Co-Utility: Designing Self-Enforcing and Mutually Beneficial Protocols

Josep Domingo-Ferrer, David Sánchez, and Jordi Soria-Comas

Abstract Protocols govern the interactions between agents, both in the information society and in the society at large. Protocols based on mutually beneficial cooperation are especially interesting because they improve the societal welfare and no central authority is needed to enforce them (which eliminates a single point of failure and possible bottlenecks). In order to guide the design of such protocols, we introduce co-utility as a framework for cooperation between rational agents such that the best strategy for each agent is to help another agent achieve her best outcome. Specifically, in this chapter we study and characterize self-enforcing protocols in game-theoretic terms. Then, we use this characterization to develop the concept of co-utile protocol and study under which circumstances co-utility arises.

1 Introduction

A protocol specifies a precise set of rules that govern the interaction between agents performing a certain task; that is, it details the expected behavior of each agent involved in the interaction for the task to be successfully completed. For example, for vehicles to avoid collisions at an intersection, they must wait for the green light.

Josep Domingo-Ferrer
Universitat Rovira i Virgili, Dept. of Computer Science and Mathematics, UNESCO Chair in Data Privacy, Av. Països Catalans 26, E-43007 Tarragona, Catalonia, e-mail: josep.domingo@urv.cat

David Sánchez
Universitat Rovira i Virgili, Dept. of Computer Science and Mathematics, UNESCO Chair in Data Privacy, Av. Països Catalans 26, E-43007 Tarragona, Catalonia, e-mail: david.sanchez@urv.cat

Jordi Soria-Comas
Universitat Rovira i Virgili, Dept. of Computer Science and Mathematics, UNESCO Chair in Data Privacy, Av. Països Catalans 26, E-43007 Tarragona, Catalonia, e-mail: jordi.soria@urv.cat

Also, in information technology there are plenty of protocols: the Internet Protocol (IP) defines how to route information packets in the Internet, the MESI protocol [19] defines how to preserve coherence between cache memories in multiprocessor architectures sharing a main memory, etc.

For protocols to be effective, they must be adhered to. This is not problematic when the participating agent cannot deviate by design, such as in the MESI protocol, but it becomes an issue when the agents are free to choose between following the protocol or not, as it happens with vehicles at a crossing regulated by traffic lights. Although free agents cannot be forced to follow a protocol, rational free agents can be persuaded to do so if the protocol is properly designed. Such properly designed protocols will be called self-enforcing in the sequel. Examples of self-enforcing protocols that can be found in the literature include those involved in rational multiparty computation [4], the shotgun clause [1] (which is a way for two rational agents to agree on the price of an item) and the Vickrey auction [29] (which is a kind of auction in which each rational agent truthfully reports her valuation), among others [23].

While self-enforcement is essential, we are interested in protocols offering more than that: we want self-enforcing protocols that result in mutual help between agents and we call them *co-utile protocols*. A prominent advantage of co-utile protocols is that they promote social welfare. To illustrate, consider an agent that is interested in querying a web search engine but does not want the search engine to learn her queries, because these may disclose her personal features or preferences. If there is another agent also interested in privacy-aware querying, both agents can exchange (some of) their queries (and results), thereby preventing the web search engine from accurately profiling either agent; this results in a mutually beneficial collaboration [6, 7].

Co-utile protocols can be crafted for scenarios where the interests of the agents are complementary (or can be made complementary by adding appropriate incentives), so that helping other agents becomes the best way of pursuing one's own interests. Similar ideas about adding artificial incentives to promote cooperation have been proposed whose scope is narrower and more *ad hoc* than the one of the co-utility framework. For example, in P2P networks for sharing of distributed resources (*e.g.*, storage, computing, data, etc.), incentives are used to achieve self-enforcing collaboration and deter the so-called *free-riders* (that is, peers who use resources from others but who do not offer their own resources) [20]; incentives in this context take the form of better service [2], or some sort of virtual money [12] for those who contribute.

In this chapter, we first formalize the notions of protocol and self-enforcing protocol. Then we move on to define co-utile protocols. Since we are assuming rational agents who freely decide whether to adhere to the protocol or not, game-theoretic modeling arises as the most natural choice. The assumption of free rational agents is plausible in peer-to-peer (P2P) scenarios lacking a central authority and a common legal framework that can be used to enforce a specific behavior.

