Truthful mechanisms for generalized utilitarian problems

G. Melideo* P. Penna† G. Proietti‡ R. Wattenhofer§ P. Widmayer¶

Abstract

In this paper we investigate extensions of the well-known Vickrey[1961]-Clarke[1971]-Groves[1973] (VCG) mechanisms to problems whose objective function is not utilitarian and whose agents' utilities are not quasi-linear. We provide a generalization of utilitarian problems, termed *consistent* problems, and prove that every consistent problem admits a *truthful mechanism*. These mechanisms, termed *VCG-consistent* mechanisms, can be seen as a natural extension of VCG mechanisms for utilitarian problems.

We then investigate extensions/restrictions of consistent problems. This yields three classes of problems for which (i) VCG-consistent mechanisms are the only truthful mechanisms, (ii) no truthful VCG-consistent mechanism exists, and (iii) no truthful mechanism exists, respectively. Showing that a given problem is in one of these three classes is straightforward, thus yielding a simple way to see whether VCG-consistent mechanisms are appropriate or not.

Finally, we apply our results to a number of basic non-utilitarian problems.

Keywords: Algorithmic Mechanism Design, Algorithms for the Internet, Game Theory

1 Introduction

In the Internet a multitude of heterogeneous entities (e.g., providers, autonomous systems, universities, private companies, etc.) offer, use, and even compete with each other for resources. As the Internet is emerging as the platform for distributed computing, new solutions should take into account the new aspects deriving from a multi-agent system in which agents cannot be assumed to be either *honest/obedient* (i.e., to follow the protocol) or *adversarial* (i.e., to "play against"). Indeed, the entities involved in the computation are driven by different goals (e.g., minimizing *their own* costs) and they may act *selfishly*. In this case, agents cannot be assumed to follow the protocol, though they respond to *incentives* (e.g., a payment received to compensate the costs).

In essence, game theory is the study of what happens when independent agents act selfishly (for a more extensive discussion of applications of game theoretic tools and micro economics to the Internet we refer the reader to [FS02, Pap01]). Mechanism design asks how one can design systems so that agents' selfish behavior results in the desired system-wide goals. In a nutshell, each agent i has a function $u_i(\cdot)$ which expresses her *utility* derived from the system outcome. For instance, if the system computes a solution X and provides agent i with a payment P_i, then the corresponding utility is equal

*Dipartimento di Informatica, Università di L'Aquila, Via Vetoio, 67010 L'Aquila, Italy. E-mail: `melideo@di.univaq.it`.

†Dipartimento di Informatica ed Applicazioni "R.M. Capocelli", Università di Salerno, via **S. Allende** 2, I-84081 Baronissi (SA), Italy. E-mail: `penna@dia.unisa.it`. Supported by the European Project IST-2001-33135, Critical Resource Sharing for Cooperation in Complex Systems (CRESCCO). Work partially done while at the Institut für Theoretische Informatik, ETH Zentrum.

‡CONTACT AUTHOR – Dipartimento di Informatica, Università di L'Aquila, Via Vetoio, 67010 L'Aquila, Italy, and Istituto di Analisi dei Sistemi ed Informatica, CNR, Viale Manzoni 30, 00185 Roma, Italy. E-mail: `proietti@di.univaq.it`. Partially supported by the Research Project GRID.IT, partially funded by the Italian Ministry of Education, University and Research. Part of this work has been developed while visiting ETH.

§Distributed Computing Group, ETH Zentrum, IFW A 47.2, Haldeneggsteig 4, 8092 Zürich, Switzerland. E-mail: `wattenhofer@inf.ethz.ch`.

¶Institut für Theoretische Informatik, ETH Zentrum, CLW building, Clausiusstrasse 49, 8092 Zürich, Switzerland. E-mail: `widmayer@inf.ethz.ch`.

to $u_i(X, P_i, t_i)$, where t_i is another parameter called the *type* of agent i. The difficulty here is that the type t_i is *not known to the system* and is also part of the input required to construct the desired solution X^* (e.g., t_i represents the speed of a router and the system goal is to forward packets optimally). This piece of information is known to agent i which may report a different (false) type $r_i \neq t_i$ in order to improve her utility: on input r_i, the system computes a solution X' and provides a payment P' such that $u_i(X', P'_i, t_i) > u_i(X^*, P_i, t_i)$, where P_i is the payment that i would have received if reporting t_i. Notice that the computed solution X' is not the desired one since the input provided to the underlying algorithm is not the correct one. Therefore, one should design a suitable payment rule $p(\cdot)$ such that, for every possible t_i, agent i cannot improve her utility by reporting $r_i \neq t_i$. Moreover, this should hold also when some other agent j does not act rationally and reports $r_j \neq t_j$. The combination of an algorithm $\mathcal{A}$ and a payment rule $p(\cdot)$ that guarantees this property, for every agent, is called *truthful mechanism* with dominant strategies.

