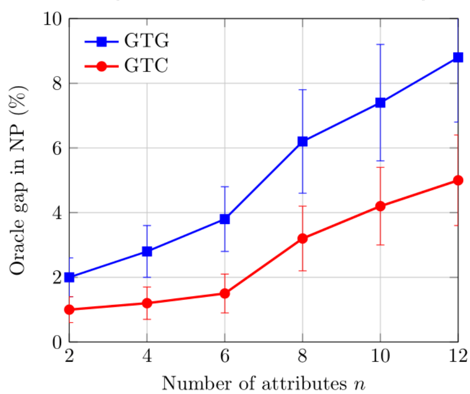
5.3.4 Oracle comparisons. Finally, we compute the *oracle centralised* NBS by directly maximising the Nash (utility) product as $\mathrm{NP}(\vec{\mathbf{v}}) = \big(u_1(\vec{\mathbf{v}}) - q_1\big)\big(u_2(\vec{\mathbf{v}}) - q_2\big)$ over the feasible set $\mathcal{F}$, where computationally feasible, $\vec{\mathbf{v}}^\star \in \arg\max_{\vec{\mathbf{v}}\in\mathcal{F}} \mathrm{NP}(\vec{\mathbf{v}})$ assuming full information and a central solver.[11] Hierarchical attributes are discretised over their DPV/ODRL vocabularies; continuous attributes (e.g., price, duration) obey bounded ranges; policy feasibility and reservation bounds are enforced exactly. We compute approximate Nash-optimal solutions by scanning through discrete attribute combinations (a grid) and optimizing continuous parameters (like price) with one-dimensional search, skipping infeasible points early. On small problems (e.g., the Bank dataset), this approach produces the same results as a full nonlinear solver, while on larger ones it is used alone for scalability. We then compare negotiated outcomes $\hat{\vec{\mathbf{v}}}$ (GTG/GTC) to $\vec{\mathbf{v}}^\star$ using: (i) NP gap $\mathrm{gap}_{\mathrm{NP}} = \frac{\mathrm{NP}(\vec{\mathbf{v}}^\star)-\mathrm{NP}(\hat{\vec{\mathbf{v}}})}{\mathrm{NP}(\vec{\mathbf{v}}^\star)}$, (ii) utility regrets $r_i = u_i(\vec{\mathbf{v}}^\star) - u_i(\hat{\vec{\mathbf{v}}})$, and (iii) distance to the Pareto frontier (dominance checks).

When the number of attributes are low to mid ($n \leq 6$) and reservation values overlap is moderate, *GTC* matches or closely tracks the oracle. median $\mathrm{gap}_{\mathrm{NP}} \leq 1.5\%$ and utility regrets $r_1, r_2 \leq 0.02$ (utilities normalised to $[0, 1]$). *GTG* remains competitive (median $\mathrm{gap}_{\mathrm{NP}} \approx 2$–$4\%$). When the number of attributes increases ($n = 8$–12), negotiated outcomes are still near–optimal; *GTC* median NP gap is 3–5% (and even in the worst 5% of runs it is $\leq$ 8%) and *GTG* is 5–9% (Figure 7b). In all settings we examined, $\hat{\vec{\mathbf{v}}}$ is undominated (Pareto-efficient among the outcomes we observed) and one player's utility can't improve without reducing the other's. The KKT condition [33] (a standard optimality check for constrained optimisation) at $\hat{\vec{\mathbf{v}}}$ correlates with the stopping tolerance ε used

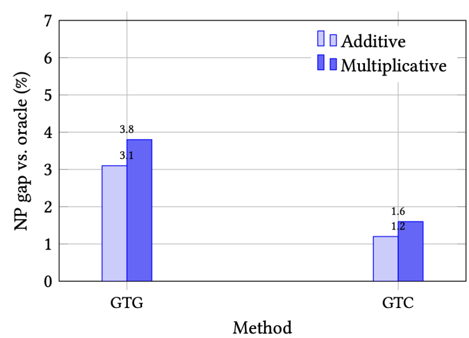
[11]In our instances with linear–additive utilities and convex $\mathcal{F}$, this reduces to a smooth, convex program in the utility image; we solve it to numerical optimality and treat the result as the oracle NBS.