2 Self-enforcing protocols

Since we focus on protocols, we first need to clarify what we understand by *protocol*. Loosely speaking, a protocol is a sequence of interactions among a community of agents, called steps, that are aimed at carrying out a certain task.

A formalization of the concept of protocol that is often used in computer science is based on *finite state machines*. A finite state machine is a mathematical model of computation. It consists of a set of states, one of which is the current state, and a set of transitions between states that are triggered by specific events and modify the current state. While a finite state machine nicely models a protocol (each step changes from one state to another), it fails to capture the behavior of rational agents who choose their actions with the aim of maximizing their utility.

With rational agents in mind, game theory [14, 18] seems the right mathematical model. This theory models interactions between self-interested agents that act strategically. An agent is self-interested if she defines a preference relation over the set of possible outcomes of the protocol. On the other hand, an agent acts strategically when she takes into account her knowledge and expectations about the state of the world and about other agents to decide on her strategy (the way she plays the game). Game theory identifies subsets of outcomes (a.k.a. solution concepts) that agents would be most interested in achieving. In this work the focus will be on equilibrium solutions, *i.e.*, outcomes which rational agents have no motivation to deviate from.

In our formalization, a game is used to model all the possible interactions among agents in the underlying scenario. In particular, the game includes also those interactions among agents that are not desired. Then, a protocol is regarded as a prescription of a specific behavior in the underlying scenario, that is, a sequence of desired interactions.

The type of interaction between agents is a key point in the outcome of a game. Game theory can model several interaction types, including:

- *Simultaneous and sequential moves.* Moves in a game are called simultaneous if each agent chooses her move independently (unaware) of the other agents' moves. On the other side, moves are called sequential if, at the time of choosing a move, previous moves made by other agents are known (at least to some extent).
- *Perfect and imperfect information.* A sequential game (one with sequential moves) is said to be a perfect-information game if the agent about to make her move has complete knowledge on the previous moves made by the other agents. If the agent's knowledge on previous moves is only partial, the game is said to be an imperfect-information game.
- *Complete and incomplete information.* If the previous category referred to knowledge on previous moves, this category refers to agents' knowledge on the underlying game. In games with complete information, the payoff of each agent at each final state is known by all agents. In games with incomplete information (a.k.a. Bayesian games), an agent is uncertain about the payoffs of the other agents.

The actual formalization of a game depends on the type of interaction one wants to model. We focus on sequential games with perfect information [13], because this is a quite common and basic type of interaction between agents.

The formal definition of a sequential game with perfect information is as follows:

Definition 1.1. *(Perfect-information game).* A perfect-information game (in extensive form) is a tuple $G = (N, A, H, Z, \chi, \rho, \sigma, u)$, where:

- N is a set of n agents;
- A is a set of actions;
- H is a set of non-terminal choice nodes;
- Z is a set of terminal nodes, disjoint from H;
- $\chi : H \to 2^A$ assigns to each choice node a set of possible actions;
- $\rho : H \to N$ assigns to each choice node a player $i \in N$ who chooses an action at that node.
- $\sigma : H \times A \to H \cup Z$ is an injective map that, given a pair formed by a choice node and an action, assigns to it a new choice node or a terminal node;
- $u = (u_1, ..., u_n)$, where $u_i : Z \to R$ is a real-valued utility function for agent i on the terminal nodes.

A perfect-information game can be represented in the so-called *extensive form* as a tree where:

- Each non-terminal choice node is labeled with the name of the agent making the move;
- Each terminal node is labeled with the utility that each agent obtains when reaching it;
- Edges going out from a node represent the actions available to the agent making the move.