A central question in (algorithmic) mechanism design is the study of which system goals are achievable via truthful mechanisms, that is, if algorithm $\mathcal{A}$ is required to produce a certain output (e.g., an optimal solution for a combinatorial optimization problem) does a payment function $p(\cdot)$ exist such that the resulting mechanism $(\mathcal{A}, p)$ is truthful?

A large body of the existing literature focuses on the class of problems in which the utilities are *quasi-linear*, that is, agent i's utility factors into $u_i(X, P_i, t_i) = v_i(X, t_i) + P_i$, where $v_i(X, t_i)$ represents the valuations of agent i of solution X. For such problems, the celebrated Vickrey-Clarke-Groves (VCG) mechanisms [Vic61, Cla71, Gro73] guarantee the truthfulness under the hypothesis that the algorithm $\mathcal{A}$ maximizes the function $\sum_i v_i(X, t_i)$. VCG mechanisms have been successfully applied to a multitude of optimization problems involving selfish agents with applications to networking [NR00, FPS01, ACP+03, AE03, PV03] and electronic commerce [McM95, Cra97]. All these works assume that the problem is *utilitarian*, that is, the utility functions are quasi-linear and the objective maximization function can be written as the sum above. Moreover, even though Nisan and Ronen [NR99] first focused on problems whose objective function is not utilitarian, their n-approximation mechanism is nothing but a VCG mechanism for a related utilitarian problem. Actually, VCG mechanisms remain the only general technique to design truthful mechanisms (see Sect. 1 for a discussion on previous related work).

Unfortunately, there are problems for which (i) the objective function is not the sum of the agents' valuations and/or (ii) the utility function is not quasi-linear. Consider the following basic problem (see Sect. 3.2.1 for a more detailed description). In a communication network, each link e can successfully transmit a message with probability $q_e \in (0, 1)$. We want to select a *most reliable path*, i.e., a path between two given nodes which maximizes the probability that none of its links fails. Links are owned by selfish agents which are asked to report a (possibly uncorrect) probability $q'_e \in (0, 1)$. We provide to a chosen link e a payment specified by a function $p_e(\cdot)$ if and only if link e performed the transmission correctly. Each agent tries to maximizes the *expected* amount of money received.[1] Hence, *both* the objective and the utility functions can be expressed by means of the common "operator" '$\cdot$'. The Most Reliable Path (MRP) problem just described can be easily reduced to a utilitarian problem by considering the logarithms of both the optimization function and of the utility functions, thus implying the existence of a truthful mechanism. It is then natural to ask whether this is just "good chance", or this problem (and others) has some "similarities" with the class of utilitarian problems.

In this paper we address this question by defining a class of problems, termed *consistent* problems (see Sect. 2), which admit truthful mechanisms. The main advantages of our approach are that:

1. It provides an answer to the following question: which mathematical properties guarantee the existence of truthful mechanisms?

   Moreover, for a given problem, it is easy to see whether it satisfies these properties (while reducing the problem to a utilitarian one may not be as simple as for the MRP).

2. It provides a more intuitive interpretation of the payments, e.g., for problems like the MRP described above and the Arbitrage problem (see Sect. 3.2.1).

[1] We assume that the costs for transmitting are negligible, say equal 0.

We define *VCG-consistent* mechanisms as a natural extension of the VCG mechanisms and show that they are truthful for consistent problems. We then consider possible extensions of our result and provide both positive and negative answers depending on which property we add/drop from the definition of consistent problems. In particular, we identify four classes of problems:

$\mathsf{C}^{\mathsf{vcgc}}_{\mathsf{only}}$**.** This class is a natural restriction of consistent problems. In particular, let Π be a problem in $\mathsf{C}^{\mathsf{vcgc}}_{\mathsf{only}}$. Then, every truthful mechanism for Π is a VCG-consistent mechanism (Theorem 3.2). Hence, VCG-consistent mechanisms characterize the set of truthful mechanisms for problems in this class.

$\mathsf{C}^{\mathsf{vcgc}}_{\mathsf{vp}}$**.** This is a subclass of consistent problems. We prove that every problem $\Pi \in \mathsf{C}^{\mathsf{vcgc}}_{\mathsf{vp}}$ admits a truthful VCG-consistent mechanism which also satisfies the *voluntary participation* condition (Theorem 3.3).

$\mathsf{C}^{\mathsf{vcgc}}_{\mathsf{none}}$**.** Let Π be a problem in $\mathsf{C}^{\mathsf{vcgc}}_{\mathsf{none}}$. Then, every VCG-consistent mechanism for Π is not truthful (Theorem 4.1). In other words, VCG-consistent mechanisms always fail for problems in this class.