(a) Oracle comparisons on Bank. Left: NP gap w.r.t. oracle (median over 50 runs). Right: (u_1, u_2) outcomes over iso–Nash-product curves with the oracle point.



(b) Oracle gap in Nash product vs. number of attributes n.



(c) Oracle NP gaps for **GTG** vs. **GTC** under *additive* and *multiplicative* utilities.

Fig. 7. Metrics for Oracle

for convergence. Tightening ε from 10^{-2} to 10^{-3} reduces the *GTC* NP gap by $\approx$ 30–40% at the cost of 1.2–1.5× more rounds (consistent with our convergence–efficiency section 4.2). Overall, *GTC* delivers oracle–level outcomes when the oracle is computable, and remains within a small, predictable gap as the number of attributes increase.

Figure 5c quantifies the cost–accuracy trade-off. Figure 5b maps outcomes in the (u_1, u_2) plane Figure 7a summarises both NP gap to the oracle and the (u_1, u_2) outcomes among all methods. We also evaluate the impact of two utility functions, additive and multiplicative. Figure 7c shows NP gaps for GTC and GTG.

5.4 Proof-of-Concept (PoC) System

To demonstrate end-to-end feasibility, we built a small PoC in the context of Negotiation and Contracting Plugin in UPCAST project [12] [30, 31]. The PoC runs our protocol natively in *NegMAS* [57] and exposes a lightweight *React* UI for side-by-side proposal comparison and automatic ODRL policy generation at acceptance.

[12]https://www.upcast-project.eu/

5.4.1 User Study. We recruited 32 data/ML practitioners who used the UI to negotiate access permissions to their actual personal data. Participation was voluntary and uncompensated.[13]

We randomly grouped participants into 16 groups of two. Each group completed two negotiation sessions: one with *GTC* (Concession) and one with *GTG* (Greedy). Participants acted as data *consumers* and *providers*. Data providers submitted their data specifications. Both providers and consumers then ran their respective agents by setting the level of importance for each strategy parameter as weights and their preferences as reservation values. They mutually agreed on utility functions, linear or multiplicative and heuristics, greedy or concession their agents apply during a negotiation session. The agents then start the negotiation and report the results: either an agreement and return the agreed proposal, or disagreement usually because of no or very tight overlap between parties' preferences. We measured: (i) *Agreement rate*; (ii) *time-to-agreement* (minutes); (iii) participants' utility u_1 and u_2 at acceptance; (iv) *fairness* (v) 5-point Likert ratings [14] [5] for *clarity* and *perceived fairness*. We log proposals traces and final ODRL JSON-LD of the negotiated policies.

Results show that agreement rate using GTC is 100% vs. 98% for GTG; median time-to-agreement using GTC is 2.1 min vs. 3.0 min for GTG; Jain fairness using GTC is 0.96 vs. 0.92 using GTG. Finally, subjective ratings (median) for both clarity and fairness is 4/5 using GTC vs. 3.5/5 for GTG.

6 Related Work

We organise the literature into four categories: (i) data–market mechanisms (consortia formation, incentives, auctions, contracts), (ii) automated negotiation, (iii) bargaining theory, and (iv) data valuation and pricing. We then position our contribution relative to the most closely related work, including [9, 14], and provide a brief, systematic comparison to recent papers on pricing and game-theoretic data markets.

Data–market mechanisms. A growing body of work studies how data should be traded or pooled via marketplaces, with emphasis on incentive design, governance, and consortium formation. [14] proposes *data-sharing markets for forming consortia and designing incentives*, focusing on how to align utility among multiple producers/consumers and when coalitions should emerge. Related contributions and collaborators include a data market platform vision [26], defences against strategic buyers in data markets [13], and systems work on secure data access, auditable computation, and trusted data exchange infrastructures [13, 25, 26]. This line complements auction- and contract-based approaches that seek efficient allocations and truthful revelation via classical market mechanisms (e.g., posted prices, Myerson-style auctions, or contract menus). Systematic reviews such as [2] survey industrial data marketplaces and highlight organisational, legal, and technical challenges. Recent domain-specific work (e.g., [51]) investigates data measurement and quality signals in decentralised markets, while [71] proposes conformal tests to guard against contamination in traded/shared data. Our work differs in scope; we *do not* design incentives for coalition/consortia formation; instead, we study the *bilateral*, agent-to-agent bargaining process over *multiple* issues (query scope, purpose/usage policy, duration, third-party sharing, price), and we provide algorithmic and convergence guarantees for the negotiation protocol itself.