Although the focus has been placed on sequential games with perfect information, nothing has been said so far about the completeness of the information. Assuming complete information seems too restrictive when trying to model the interactions between a set of potentially unrelated agents. Fortunately, the above tree representation can easily accommodate both complete and incomplete-information games:

- In games with complete information, the utilities at the terminal nodes are fixed values known to all the agents. Fig. 1 shows sequential adaptations of two well-known games: the Prisoners' Dilemma and the Battle of the Sexes [15]. Both are perfect-information and complete-information games.
- In games with incomplete information, the utilities at terminal nodes are not completely known. This is modeled by replacing the fixed utilities at terminal nodes by utility functions that depend on an additional parameter: the type of each agent. The type of an agent encapsulates all the information on that agent that is not common knowledge. The set of types of the game is the Cartesian product of the set of types of each agent: $\Sigma = \Sigma_1 \times \ldots \times \Sigma_n$. Each agent knows her type but is uncertain about the types of the other agents. The agent models this uncertainty by attributing to every other agent a prior distribution over the possible types. Usually the same prior distribution is assumed for all agents.

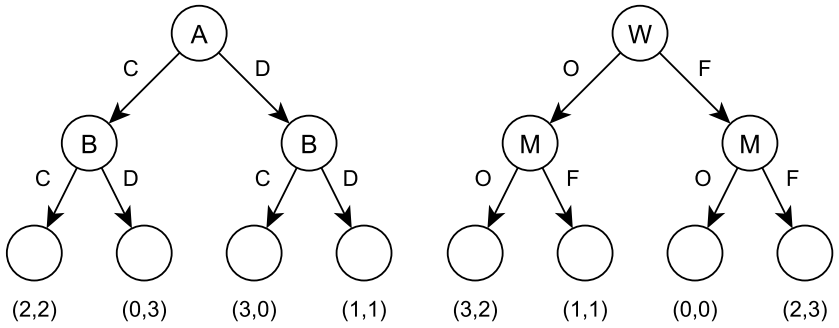

Fig. 1 Sequential versions of two well-known simultaneous games: the Prisoners' Dilemma (left) and the Battle of the Sexes (right). In the first game, prisoner A first decides whether to cooperate (C) with prisoner B, by *not* betraying him, or defect (D), by betraying him; then prisoner B faces the same decision (unawares of A's move in the simultaneous version). The utility of each prisoner is the reduction of the number of years in prison she or he obtains. In the second game, the woman, W, in a couple chooses between going to the opera (O) or to a football match (F); then the man, M, does the same (unawares of W's choice in the simultaneous version). The utility is how much they enjoy themselves: both would like to be together, but W prefers opera and M football.

We have defined a game as containing the set of actions, A, that are available to the agents. The strategies of a game describe the possible ways in which each agent can choose actions. The most basic type of strategies are pure strategies, defined as follows.

Definition 1.2. *(Pure strategy)* Let $G = (N, A, H, Z, \chi, \rho, \sigma, u)$ be a perfect-information game in extensive form. A pure strategy for an agent i is a function s_i that selects an available action at each non-terminal node at which i has to make a decision, that is, at each $h \in H$ such that $\rho(h) = i$. We denote by S_i the strategy set of agent i, that is, the set of pure strategies available to agent i.

In other words, a pure strategy provides a complete definition of how an agent will play a game, because it determines the move the agent will make for any situation he or she could face. Beyond pure strategies, more sophisticated strategies can be defined, such as mixed and correlated strategies, in which the selection of actions at each decision node h is randomized. We can now define the notion of strategy profile.

Definition 1.3. *(Strategy profile)* In a game G with n agents, a strategy profile s is a tuple $s = (s_1, \ldots, s_n)$, where s_i is the strategy chosen by agent i. The set of possible strategy profiles is $S_1 \times \ldots \times S_n$, where S_i is the set of strategies available to agent i.

As introduced above, after modeling the underlying scenario as a game, we can now model a protocol as a sequence of prescribed interactions. That is, a protocol embodies the agents' behavior that the designer wants to favor.

Definition 1.4. *(Protocol)*. Given a perfect-information game G, a protocol P is a set of strategy profiles. If, at each non-terminal node h, agent $\rho(h)$ is allowed only one action by the protocol, then the protocol consists of a single strategy profile.

If G is represented in extensive form as a tree, a protocol can be viewed as a subtree from the root to several leaves. If the protocol consists of a single strategy profile, then it can be viewed as a path from the root to a leaf.