$\mathsf{C}_{\mathsf{none}}$**.** No problem $\Pi \in \mathsf{C}_{\mathsf{none}}$ admits a truthful mechanism (Theorem 4.6). This is a subclass of $\mathsf{C}^{\mathsf{vcgc}}_{\mathsf{none}}$. Clearly, for problems in $\mathsf{C}_{\mathsf{none}}$, our technique does not work. This, however, is not due to its weakness, but to a general impossibility result.

Interestingly, both classes $\mathsf{C}^{\mathsf{vcgc}}_{\mathsf{none}}$ and $\mathsf{C}_{\mathsf{none}}$ concern problems in which the set of feasible solutions *depends* on the private part of the input, thus not satisfying one of the constraints of the definition of consistent problem (see Constraint (1) in Def. 3.1). We then show that the 2nd Shortest Path problem (see Sect. 4.1.2) is in $\mathsf{C}_{\mathsf{none}}$. This implies that this class in non-empty and therefore our assumption on the set of feasible solutions is necessary (indeed, removing this assumption would give a *superclass* of $\mathsf{C}_{\mathsf{none}}$).

Other problems to which we apply our results are:

α-Rent Task Scheduling (see Sect. 3.2.2) This is a variant of the Task Scheduling problem considered in [NR99] obtained by modifying the (quasi-linear) utility functions. The resulting problem is consistent, though straightforward reductions to a utilitarian problem do not seem to exist. This shows that the non-existence of an exact mechanism in [NR99] is due to the "combination" of quasi-linear utilities with a non additive objective function (i.e., the makespan). Finally, the problem does not admit a truthful mechanism satisfying voluntary participation, thus implying that $\mathsf{C}^{\mathsf{vcgc}}_{\mathsf{vp}}$ is a proper subclass of consistent problems.

Knapsack (see Sect. 4.1.1) We consider three variants of this problem depending on which part of the input is held by the agents (namely, the item profits, the item sizes, or both). The corresponding versions belong to $\mathsf{C}^{\mathsf{vcgc}}_{\mathsf{vp}}$, $\mathsf{C}^{\mathsf{vcgc}}_{\mathsf{none}}$ and $\mathsf{C}_{\mathsf{none}}$, respectively. This basic problem has applications to scheduling, resource allocation and to a problem of web advertising [DG03].

Further related work Green and Laffont [GL77] showed that for certain utilitarian problems VCG mechanisms are the only truthful mechanisms. Nisan and Ronen [NR00] considered the approximability of NP-hard optimization problems via *VCG-based* mechanisms: these mechanisms are obtained from VCG ones by replacing an optimal algorithm $\mathcal{A}$ with a (polynomial-time) non-optimal one $\mathcal{A}'$. Archer and Tardos [AT01] considered so called *one-parameter* agents: here the valuation functions factor as $v_i(X, t_i) = w_i(X) \cdot t_i$. The authors provided a technique which allows to obtain truthful mechanisms (A, p) whenever A satisfies a "monotonicity" property. To the best of our knowledge this is the only technique other than the VCG one. All above mentioned results apply to the case of quasi-linear utility functions only.

Organization of the paper. We present some basic definitions and notation in Sect. 2. In Sect. 3 we provide the definition of consistent problem, VCG-consistent mechanisms and prove our main positive results. Sect. 3.1 deals with the voluntary participation condition, while Sect. 3.2 contains some applications of our positive results. Finally, we prove the negative results in Sect. 4 where we also apply these results to some of the above mentioned problems. Conclusions and open problems are in Sect. 5.

Due to lack of space some details concerning the problems formulation and some proofs are contained in Appendix A and in Appendix B, respectively.

2 Preliminaries

Informally, in a mechanism design problem one can imagine that the input $\mathcal{I} = (I_P, I)$ is split into a public and into a private part held by k agents. Public valuation and utility functions express the agents' preferences and how each agent "responds" to incentives.

We next provide a formal setting. Without loss of generality, we present the definition for maximization problems.

Given any vector $I = \langle y_1, \ldots, y_k \rangle \in \Theta_1 \times \ldots \times \Theta_k$, let $I_{-i} = \langle y_1, \ldots, y_{i-1}, y_{i+1}, \ldots, y_k \rangle$ and $\langle I_{-i}, x_i \rangle = \langle y_1, \ldots, y_{i-1}, x_i, y_{i+1}, \ldots, y_k \rangle$. Moreover, if $\mathcal{I} = (I_P, I)$, we let $\langle \mathcal{I}_{-i}, x_i \rangle = (I_P, \langle I_{-i}, x_i \rangle)$.