Automated negotiation (multi-issue and agent-based). Classical agent-based negotiation has long addressed multi-issue settings where parties exchange offers over several attributes under private preferences; see early frameworks and heuristics [23] and surveys [24, 39]. Typical ingredients are utility models, concession strategies, and acceptance rules under reservation values. Our protocol follows this tradition but tailors it to data-sharing; (i) we support *heterogeneous*

[13]The study used anonymous and non-sensitive logs, obtaining informed consent, not collecting personally identifying data.

[14]The Likert-type scale measures attitudes, opinions, or perceptions, typically structured on a five-point scale ranging from "Strongly Disagree" to "Strongly Agree".

attributes (numerical, hierarchical/ontology-based via DPV/ODRL, and query-based views), (ii) we integrate *policy semantics* for usage control, and (iii) we analyse convergence to the Nash Bargaining Solution (NBS) using Zangwill's theorem, making assumptions explicit and verifiable.

Bargaining theory and equilibrium notions. Foundational results include Nash's axiomatic solution [61], Kalai–Smorodinsky [40], and strategic models such as Rubinstein's alternating-offers game (and standard refinements of equilibria). Potential-game perspectives [58] and cooperative solution concepts (e.g., Shapley value [67]) inform fairness/efficiency trade-offs; rationalisability and incomplete information are treated in, (e.g., [35–37, 56]). We use the Nash product as the objective over the feasible set. Under mild regularity assumptions, our distributed, turn-by-turn best-response with concession steps converges to the NBS, linking practical agent heuristics to classic bargaining theory.

Data valuation and pricing. Query-based pricing provides arbitrage-free posted-price systems where the cost of a query equals the minimum price of determining base views [47]. This thread connects to recent work at the intersection of pricing and games; [10] studies strategic behaviour around data pricing mechanisms, [4] develop theoretical foundations for data-exchange economies (equilibria, welfare), [52] discuss a practical pricing framework for data exchange. Closer to strategic behaviour, [10] studies data pricing under non-cooperative game theory, characterising equilibria and strategic responses of market participants; this complements arbitrage-free posted pricing by analysing strategic dynamics beyond static prices. Our instantiation uses query pricing as one building block for local utility over query/view attributes; however, our emphasis is on *multi-issue negotiation* (price plus usage/control policies and query scope) rather than pricing alone.

Bargaining-based data markets. [9] studies *bargaining-based data markets*, casting bilateral agreements as bargaining problems and analysing outcomes under different preference structures and protocols. Pricing-game analysis in [10] is complementary to their bargaining model [9]. While the former studies strategic pricing interactions, the latter addresses bilateral bargaining over multi-issue contracts. Relative to these works, our contribution is complementary; (i) we explicitly accommodate *policy-rich* attributes (purpose, third-party, action, actor) grounded in ODRL/DPV; (ii) we include *query-based* attributes with practical evaluation/ordering (e.g., pricing or overlap); (iii) we provide a *distributed algorithm* with best-response and concession heuristics and prove convergence to the NBS via Zangwill's GCT [74] under stated assumptions; and (iv) we align with data-space standards (e.g., IDSA CNP) to ease deployment.

Standards and policy semantics. Legal/usage-control semantics matter for practical data-sharing. We model policies in ODRL (W3C) and adopt the Data Privacy Vocabulary (DPV) for hierarchical concepts (purpose, actor, action), enabling machine-processable constraints and reservation bounds that are comparable across parties. This complements marketplace-centric surveys (e.g., [2]) by providing a concrete formal layer on which automated negotiation can operate.