We are interested in self-enforcing protocols, that is, in prescribing behaviors that are adhered to by rational agents. In other words, no rational agent should have any incentive to deviate from the protocol provided that the other agents stick to it. This requirement can be rephrased in game-theoretic jargon by saying that each of the strategy profiles in the protocol must be an equilibrium of the underlying game. There is a large variety of equilibrium concepts such as Nash equilibrium, correlated equilibrium, subgame-perfect equilibrium, etc. In the context of protocols defined over perfect-information sequential games, we expect each agent to select a move that yields an equilibrium of the remaining subgame, that is, of the portion of the game that remains to be played after some moves have already been made by the agents. This kind of equilibrium is known as subgame-perfect equilibrium.

Definition 1.5. *(Subgame)*. Given a game G in extensive form, the subgame of G rooted at node h is the restriction of G to h and its descendants.

Definition 1.6. *(Subgame-perfect equilibrium)*. Given a game G in extensive form, a strategy profile s is a subgame-perfect equilibrium if, for any subgame G' of G, the restriction of s to G' is an equilibrium.

When computing Nash equilibria to check whether Definition 1.6 holds for them, in the case of complete-information games the given utilities are used, whereas in the case of incomplete-information games one resorts to expected utilities (derived from the prior distribution of types).

We are now in a position to formally define self-enforcing protocols using the concept of equilibrium:

Definition 1.7. *(Self-enforcing protocol)*. A protocol P over a perfect-information game G is self-enforcing if no agent can increase her utility by deviating from one of the strategy profiles in P, provided that the other agents stick to that strategy profile. Equivalently, each of the strategy profiles in P is a subgame-perfect equilibrium of G.

In the sequential Battle of the Sexes (BoS) game described in Fig. 1, the protocol $P = (O, O)$ is self-enforcing: (O, O) is a Nash equilibrium of the BoS game, and (O) is a Nash equilibrium of the game that remains after the first action.

An interesting property of subgame-perfect equilibria is that they always exist; that is, every perfect-information game has a subgame-perfect equilibrium. Such an equilibrium is usually found by means of backward induction [14]. The backward induction algorithm assumes that, at each node, the agent making the move selects the action that gives her the best outcome. The algorithm starts evaluating the choice

nodes that are parents of terminal nodes. Once the moves at these choice nodes have been selected, the algorithm proceeds backwards by evaluating the choice nodes that are parents of the previous choice nodes. For example, in the sequential BoS game (see Fig. 1), M is the agent making the last choice. In the left branch, the best option for M is O, which leads to utility $(3,2)$. In the right branch, the best option for M is F, which leads to utility $(2,3)$. Because W knows that M will seek to maximize his own utility, W can simplify the original tree to get a tree with a single choice node (hers) in which choosing O leads to utility $(3,2)$ and choosing F leads to utility $(2,3)$. Thus, in the original (unsimplified) tree with two decision stages, W should select O in the first place, and M should then also select O.

Let us find self-enforcing protocols in the two games proposed in Fig. 1. In the sequential version of the Prisoners' Dilemma, it is easy to see that the only self-enforcing protocol is (D,D). Choosing C is not self-enforcing for any agent because switching from C to D improves the agent's utility (provided that the other agent does not alter his/her move). In the sequential BoS game, there are two self-enforcing protocols: (O,O) and (F,F). In either protocol, no agent can improve her utility by deviating from the protocol provided that the other agent sticks to it. Of course, since in the sequential version agent W chooses her move first, she is most likely to favor (O,O) over (F,F).

3 Co-utility and co-utile protocols

As motivated in the introduction, we are interested in protocols that, beyond being self-enforcing, induce a rational agent to collaborate with other agents to increase their utilities. We say a protocol promotes collaboration between a set of agents if the utility of each participating agent is strictly greater than her utility when not participating (an agent is said to participate in the protocol if and only if she follows the prescriptions of the protocol). More precisely, we are interested in a protocol that promotes the most rewarding collaboration for all agents, that is, such that there is no alternative protocol whereby all agents could get a better outcome and at least one of them a strictly better outcome; in game-theoretic terms, we seek Pareto-optimality of the utilities of agents. This Pareto-optimal collaboration is the essence of co-utility, which can be defined as follows:

Definition 1.8. *(Co-utile protocol).* A self-enforcing protocol P is co-utile if each strategy profile in P is Pareto-optimal.