Definition 2.1 *A* Mechanism Design Maximization (MDMax) *problem is specified as follows:*

- Private instance. *Each agent a_i has available a* private input type $t_i \in \Theta_i$, *where Θ_i denotes the type space of agent a_i which is public knowledge. Given the part of the instance I_P which is* public knowledge, *$\mathcal{I}_T = (I_P, I_T)$ is the* private (or true) instance *specified by the true agents' types $I_T = \langle t_1, \ldots, t_k \rangle$.*
- Reported instance. *Each agent a_i makes public a reported type $r_i \in \Theta_i$; then, for $I_R = \langle r_1, \ldots, r_k \rangle$, the* reported instance *$\mathcal{I}_R = (I_P, I_R)$ is the input provided to the algorithm.*

  In the following, we will often write $\mathcal{I} = (I_P, I)$, for a vector $I = (y_1, \ldots, y_k) \in \Theta_1 \times \ldots \times \Theta_k$ to denote any possible input of the algorithm (i.e., any "reportable" instance) as opposed to $\mathcal{I}_T$ and $\mathcal{I}_R$ representing the specific private and reported instances, respectively.
- Feasible solutions. *Given any instance $\mathcal{I} = (I_P, I)$, $\Phi(\mathcal{I})$ denotes the set of feasible solutions, and $\overline{\Phi}(I_P) = \bigcup_{I' \in \Theta_1 \times \ldots \times \Theta_k} \Phi(I_P, I')$. The set of feasible solutions does not depend on the private part of the input, i.e.,*
$$\forall I_P \quad \forall I \in \Theta_1 \times \ldots \times \Theta_k, \quad \Phi(I_P, I) = \overline{\Phi}(I_P). \tag{1}$$
- Objective function. *A function $\mu(X, \mathcal{I})$ expresses the measure of a solution X, given any instance $\mathcal{I}$.*
- Valuation functions. *For every agent a_i, a function $v_i(X, t_i)$ expresses the* valuation *of a_i of a solution X, given any value $t_i \in \Theta_i$. The function $v_i(\cdot, \cdot)$ is public knowledge, while one of its arguments is not (namely, the type t_i).*

  We say that a solution X does not involve agent a_i *if $v_i(X, y_i) = v_i^0$, for a fixed value v_i^0 and for every $y_i \in \Theta_i$. We assume that v_i^0 is public knowledge and that, for every X, it is possible to decide whether X does not involve a_i.*
- Agent payments and utility functions. *For every agent a_i it is possible to define a* payment function $p_i(\cdot)$, *representing some sort of incentive for agent a_i. Then, a function $u_i(X, t_i, P_i)$ expresses the* utility *of a_i of a solution X, given its (true) type t_i and given $p_i(\cdot) = P_i$ (this value represents how much a_i benefits if a solution X is output and a_i receives a payment[2] equal to P_i). This function depends only on the values $v_i(X, t_i)$ and P_i, and represents what agent a_i tries to maximize.*

  We use the symbol P^0 to denote the fact that a_i receives no payment. In this case, for every X, we have that $u_i(X, t_i, P^0) = v_i(X, t_i)$.
- Goal. *Find an optimal solution for the true instance, that is, a solution $X^* \in \Phi(\mathcal{I}_T)$ such that*
$$\mu(X^*, \mathcal{I}_T) = \max\{\mu(X, \mathcal{I}_T) \mid X \in \Phi(\mathcal{I}_T)\}. \tag{2}$$

[2]The term 'payment' does not necessarily mean money as it actually denotes any form of incentive.

Observe that, because of Constraint (1), it is always possible to obtain a feasible solution. However, our goal is to find an optimal one, which *depends* on the agents' types (i.e., the true instance). In order to solve a MDMax problem we need a suitable combination of a payment scheme and an algorithm which guarantees that (i) no agent has an incentive in misreporting her type and (ii) the algorithm, once provided with the true instance $\mathcal{I}_T$, returns an optimal solution for that. In particular, the usual underlying assumption in mechanism design is that an agent misreports her type only in the case this might improve her utility (see e.g. [OR94]).

Definition 2.2 (truthful mechanism) *A* mechanism *for a* MDMax *problem is a pair* $\mathcal{M} = (\mathcal{A}, \mathcal{P})$, *where* $\mathcal{A}$ *is an algorithm computing a solution* $\mathcal{A}(\mathcal{I}_R)$ *and* $\mathcal{P}(\mathcal{I}_R) = \langle p_1(\mathcal{I}_R), \ldots, p_k(\mathcal{I}_R)\rangle$ *is the payment scheme. A mechanism* $\mathcal{M} = (\mathcal{A}, \mathcal{P})$ *for a* MDMax *problem is* truthful *if*

$$\forall a_i \quad \forall I_{-i} \quad \forall r_i \neq t_i \quad u_i(\mathcal{A}\langle \mathcal{I}_{-i}, t_i\rangle, t_i, p_i\langle \mathcal{I}_{-i}, t_i\rangle) \geq u_i(\mathcal{A}\langle \mathcal{I}_{-i}, r_i\rangle, t_i, p_i\langle \mathcal{I}_{-i}, r_i\rangle).$$

Observe that truthful mechanisms guarantee that, for every a_i, reporting $r_i = t_i$ is the best strategy even when some other agents misreport their type (i.e., $I_{-i} \neq \langle t_1, \ldots, t_{i-1}, t_{i+1}, \ldots, t_k\rangle$).