In summary, prior work has advanced *market design* (pricing, incentives, consortia) and *agent negotiation* (multi-issue heuristics) largely in isolation. Taken together, these lines of work—on consortium formation and incentives [14], data market platforms [26], strategic-buyer defences [13], controlled data access and trusted data sharing infrastructures [13, 25, 26], bargaining-based markets [9], query pricing [47], strategic pricing under non-cooperative game theory [10], exchange economies [4], decentralised measurement [51], contamination testing [71], and pricing frameworks [52]—frame our contribution. We unify these strands for the bilateral, policy-rich data-sharing setting by modelling heterogeneous attributes (queries and policies), aligning with standards (ODRL/DPV, IDSA CNP), and delivering an implementable, distributed protocol that operationalises these ideas as concrete proposals, utilities, and turn-by-turn dynamics, with provable convergence to the NBS under transparent assumptions.

7 Discussion and Future Directions

In real-world data-sharing scenarios, such as negotiations between a data provider and a consumer, the proposed framework facilitates adaptive and efficient resolution of key negotiation factors, including price, query, and data usage rules. The alternating proposal mechanism allows both parties to iteratively adjust their proposals, dynamically balancing their priorities to maximise utility while respecting constraints. By integrating game theory principles and employing a multi-issue utility function, our framework supports rational decision-making and ensures continuous utility improvement throughout the negotiation process. This structured approach enables flexible and adaptive negotiations, resulting in mutually beneficial agreements that address the complexities of modern data-sharing environments.

Experimental results clearly demonstrate the effectiveness of our game theory-based protocol in automating complex multi-issue negotiations. Compared to the Take-It-or-Leave-It approach and the Monotonic Concession Protocol, our framework consistently achieves superior outcomes in terms of agreement rates, utility balance, and fairness for both data providers and consumers. The flexibility of our protocol allows both parties to negotiate multiple issues simultaneously, and has proven effective in fostering efficient and mutually beneficial agreements.

The comparison with the Take-It-or-Leave-it and Monotonic Concession benchmarks highlights the limitations of rigid, one-sided negotiation strategies. These approaches often fail to address the nuanced, multi-issue nature of data-sharing negotiations. In contrast, our protocol's multi-issue flexibility significantly improves negotiation outcomes, achieving higher utility for both parties and better accommodating their diverse objectives.

Additionally, comparing with the oracle centralised NBS gives a principle upper bound on efficiency and fairness, and let us quantify how close our distributed protocol gets to the optimum and where any loss comes from. Results show that our protocol is both efficient and controllably accurate, and in low-mid dimensions it can matches the oracle.

Several promising avenues for future research arise from this work. First, integrating machine learning techniques could enable agents to learn from historical negotiations, improving their ability to predict and adapt to opponent strategies in real-time. This would enhance the adaptability of the framework and support more efficient negotiation processes. Second, expanding the framework to accommodate multi-party negotiations—where multiple data consumers, providers, or third-party stakeholders are involved would introduce new complexities and reflect real-world data-sharing scenarios better. Addressing the challenges of multi-party negotiations could unlock broader applications in collaborative data marketplaces and consortia. Lastly, incorporating real-time regulatory compliance mechanisms into the negotiation framework is essential. As data-sharing agreements increasingly intersect with evolving privacy laws and data protection regulations, integrating mechanisms to ensure adherence to these requirements would enhance the framework's relevance and usability. By addressing regulatory concerns dynamically, the framework could better align with the operational needs of modern data markets while supporting robust, legally compliant agreements. Through these advancements, the proposed framework has the potential to become a cornerstone in the evolution of automated, adaptive, and efficient data-sharing negotiation systems, driving innovation and growth in the digital economy.

Acknowledgments

This research was partially funded by the Horizon Europe projects DATAPACT (No. 101189771) and RAISE Suite (No. 101188337), and the UKRI Horizon Europe Guarantee Scheme for projects UPCAST (No. 101093216) and RAISE (No. 101058479).

References

[1] 2024. *Dataspace Protocol: Contract Negotiation Protocol (Version 2024-1).* Technical Report. International Data Spaces Association (IDSA). https://docs.internationaldataspaces.org/ids-knowledgebase/dataspace-protocol/contract-negotiation/contract.negotiation.protocol Specification.