In the sequential BoS game described in Fig. 1, the protocol $P = (O,O)$ is co-utile. As we have already seen that it is self-enforcing, it only remains to check that it is Pareto optimal: W cannot improve her utility with respect to (O,O); M increases his utility if (F,F) is played, but that decreases the utility of W.

Co-utility is reminiscent of cooperative game theory [3] in that it deals with agents that collaborate to get a more desirable outcome. However, there is a substantial difference: in cooperative game theory, coalitions of agents are formed and

the agents in a coalition coordinate their strategy to maximize the coalition's payoff, which is then divided among the agents. Assuming that the payoff can be divided among the agents in a coalition is essential for the coalitions to be formed. In contrast, in co-utility we assume that each agent acts autonomously and keeps to herself the payoff she gets: this allows co-utility to deal with non-divisible payoffs, such as privacy or security.

On the other hand, since the goal of co-utility is to lead to (co-utile) protocols, mechanism design is a natural framework to compare with. In mechanism design [17, 16], the ultimate goal is to come up with mechanisms that yield a previously defined socially desirable outcome. In this sense, co-utility is less demanding than mechanism design: it does not aim at enforcing a specific socially beneficial outcome, but rather at promoting a mutually beneficial collaboration between agents. It may well happen that the mutually beneficial outcome of a co-utile protocol is also socially desirable (but this does not necessarily happen).

Another difference between mechanism design and co-utility is that the former requires preference alignment, whereas the latter does not:

- In mechanism design, the outcome is selected based on the preferences reported by the agents. Untruthful reporting of preferences by agents in an attempt to obtain a more desirable outcome is an important issue that must be addressed by mechanism design. This is usually tackled by requiring some payment from agents that is calculated to make misrepresentation of utilities unfruitful. This kind of mechanism to align the preferences of the individual agents with the socially desirable outcome is known as incentive-compatible mechanism.
- In co-utility, rather than aligning preferences to the socially desirable outcome via incentives, we seek to promote collaboration between agents that have complementary preferences. For instance, for the case of privacy-preserving query submission to a web search engine, we assume that the agents that participate have complementary preferences regarding the topics they wish to query about, but they are all concerned about their privacy. Moreover, decisions in co-utility are made by the agents themselves rather than in a centralized way: when asked for collaboration, each agent decides which strategy is best for her.

In the sequential version of the Prisoners' Dilemma (Fig. 1), the only self-enforcing protocol is (D,D) but it is not Pareto-optimal because (C,C) provides a strictly greater payoff to both agents. Therefore, in this game no co-utile protocol exists. In the sequential version of the Battle of the Sexes (Fig. 1), it is easy to check that both self-enforcing protocols (O,O) and (F,F) are Pareto-optimal solutions of the game. Thus, both protocols are co-utile.

A specially interesting case of co-utility happens when the protocol is not only Pareto-optimal, but maximizes the utility of all agents:

Definition 1.9. *(Strict co-utility)*. A protocol P on a game G is strictly co-utile if the utility that every agent derives from following any strategy in P is strictly greater than the utility the agent would obtain by following any strategy not in P.

We next prove that strict co-utility as per Definition 1.9 implies co-utility as per Definition 1.8.

Proposition 1.1. *If a protocol P on a game G is strictly co-utile, then it is co-utile.*

Proof. We must check that P is self-enforcing and Pareto-optimal. Since all strategies not in P yield less payoff than any strategy in P, we have that:

1. No agent can increase her payoff by deviating from P and, thus, P is self-enforcing;
2. No protocol P' different from P yields a better utility to all agents and a strictly better utility to at least one agent; hence, P is Pareto-optimal.