Another relevant feature of a mechanism is that of guaranteeing that a truthfully behaving agent a_i incurs in a utility which is not worse than the utility she would obtain if not "participating in the game", that is, if a solution X not involving a_i is computed and a_i receives no payment (see also the discussion in Sect. 3.1):

Definition 2.3 (voluntary participation) *A mechanism* $\mathcal{M} = (\mathcal{A}, \mathcal{P})$ *for a* MDMax *problem satisfies the* voluntary participation condition (VP) *if*

$$\forall a_i \quad \forall I_{-i} \quad u_i(\mathcal{A}\langle \mathcal{I}_{-i}, t_i\rangle, t_i, p_i\langle \mathcal{I}_{-i}, t_i\rangle) \geq v_i^0.$$

Given an instance $\mathcal{I}$, for the sake of simplicity, we denote by $\mathcal{I}_{-i}$ the instance $\langle \mathcal{I}_{-i}, \perp\rangle$, where $\perp \notin \Theta_i$ is a "dummy" value which makes unfeasible every feasible solution involving agent a_i.

In the rest of the paper we consider *optimal mechanisms*, that is, mechanisms $\mathcal{M} = (\mathcal{A}, \mathcal{P})$ that use an algorithm $\mathcal{A}$ computing an optimal solution w.r.t. the *reported instance*. A truthful optimal mechanism provides a solution for a MDMax problem: the truthfulness guarantees that the agents, being rational, report their types t_i and then algorithm $\mathcal{A}$ computes a solution $X^* = \mathcal{A}(\mathcal{I}_T)$ satisfying Eq. (2).

3 Truthful mechanisms for consistent problems

In this section we first introduce the class of *consistent problems* (Def. 3.1) and a family of mechanisms for this class which we call *VCG-consistent mechanisms* (Def. 3.2). We show that VCG-consistent mechanisms are truthful for consistent problems (Theorem 3.1) and prove that, under some natural assumptions, VCG-consistent mechanisms are the only truthful mechanisms for consistent problems (Theorem 3.2).

Definition 3.1 (consistent problem) *A* MDMax *problem is* consistent *if*

- μ *is a consistent objective function, i.e., for any instance* $\mathcal{I} = (I_P, I)$, *with* $I = \langle y_1, \ldots, y_k\rangle$, *and for any* $X \in \Phi(\mathcal{I})$, *it holds that* $\mu(X, \mathcal{I}) = \bigoplus_i v_i(X, y_i)$, *where '*$\oplus$*' is a suitable operator which enjoys the following properties: associativity, commutativity and monotonicity in its arguments;*
- *the utility function is such that*

$$\forall a_i \quad \forall X \in \overline{\Phi}(I_P) \quad \forall P_i \quad u_i(X, t_i, P_i) = v_i(X, t_i) \oplus P_i. \tag{3}$$

The class of all consistent problems is denoted as consistent.

Definition 3.2 (VCG-consistent mechanisms) *A (optimal) mechanism* $(\mathcal{A}, \mathcal{P})$ *for a consistent problem is a* VCG-consistent mechanism *if, for all* i*, there exists a function* $h_i(\mathcal{I}_{-i})$ *such that*

$$p_i(\mathcal{I}) = \mu_{-i}(\mathcal{A}(\mathcal{I}), \mathcal{I}) \oplus h_i(\mathcal{I}_{-i}), \tag{4}$$

where $\mu_{-i}(X, \mathcal{I}) = \bigoplus_{j \neq i} v_j(X, y_j)$*.*

The following theorem generalizes the (proof of the) analogous result in [Gro73] about the truthfulness of VCG mechanisms for utilitarian problems (i.e., the case '$\oplus$'='$+$'). Noticeably, the proof exploits Constraint (1) (see Appendix B.1).

Theorem 3.1 *A VCG-consistent mechanism for a consistent problem is truthful.*

We next show that, under some natural assumptions, VCG-consistent mechanisms are the only truthful mechanisms for consistent problems.

Definition 3.3 (the class $\mathsf{C}^{\mathsf{vcgc}}_{\mathsf{only}}$.) *A consistent problem* Π *belongs to* $\mathsf{C}^{\mathsf{vcgc}}_{\mathsf{only}}$ *if its operator enjoys the following properties: identity element* $i_\oplus$*, inverse and strict monotonicity,*[3] *and the type spaces are complete, (i.e.,* $\forall \mathcal{I}$*,* $\forall i, \{v_i(\cdot, y_i) \mid y_i \in \Theta_i\} = \{f : \Phi(\mathcal{I}) \mapsto \mathbb{R}\}$*).*

The proof of the following theorem is a non-trivial adaptation of the proof of a similar result for (a subclass of) utilitarian problems [GL77]. However, our result is stronger since it shows that *every* consistent problem in $\mathsf{C}^{\mathsf{vcgc}}_{\mathsf{only}}$ has essentially a "unique" truthful mechanism: the VCG-consistent mechanism in Def. 3.2, where the only degree of freedom is on the definition of the function $h_i(\cdot)$.