[2] Antragama Ewa Abbas, Wirawan Agahari, Montijn Van de Ven, Anneke Zuiderwijk, and Mark De Reuver. 2021. Business data sharing through data marketplaces: A systematic literature review. *Journal of Theoretical and Applied Electronic Commerce Research* 16, 7 (2021), 3321–3339.

[3] Serge Abiteboul, Richard Hull, and Victor Vianu. 1995. *Foundations of Databases.* Addison-Wesley.

[4] Hannaneh Akrami, Bhaskar Ray Chaudhury, Jugal Garg, and Aniket Murhekar. 2024. On the Theoretical Foundations of Data Exchange Economies. *arXiv preprint arXiv:2412.01968* (2024). doi:10.48550/arXiv.2412.01968

[5] William E Arnold, James C McCroskey, and Samuel VO Prichard. 1967. The Likert-type scale. (1967).

[6] Richard E. Barlow and Frank Proschan. 1975. *Statistical Theory of Reliability and Life Testing: Probability Models.* Holt, Rinehart and Winston, New York. Classical reference on multiplicative success probabilities.

[7] HH Bauschke. 2011. Convex Analysis and Monotone Operator Theory in Hilbert Spaces.

[8] Valerie Belton and Theodor J. Stewart. 2002. *Multiple Criteria Decision Analysis: An Integrated Approach.* Springer, Boston, MA. Discusses compensatory and non-compensatory preference models.

[9] Yuran Bi, Jinfei Liu, Kui Ren, Yihang Wu, and Yang Cao. 2025. Bargaining-Based Data Markets. In *2025 IEEE 41st International Conference on Data Engineering (ICDE).* IEEE, 1456–1469.

[10] Yuran Bi, Yihang Wu, Jinfei Liu, and Li Xiong. 2024. When Data Pricing Meets Non-Cooperative Game Theory. In *2024 IEEE 40th International Conference on Data Engineering (ICDE).* IEEE, 5548–5559.

[11] Sizhen Bian, Vitor Fortes Rey, Siyu Yuan, and Paul Lukowicz. 2025. Hybrid CNN-Dilated Self-attention Model Using Inertial and Body-Area Electrostatic Sensing for Gym Workout Recognition, Counting, and User Authentification. *arXiv preprint arXiv:2503.06311* (2025).

[12] Stephen Boyd and Lieven Vandenberghe. 2004. *Convex Optimization.* Cambridge University Press, Cambridge, UK.

[13] Raul Castro Fernandez. 2022. Protecting Data Markets from Strategic Buyers. In *Proceedings of the 2022 ACM SIGMOD International Conference on Management of Data (SIGMOD '22).* ACM, New York, NY, USA, 1755–1769. doi:10.1145/3514221.3517849

[14] Raul Castro Fernandez. 2023. Data-sharing markets: model, protocol, and algorithms to incentivize the formation of data-sharing consortia. *Proceedings of the ACM on Management of Data* 1, 2 (2023), 1–25.

[15] Soulmaz Cheisari, Jaime Salas, and George Konstantindis. 2025. Towards Compliant and Trustworthy Data Markets forDistributed AI. In *Proceedings of the 2025 7th International Conference on Distributed Artificial Intelligence.* 150–160.

[16] Siqi Chen and Gerhard Weiss. 2019. Automated negotiation: an efficient approach to interaction among agents. In *Interactions in Multiagent Systems.* World Scientific, 149–177.

[17] Y. Chen, W. Zhang, and K. R. Liu. 2018. Game-theoretic modeling of joint data privacy and security in interconnected networks. *IEEE Transactions on Information Forensics and Security* (2018).

[18] Mung Chiang, Steven H. Low, A. R. Calderbank, and John C. Doyle. 2007. Layering as Optimization Decomposition: A Mathematical Theory of Network Architectures. *Proc. IEEE* 95, 1 (2007), 255–312.

[19] World Wide Web Consortium. 2024. Open Digital Rights Language (ODRL) Version 2. http://www.w3.org/ns/odrl/2/.