Co-utility is about making selfish behavior compatible with mutually beneficial collaboration. However, given a game G, in general there is no guarantee that the selfish behavior of the agents will lead to co-utility, let alone strict co-utility, even if there are co-utile (or strictly co-utile) protocols in the game. However, for some specific games, such a guarantee can be given:

Proposition 1.2. *In a perfect-information game G where all the agents maximize their utilities in exactly the same set of terminal nodes of the tree that represents the game in extensive form (and only in these terminal nodes), selfish behavior by the agents will cause them to follow a strictly co-utile protocol.*

Proof. Let $\mathscr{M}$ be the set of terminal nodes where the utilities of all agents are maximized. Let l be the length of the shortest path from the root to a node in $\mathscr{M}$. We will prove that selfish behavior leads to a node in $\mathscr{M}$ by induction over l. If $l = 1$, there is a single action to be chosen and at least one node $m \in \mathscr{M}$ can be reached by the agent making the choice. This agent could certainly take a longer path (with more than one action to be chosen), but this path should equally lead to a node in $\mathscr{M}$ (otherwise the agent would obtain a suboptimal payoff). We assume that the proposition is satisfied when $l \leq k$ and we need to show that it is also satisfied for $l = k+1$. To apply the induction hypothesis, we split the shortest path to $\mathscr{M}$ into two parts: P_1 (containing the k initial steps) and P_2 (containing the final step). The same reasoning used for $l = 1$ shows that the agent making the decision at $leading(P_2)$, the leading node of P_2, chooses to follow P_2. As we have determined the path that will be followed if $leading(P_2)$ is reached, we can simplify the game tree by removing all the subtrees rooted at $leading(P_2)$ and copying the utilities of m (which are the same as the utilities of any node in $\mathscr{M}$) to $leading(P_2)$. To show that m is reached in the original tree, it is enough to show that $leading(P_2)$ is reached in the simplified tree. But the latter is immediate by applying the induction hypothesis to P_1 , because the terminal node of P_1 is $leading(P_2)$.

A co-utile protocol always prescribes selfish behaviors (because it is self-enforcing) but not all selfish behaviors need to be co-utile. The interest of Proposition 1.2 is that it describes a class of games in which selfish behavior by everyone always yields strict co-utility. This is indeed very helpful to find a co-utile protocol. The alternative to find co-utile protocols is to use the backward induction algorithm (see [14]) on the game tree in order to seek subgame-perfect equilibria. However, this algorithm involves substantial computational cost, because it needs to traverse all the nodes of the tree (from leaves to root).

4 Application fields

Although presenting detailed applications of co-utility will be done in other chapters, we do list here some areas in which co-utility seems to be a useful notion. In some of these applications areas, the interests of agents are naturally complementary (and hence mutual collaboration is a natural way to proceed); in other areas, the agents' interests can be made complementary by properly designing their interactions and/or adding appropriate incentives.

- *Anonymous keyword search*. Web search engines have become an essential tool to find online resources. However, their use can reveal very detailed information about the interests of their users. In [10, 8, 11] a co-utile protocol for exchanging queries between users is proposed. By exchanging queries with other users, the query profile associated to each user is masked and approximates the average group interests. However, the query exchange protocol has a cost. To prevent free-riders (who take advantage of the protocol to mask their own profile but are reluctant to help others) [5] proposes to use a reputation management system (see also the next chapter in this book).
- *Collaborative anonymization*. In surveys collecting individual data (microdata), each respondent is usually required to report values for a set of attributes. If some of these attributes contain sensitive information, the respondent must trust the collector not to make an inappropriate use of the data. If the respondent does not trust the data collector (which is understandable, given the large number of data collectors in our big data society [25]), she can anonymize her data locally by reporting inaccurately or by reporting nothing at all. The sensitivity of a piece of data depends on its rareness within a population [21]. In this respect, local anonymization is clueless in determining the right amount of masking. In [24] a co-utile protocol is proposed whereby users interact to determine the level of sensitivity of their respective pieces of data.
- *Content distribution*. Multicast distribution of content is not suited to content-based electronic commerce because all buyers obtain exactly the same copy of the content, in such a way that unlawful redistributors cannot be traced. Unicast distribution has the shortcoming of requiring one connection with each buyer, but it allows the merchant to embed a different serial number in the copy obtained by each buyer, which enables redistributor tracing. Peer-to-peer (P2P) distribution is a third option that can combine some of the advantages of multicast and unicast: on the one hand, the merchant only needs unicast connections with a few seed buyers, who take over the task of further spreading the content; on the other hand, if a proper fingerprinting mechanism is used, unlawful redistributors of the P2P distributed content can still be traced. In [9], a co-utile fingerprinting mechanism for P2P content distribution if proposed; it allows redistributor tracing, while preserving the privacy of most honest buyers and offering collusion resistance and buyer frameproofness.
- *Distributed computing*. In distributed scenarios, such as cloud provider federations, the Internet of Things, etc., a number of autonomous or entirely indepen-

dent systems are supposed to interact in certain prescribed ways. Since those systems often operate in different legal jurisdictions, compelling them to perform as agreed (even if stated in a contract) may be difficult. Making prescribed interactions rationally sustainable would be hence interesting. Co-utile protocols induced by artificial incentives such as reputation [5] (see also the next chapter in this book) or quality of service could go a long way in this direction.