Theorem 3.2 *Let* $(\mathcal{A}, \mathcal{P})$ *be a truthful mechanism for a problem* $\Pi \in \mathsf{C}^{\mathsf{vcgc}}_{\mathsf{only}}$*. Then,* $(\mathcal{A}, \mathcal{P})$ *is a VCG-consistent mechanism for* Π*.*

Proof. We will show that, for each i, there exists a function $h_i(\mathcal{I}_{-i})$ such that the payment function is of the form (4). Let us define $h_i(\mathcal{I}) = \mu_{-i}(\mathcal{A}(\mathcal{I}), \mathcal{I})^{-1} \oplus p_i(\mathcal{I})$, thus implying that

$$p_i(\mathcal{I}) = \mu_{-i}(\mathcal{A}(\mathcal{I}), \mathcal{I}) \oplus h_i(\mathcal{I}).$$

In order to show that $h_i(\cdot)$ is independent of $y_i \in \Theta_i$ we assume, by contradiction, that there exist i, $\mathcal{I}$, $\widehat{y}_i$ such that $h_i(\mathcal{I}) \neq h_i(\langle \mathcal{I}_{-i}, \widehat{y}_i \rangle)$. For simplicity, let $\widehat{\mathcal{I}} = \langle \mathcal{I}_{-i}, \widehat{y}_i \rangle$, $X = \mathcal{A}(\mathcal{I})$ and $\widehat{X} = \mathcal{A}(\widehat{\mathcal{I}})$. We define ε as follows:

$$\varepsilon = \begin{cases} h_i(\mathcal{I}) \oplus h_i(\widehat{\mathcal{I}})^{-1} & \text{if } h_i(\mathcal{I}) < h_i(\widehat{\mathcal{I}}), \\ h_i(\mathcal{I})^{-1} \oplus h_i(\widehat{\mathcal{I}}) & \text{otherwise.} \end{cases} \tag{5}$$

From the strict monotonicity of '$\oplus$', there exists a δ such that $\varepsilon < \varepsilon \oplus \delta < i_\oplus$. For any such δ, we can consider the following function f:

$$f(X) = \begin{cases} \mu_{-i}(X, \mathcal{I})^{-1} & \text{if } X = \widehat{X}, \\ \mu_{-i}(X, \mathcal{I})^{-1} \oplus (\varepsilon \oplus \delta) & \text{otherwise.} \end{cases} \tag{6}$$

Because of the completeness of type spaces, there exists $\widetilde{y}_i$ such that, for every $X \in \Phi(\mathcal{I})$, $f(X) = v_i(X, \widetilde{y}_i)$. Let $\widetilde{\mathcal{I}} = \langle \mathcal{I}_{-i}, \widetilde{y}_i \rangle$ and $\widetilde{X} = \mathcal{A}(\widetilde{\mathcal{I}})$. By contradiction, assume that $\widetilde{X} \neq \widehat{X}$, thus implying

$$\mu(\widetilde{X}, \widetilde{\mathcal{I}}) = \mu_{-i}(\widetilde{X}, \widetilde{\mathcal{I}}) \oplus v_i(\widetilde{X}, \widetilde{y}_i) = \mu_{-i}(\widetilde{X}, \widetilde{\mathcal{I}}) \oplus f(\widetilde{X}) = (\varepsilon \oplus \delta) < i_\oplus. \tag{7}$$

Where the last equality follows from Eq. 6. Similarly, we obtain $\mu(\widehat{X}, \widetilde{\mathcal{I}}) = i_\oplus$. This and Eq. 7 contradict the fact that $\mu(\widetilde{X}, \widetilde{\mathcal{I}}) \geq \mu(\widehat{X}, \widetilde{\mathcal{I}})$. Hence, it must be $\widetilde{X} = \widehat{X}$.

It is easy to show that, from the truthfulness of $(\mathcal{A}, \mathcal{P})$, it must hold that

[3] The inverse of x is denoted by x^{-1} and satisfies $x \oplus x^{-1} = i_\oplus$. We say that an operator $\oplus$ satisfies *strict monotonicity* if for every a,a' and b, with $a < a'$, it holds that $a \oplus b < a' \oplus b$.

$$\widehat{X} = \widetilde{X} \quad \Rightarrow \quad p_i(\mathcal{I}) = p_i(\widetilde{\mathcal{I}}), \tag{8}$$
$$h_i(\mathcal{I}) \neq h_i(\widehat{\mathcal{I}}) \quad \Rightarrow \quad X \neq \widehat{X}. \tag{9}$$