[20] Eddie Dekel and Faruk Gul. 1997. Rationality and knowledge in game theory. *Econometric Society Monographs* 26 (1997), 87–172.

[21] Muhammad Emad-Ud-Din and Ya Wang. 2023. Promoting occupancy detection accuracy using on-device lifelong learning. *IEEE Sensors Journal* 23, 9 (2023), 9595–9606.

[22] U. Endriss. 2006. Monotonic concession protocols for multilateral negotiation. In *Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems.* 392–399.

[23] Peyman Faratin, Carles Sierra, and Nicholas R. Jennings. 1998. Negotiation Decision Functions for Autonomous Agents. *Robotics and Autonomous Systems* 24, 3–4 (1998), 159–182. doi:10.1016/S0921-8890(98)00029-3

[24] S. Fatima, S. Kraus, and M. Wooldridge. 2014. *Principles of Automated Negotiation.* Cambridge University Press.

[25] Raul Castro Fernandez, Ioannis Alagiannis, Boris N. Alexeev, Michael Cafarella, Sudeepa Roy, Tim Kraska, Samuel Madden, and Michael Stonebraker. 2022. Data Station: Delegated, Trustworthy, Auditable Data Science. *Proc. VLDB Endow.* 15, 12 (2022), 3526–3539. doi:10.14778/3565838.3565865

[26] Raul Castro Fernandez, Kyle Chard, Ben Blaiszik, Sanjay Krishnan, Aaron Elmore, Ziad Obermeyer, Josh Risley, Sendhil Mullainathan, Michael Franklin, and Ian Foster. 2020. The Data Station: Combining Data, Compute, and Market Forces. *arXiv preprint arXiv:2009.00035* (2020).

[27] Francesca Flamini. 2023. The Role of the Status-Quo in Dynamic Bargaining. *Games* 14, 3 (2023), 35.

[28] D. Fudenberg and J. Tirole. 1991. *Game Theory.* MIT Press.

[29] Wei Gao, Mohammad Reza Farahani, Adnan Aslam, and Sunilkumar Hosamani. 2017. Distance learning techniques for ontology similarity measuring and ontology mapping. *Cluster Computing* 20, 2 (2017), 959–968.

[30] Soulmaz Gheisari, Christopher Maidens, Semih Yumusak, Jaime Osvaldo Salas, Paolo Pareti, George Giamouridis, Miao Hu, Adeel Aslam, Bijay Prasad Jaysawal, Luis-Daniel Ibáñez, George Konstantinidis, and Dumitru Roman. 2025. Core Services for Compliant and Trustworthy Data Marketplaces: the EU UPCAST Project. In *RuleML+RR'25: Companion Proceedings of the 9th International Joint Conference on Rules and Reasoning (CEUR Workshop Proceedings, Vol. 4083)*. CEUR-WS.org, Istanbul, Turkey, 1–6. https://ceur-ws.org/Vol-4083/paper70.pdf

[31] Soulmaz Gheisari, Semih Yumusak, Jaime Osvaldo Salas, Luis-Daniel Ibáñez, George Konstantinidis, and Dumitru Roman. 2024. Towards Modular Data Marketplaces. In *RuleML+RR'24: Companion Proceedings of the 8th International Joint Conference on Rules and Reasoning (CEUR Workshop Proceedings, Vol. 3816)*. CEUR-WS.org, Bucharest, Romania, 1–5. https://ceur-ws.org/Vol-3816/paper61.pdf

[32] P. B. Goes, M. Lin, and C. M. Au Yeung. 2014. Popularity Effect in user-generated content: Evidence from online product reviews. *Information Systems Research* (2014).

[33] Geoff Gordon and Ryan Tibshirani. 2012. Karush-kuhn-tucker conditions. *Optimization* 10, 725/36 (2012), 725.

[34] Michel Grabisch. 1996. The application of fuzzy integrals in multicriteria decision making. *European journal of operational research* 89, 3 (1996), 445–456.

[35] John C. Harsanyi. 1967. Games with Incomplete Information Played by "Bayesian" Players, I: The Basic Model. *Management Science* 14, 3 (1967), 159–182.