- *Collaborative economy*. Collaborative markets are booming, such as crowdsourcing [27, 26], P2P lending [28], ridesharing [22], etc. The lack of a common legal framework and, hence, the lack of trust among peers hamper a more generalized take-off of the collaborative economy. Especially interesting would be a fully distributed collaborative economy, to empower the individual agents and overcome the current oxymoron of some platform-owning companies acting as oligopolies in the collaborative market. Just as in the case of distributed computing, suitable artificial incentives like distributed reputation [5] (see also the next chapter) could be leveraged to advance towards a truly distributed collaborative economy.

5 Conclusions

We have presented and formalized *co-utility*, which is a new paradigm to design self-enforcing protocols that make mutual help between rational agents the best strategy. Co-utility has the potential to boost social welfare in a great number of interactions, both in information society and in the society at large.

We have presented a list of possible applications of co-utility. In all of them, the interests of the agents are either complementary or can be made complementary by crafting a specific protocol with appropriate incentives (e.g. reputation management).

Acknowledgements Funding by the Templeton World Charity Foundation (grant TWCF0095/AB60 "CO-UTILITY") is gratefully acknowledged. Also, partial support to this work has been received from the Government of Catalonia (ICREA Acadèmia Prize to J. Domingo-Ferrer and grant 2014 SGR 537), the Spanish Government (projects TIN2014-57364-C2-1-R "SmartGlacis", TIN2015-70054-REDC and TIN2016-80250-R "Sec-MCloud") and the European Commission (projects H2020-644024 "CLARUS" and H2020-700540 "CANVAS"). The authors are with the UNESCO Chair in Data Privacy, but the views in this work are the authors' own and are not necessarily shared by UNESCO or any of the funding bodies.

References

1. Brooks, R. R. W., Landeo, C. M., Spier, K. E. (2010) Trigger happy or gun shy? Dissolving common-value partnerships with Texas shootouts. The RAND Journal of Economics 41(4):649–673.