In the case $h_i(\mathcal{I}) > h_i(\widehat{\mathcal{I}})$ (the opposite is similar) it holds that

$$u_i(\widetilde{X}, \widetilde{y}_i, p_i(\widetilde{\mathcal{I}})) = v_i(\widetilde{X}, \widetilde{y}_i) \oplus h_i(\widetilde{\mathcal{I}}) \oplus \mu_{-i}(\widetilde{X}, \widetilde{\mathcal{I}}) = h_i(\widetilde{\mathcal{I}}) \oplus \mu(\widetilde{X}, \widetilde{\mathcal{I}}) = h_i(\widehat{\mathcal{I}}) \oplus i_{\oplus} = h_i(\widehat{\mathcal{I}}), \tag{10}$$
$$u_i(\widehat{X}, \widetilde{y}_i, p_i(\widehat{\mathcal{I}})) = v_i(X, \widetilde{y}_i) \oplus h_i(\mathcal{I}) \oplus \mu_{-i}(X, \mathcal{I}) = h_i(\mathcal{I}) \oplus \mu(X, \widetilde{\mathcal{I}}) = h_i(\mathcal{I}) \oplus (\varepsilon + \delta). \tag{11}$$

Since $h_i(\mathcal{I}) \oplus (\varepsilon \oplus \delta) > h_i(\mathcal{I}) \oplus \varepsilon = h_i(\widehat{\mathcal{I}})$, the two equations above contradict the truthfulness of the mechanism. □

3.1 The voluntary participation condition

In practical applications, agents have the freedom/right to put themselves out of the "game" if the final mechanism outcome (i.e., the utility) turns out to be disadvantageous for them.

For example, consider the case in which the valuation $v_i(X, t_i)$ represents a cost required to a_i in order to implement the solution X and $p_i(\mathcal{I}_R)$ is the amount of money that a_i receives for that. Agent a_i has the freedom to refuse the payments and to not implement the solution, if the utility deriving from $v_i(X, t_i)$ and $p_i(\mathcal{I}_R)$ is less than 0 (i.e., the utility in case agent a_i does not perform any work nor receives money).

Definition 3.4 (the class $\mathsf{C}_{\mathsf{vp}}^{\mathsf{vcgc}}$.) *A consistent problem* Π *belongs to* $\mathsf{C}_{\mathsf{vp}}^{\mathsf{vcgc}}$ *if the operator enjoys the following properties: identity element, inverse and strict monotonicity, and*

$$\forall a_i \quad \emptyset \neq \Phi(\mathcal{I}_{-i}) \subseteq \Phi(\mathcal{I}). \tag{12}$$

The following theorem gives a sufficient condition for the existence of VCG-consistent mechanisms which satisfies VP (see Def. 2.3).

Theorem 3.3 *Let* Π *be a consistent problem in* $\mathsf{C}_{\mathsf{vp}}^{\mathsf{vcgc}}$ *and* $(\mathcal{A}, \mathcal{P})$ *be the VCG-consistent mechanism for* Π *with* $h_i(\mathcal{I}_{-i}) = \mu_{-i}(\mathcal{A}(\mathcal{I}_{-i}), \mathcal{I}_{-i})^{-1}$. *Then* $(\mathcal{A}, \mathcal{P})$ *satisfies VP.*

Proof. Consider $X = \mathcal{A}\langle \mathcal{I}_{-i}, t_i \rangle$ and $P_i = p_i\langle \mathcal{I}_{-i}, t_i \rangle$, for any $\mathcal{I}_{-i}$. Since $\mathcal{A}(\mathcal{I}_{-i}) \in \Phi(\mathcal{I}_{-i})$, it holds that $\mu(\mathcal{A}(\mathcal{I}_{-i}), \langle \mathcal{I}_{-i}, t_i \rangle) = \mu_{-i}(\mathcal{A}(\mathcal{I}_{-i}), \mathcal{I}_{-i}) \oplus v_i^0$. By Def. 3.1, $u_i(X, t_i, P_i) = v_i(X, t_i) \oplus p_i\langle \mathcal{I}_{-i}, t_i \rangle = v_i(X, t_i) \oplus \mu_{-i}(X, \langle \mathcal{I}_{-i}, t_i \rangle) \oplus h_i(\mathcal{I}_{-i})$. By associativity, monotonicity and existence of the inverse, we have that:

$$\begin{aligned} u_i(X, t_i, P_i) &= \mu(X, \langle \mathcal{I}_{-i}, t_i \rangle) \oplus \left(\mu_{-i}(\mathcal{A}(\mathcal{I}_{-i}), \mathcal{I}_{-i})^{-1} \oplus (v_i^0)^{-1} \oplus v_i^0\right) \\ &= \mu(X, \langle \mathcal{I}_{-i}, t_i \rangle) \oplus \mu(\mathcal{A}(\mathcal{I}_{-i}), \langle \mathcal{I}_{-i}, t_i \rangle)^{-1} \oplus v_i^0. \end{aligned} \tag{13}$$