[36] John C. Harsanyi. 1968. Games with Incomplete Information Played by "Bayesian" Players, II: Bayesian Equilibrium. *Management Science* 14, 5 (1968), 320–334.

[37] John C Harsanyi. 2020. PLAYED BY "BAYESIAN" PLAYERS. *Classics in Game Theory* (2020), 216.

[38] Raj Jain, Dah-Ming W Chiu, and William R Hawe. 1984. A quantitative measure of fairness and discrimination for resource allocation in shared computer systems. *DEC Research Report TR-301* (1984).

[39] Nicholas R. Jennings, Peyman Faratin, Alessio R. Lomuscio, Simon Parsons, Carles Sierra, and Michael Wooldridge. 2001. Automated Negotiation: Prospects, Methods and Challenges. *Group Decision and Negotiation* 10, 2 (2001), 199–215. doi:10.1023/A:1008746126376

[40] Ehud Kalai and Meir Smorodinsky. 1975. Other Solutions to Nash's Bargaining Problem. *Econometrica* 43, 3 (1975), 513–518. doi:10.2307/1914280

[41] Ralph L. Keeney. 1974. Multiplicative Utility Functions. *Operations Research* 22, 1 (1974), 22–34.

[42] R. L. Keeney and H. Raiffa. 1976. *Decision analysis with multiple conflicting objectives*. Wiley & Sons.

[43] Ralph L. Keeney and Howard Raiffa. 1976. *Decisions with Multiple Objectives: Preferences and Value Tradeoffs*. Wiley, New York. Classic text on multi-attribute utility theory.

[44] Frank P. Kelly, Aman K. Maulloo, and David K. H. Tan. 1998. Rate Control for Communication Networks: Shadow Prices, Proportional Fairness and Stability. *Journal of the Operational Research Society* 49, 3 (1998), 237–252.

[45] J. Kleinberg and E. Tardos. 2005. *Algorithm Design*. Addison Wesley Publisher.

[46] George Konstantinidis, Jet Holt, and Adriane Chapman. 2021. Enabling personal consent in databases. *Proceedings of the VLDB Endowment* 15, 2 (2021), 375–387.

[47] Paraschos Koutris, Prasang Upadhyaya, Magdalena Balazinska, Bill Howe, and Dan Suciu. 2015. Query-Based Data Pricing. *J. ACM* 62, 5, Article 43 (Nov. 2015), 44 pages. doi:10.1145/2770870

[48] N. Kshetri. 2014. Big data's impact on privacy, security and consumer welfare. *Telecommunications Policy* (2014).

[49] K. Leyton-Brown and Y. Shoham. 2022. *Essentials of game theory: A concise multidisciplinary introduction*. Springer Nature.

[50] K. Ligett and A. Roth. 2012. Take it or leave it: Running a survey when privacy comes at a cost. In *International workshop on internet and network economics*. 378–391.

[51] Charles Lu, Mohammad Mohammadi Amiri, and Ramesh Raskar. 2024. Data measurements for decentralized data markets. *arXiv preprint arXiv:2406.04257* (2024).

[52] Rupsa Majumdar, Anjula Gurtoo, and Minnu Maileckal. 2025. Developing a data pricing framework for data exchange. *Future Business Journal* 11, 1 (2025), 4. doi:10.1186/s43093-025-00422-z

[53] A. Mas-Colell, M. D. Whinston, and J. R. Green. 1995. *Microeconomic theory*. Vol. 1. 545–577 pages.

[54] Andreu Mas-Colell, Michael Dennis Whinston, Jerry R Green, et al. 1995. *Microeconomic theory*. Vol. 1. Oxford university press New York.

[55] Gerome Miklau and Dan Suciu. 2004. A formal analysis of information disclosure in data exchange. In *Proceedings of the 2004 ACM SIGMOD international conference on Management of data*. 575–586.