2. Buragohain, C., Agrawal, D., Suri, S. (2003) A game theoretic framework for incentives in P2P systems. In Shahmehri, N., Graham, R. L., Caronni, G. (eds) Peer-to-Peer Computing, IEEE Computer Society, pp. 48–56.
3. Chalkiadakis, G., Elkind, E., Wooldridge, M. (2012) Cooperative game theory: basic concepts and computational challenges. IEEE Intelligent Systems 27(3):86–90.
4. Dodis, Y., Rabin, T. (2007) Cryptography and game theory. In Nisan N. et al. (eds) Algorithmic Game Theory. Cambridge University Press, 2007, pp. 181–205.
5. Domingo-Ferrer, J., Farràs, O., Martínez, S., Sánchez, D. and Soria-Comas, J. (2016) Self-enforcing protocols via co-utile reputation management. Information Sciences 367(C):159–175.
6. Domingo-Ferrer, J., González-Nicolás, Ú. (2012) Rational behavior in peer-to-peer profile obfuscation for anonymous keyword search. Information Sciences 185:191–204.
7. Domingo-Ferrer, J., González-Nicolás, Ú. (2012) Rational behavior in peer-to-peer profile obfuscation for anonymous keyword search: the multi-hop scenario. Information Sciences 200:123–134.
8. Domingo-Ferrer, J., Martínez, S., Sánchez, D., Soria-Comas, J. (2017) Co-utility: self-enforcing protocols for the mutual benefit of participants. Engineering Applications of Artificial Intelligence 59:148–158.
9. Domingo-Ferrer, J., Megías, D. (2016) Co-utility for digital content protection and digital forgetting. In: Proceedings of the 15th Annual Mediterranean Ad Hoc Networking Workshop, MedHocNet 2016, IEEE, pp. 1–7.
10. Domingo-Ferrer, J., Sánchez, D., Soria-Comas, J. (2016) Co-utility: self-enforcing collaborative protocols with mutual help. Progress in Artificial Intelligence 5(2):105–110.
11. Domingo-Ferrer, J., Soria-Comas, J., Ciobotaru, O. (2016) Co-utility: self-enforcing protocols without coordination mechanisms. In: Proceedings of the 2015 International Conference on Industrial Engineering and Operations Management, IEOM 2015, IEEE, pp. 1-17.
12. Friedman, E. J., Halpern, J. Y., Kash, I. A. (2006) Efficiency and nash equilibria in a scrip system for P2P networks. In Eigenbaum, J., Chuang, J. C.-I., Pennock, D. M. (eds) EC'06, ACM, pp. 140–149.
13. Kuhn, H. W. (1953) Extensive games and the problem of information. In Kuhn, H. W., Tucker, A. (eds), Classics in Game Theory. Princeton University Press, pp. 193-216.
14. Leyton-Brown, K., Shoham, Y. (2008) Essentials of Game Theory: A Concise, Multidisciplinary Introduction. Morgan & Claypool.
15. Luce, R.D., Raiffa, H. (1957) Games and Decisions: An Introduction and Critical Survey. John Wiley & Sons.
16. Maskin, E. (1999) Nash equilibrium and welfare optimality. The Review of Economic Studies 66(1):23–28.
17. Nisan, N. (2014) Algorithmic mechanism design. In: Handbook of Game Theory. Elsevier, pp. 477-516.
18. Osborne, M., Rubinstein, A. (1994) A Course in Game Theory. MIT Press.
19. Papamarcos, M. S., Patel, J. H. (1984) A low-overhead coherence solution for multiprocessors with private cache memories. In Agrawal, D. P. (eds) ISCA , ACM, pp. 348–354 .
20. Rahman, R., Vinkó, T., Hales, D., Pouwelse, J. A., Sips, H. J. (2011) Design space analysis for modeling incentives in distributed systems. In Keshav, S., Liebeherr, J., Byers, J. W., Mogul, J. C. (eds) SIGCOMM, ACM, pp. 182–193.
21. Sánchez, D., Martínez, S., Domingo-Ferrer, J. (2016) Comment on 'Unique in the shopping mall: On the reidentificability of credit card metadata'. Science 351:1274.
22. Sánchez, D., Martínez, S., Domingo-Ferrer, J. (2016) Co-utile P2P ridesharing via decentralization and reputation management. Transportation Research Part C 73:147–166.
23. Schneier, B. (2009) The value of self-enforcing protocols. In: ThreatPost. https://threatpost.com/value-self-enforcing-protocols-081009/72980/. Accessed: 17 Jan 2016.
24. Soria-Comas, J., Domingo-Ferrer, J. (2015) Co-utile collaborative anonymization of microdata. In Torra, V., Narukawa, Y. (eds) MDAI. LNCS 9321, Springer, pp. 192–206.

25. Soria-Comas, J., Domingo-Ferrer, J. (2015) Big data privacy: challenges to privacy principles and models. Data Science and Engineering, 1(1):21–28.
26. Turi, A. N., Domingo-Ferrer, J., Sánchez, D., Osmani D. (2015) Co-utility: conciliating individual freedom on common good in the crowd based business model. In: Proceedings of the 12th International Conference on e-Business Engineering, ICBE 2015, IEEE, pp. 62–67.
27. Turi, A. N., Domingo-Ferrer, J., Sánchez, D., Osmani D. (2016) A co-utility approach to the mesh economy: the crowd-based business model. Review of Managerial Science. In press.
28. Turi, A. N., Domingo-Ferrer, J., Sánchez, D. (2016) Filtering P2P loans based on co-utile reputation. In: Proceedings of the 13th International Conference on Applied Computing, AC 2016, pp. 139–146..
29. Vickrey, W. (1961) Counterspeculation, auctions, and competitive sealed tenders. The Journal of Finance 16(1):8–37.