[56] Paul Milgrom and John Roberts. 1990. Rationalizability, Learning, and Equilibrium in Games with Strategic Complementarities. *Econometrica* 58, 6 (1990), 1255–1277. doi:10.2307/2938316

[57] Yasser Mohammad, Shinji Nakadai, and Amy Greenwald. 2019. Negmas: a platform for situated negotiations. In *International Workshop on Agent-Based Complex Automated Negotiation*. Springer, 57–75.

[58] Dov Monderer and Lloyd S. Shapley. 1996. Potential Games. *Games and Economic Behavior* 14, 1 (1996), 124–143. doi:10.1006/game.1996.0044

[59] S. Moro, P. Cortez, and P. Rita. 2014. A data-driven approach to predict the success of bank telemarketing. *Decision Support Systems* 62 (2014), 22–31.
[60] Alan Nash, Luc Segoufin, and Victor Vianu. 2010. Views and queries: Determinacy and rewriting. *ACM Transactions on Database Systems (TODS)* 35, 3 (2010), 1–41.
[61] J. Nash. 1950. The Bargaining Problem. *Econometrica* 18, 2 (1950), 155–162.
[62] John F. Nash. 1950. The Bargaining Problem. *Econometrica* 18, 2 (1950), 155–162.
[63] Martin J Osborne and Ariel Rubinstein. 1990. *Bargaining and markets.* Academic Press Limited.
[64] H. J. Pandit, B. Esteves, G. P. Krog, P. Ryan, D. Golpayegani, and J. Flake. 2024. Data Privacy Vocabulary (DPV)–Version 2. arXiv preprint arXiv:2404.13426.
[65] S. A. Raza, K. S. Qureshi, M. Iqbal, and Z. Haider. 2022. Context Aware Win-Win Situation For Automatic Negotiation In Intelligent Agents. *Journal of NCBAE* 1, 2 (2022), 31–44.
[66] Jaime Osvaldo Salas, Paolo Pareti, Semih Yumuşak, Soulmaz Gheisari, Luis-Daniel Ibáñez, and George Konstantinidis. 2025. Evaluation and Comparison Semantics for ODRL. In *Proceedings of the 14th International Joint Conference on Knowledge Graphs (IJCKG 2025)*. Springer Nature. arXiv:2509.05139 [cs.AI]
[67] Lloyd S. Shapley. 1964. Some Topics in Two-Person Games. In *Advances in Game Theory*, Melvin Dresher, Lloyd S. Shapley, and Albert W. Tucker (Eds.). Annals of Mathematics Studies, Vol. 52. Princeton University Press, Princeton, NJ, 1–28. doi:10.1515/9781400882014-002
[68] Markus Spiekermann. 2019. Data marketplaces: Trends and monetisation of data goods. *Intereconomics* 54, 4 (2019), 208–216.
[69] Simon Steyskal and Axel Polleres. 2015. Towards Formal Semantics for ODRL Policies. In *Rule Technologies: Foundations, Tools, and Applications*, Nick Bassiliades, Georg Gottlob, Fariba Sadri, Adrian Paschke, and Dumitru Roman (Eds.). Springer International Publishing, Cham, 360–375.
[70] Hal R. Varian. 1992. *Microeconomic Analysis* (3rd ed.). W. W. Norton & Company, New York. Covers homothetic and multiplicative utility representations.
[71] Martin Voigt Vejling, Shashi Raj Pandey, Christophe Ange Napoléon Biscio, and Petar Popovski. 2025. Conformal Data Contamination Tests for Trading or Sharing of Data. *arXiv preprint arXiv:2507.13835* (2025).
[72] Data Privacy Vocabulary. 2024. Data Privacy Vocabulary. https://w3id.org/dpv/.
[73] Paul Voigt and Axel von dem Bussche. 2017. *The EU General Data Protection Regulation (GDPR): A Practical Guide.* Springer, Cham. doi:10.1007/978-3-319-57959-7
[74] Willard I Zangwill. 1969. Convergence conditions for nonlinear programming algorithms. *Management Science* 16, 1 (1969), 1–13.
[75] W. Zhang, D. Wang, and J. Liu. 2015. A survey on game-theoretic approaches in wireless sensor networks security. *Journal of Communications and Networks* (2015).

Received July 2025; revised October 2025; accepted November 2025