From Condition 12 and from the optimality of $\mathcal{A}$, if follows that $\mu(X, \langle \mathcal{I}_{-i}, t_i \rangle) \geq \mu(\mathcal{A}(\mathcal{I}_{-i}), \langle \mathcal{I}_{-i}, t_i \rangle)$. From the monotonicity of '$\oplus$', we obtain $\mu(X, \langle \mathcal{I}_{-i}, t_i \rangle) \oplus \mu(\mathcal{A}(\mathcal{I}_{-i}), \langle \mathcal{I}_{-i}, t_i \rangle)^{-1} \geq i_{\oplus}$. This, Eq. 13 and the monotonicity of '$\oplus$' yield $u_i(X, t_i, P_i) \geq v_i^0$. Hence the theorem follows. □

3.2 Applications to non-utilitarian problems

We now provide three examples of non-utilitarian consistent problems whose operator is '$\oplus$'='$\cdot$' (the MRP and the Arbitrage problems in Sect. 3.2.1) and '$\oplus$'='min' (the α-Rent Task Scheduling problem in Sect. 3.2.2).

3.2.1 Two multiplicative problems

Before introducing the MRP and the ARBITRAGE problems, let us consider a general framework in which a truthful mechanism has to be designed on a directed weighted graph $G = (V, E, w)$ that has an edge weight $w_e \in \Theta$ associated with each edge $e \in E$. We are given $s, t \in V$, called the source and the destination, respectively The goal is to find a path from s to t which maximizes the product of the edge weights. Each edge e is owned by a distinct selfish agent a_e [4] which knows the weight $w_e \in \Theta$ (i.e., her type). In the following, we will refer to this problem as a LONGEST MULTIPLICATIVE PATH problem (LMP[Θ]).

The LMP[Θ] problem can be formalized as a consistent problem whenever the valuation functions $v_e(\cdot)$ and the utility functions $u_e(\cdot)$ satisfy

$$v_e(\pi, y_e) = \begin{cases} v_i^0 = 1 & \text{if } e \text{ is not on the path } \pi, \\ y_e & \text{otherwise,} \end{cases} \tag{14}$$

and $u_e(\pi, w_e, P_e) = v_e(\pi, w_e) \cdot P_e$.

Since the set of feasible solutions depends ont the topology of the graph only, for 2-connected graphs,[5] Constraint 12 is met. Moreover, for every $\Theta \subseteq \mathbb{R}^+$, the standard product operator is strictly monotone, thus implying that LMP[Θ]$\in \mathsf{C}_{\mathsf{vp}}^{\mathsf{vcgc}}$. Hence, by Theorem 3.3 we obtain the following:

Corollary 3.1 *For every $\Theta \subseteq \mathbb{R}^+$, there exists a truthful mechanism $(\mathcal{A}, p)$ for* LMP[Θ] *which, for 2-connected graphs, also meets VP. In this case, for every $e \in E$,*

$$p_e(\mathcal{I}) = \frac{\mu_{-e}(\mathcal{A}(\mathcal{I}), \mathcal{I})}{\mu_{-e}(\mathcal{A}(\mathcal{I}_{-e}), \mathcal{I}_{-e})} \tag{15}$$

and

$$u_e(X, t_e, p_e(\mathcal{I}_T)) = \frac{\mu(X, \mathcal{I}_T)}{\mu_{-e}(X_{-e}, \mathcal{I}_{T-e})}, \tag{16}$$

where $X = \mathcal{A}(\mathcal{I}_T)$ and $X_{-e} = \mathcal{A}(\mathcal{I}_{T-e})$.

In the following we apply the above result to the MOST RELIABLE PATH and to the ARBITRAGE problems.

The MOST RELIABLE PATH (MRP) **problem.** Consider the MRP problem discussed in Sect. 1. In particular, the message is forwarded from one node to the next one until either (i) the message reaches the destination t or (ii) the link fails. In the latter case, the transmission is lost and a "dummy" message is forwarded throughout the selected path in place of the original one.

In order to satisfy Eq. (14), we use the following rule for the agents' payment. If edge e is not on the chosen path, then the corresponding agent receives a payment equal to $P_e = 1$. Moreover, an agent in the selected path is rewarded after (and only if) her link has *successfully* forwarded the message. Hence, the *true* agent's expected utility is $q_e P_e$. It is easy to see that the MRP problem is the LMP[$(0, 1)$] problem. Corollary 3.1 implies the existence of a truthful mechanism $(\mathcal{A}, \mathcal{P})$ which, if at least two disjoint st-paths exists, also meets VP. In this case, Eq.s (15) and (16) yield the following intuitive interpretation of payments and of utilities, respectively:

$$p_e(\mathcal{I}_T) = \frac{Pr[\text{no link in } \pi \text{ fails} | \ e \text{ does not fail}]}{Pr[\text{no link in } \pi_{-e} \text{ fails}]}, \quad u_e(\pi, q_e, p_e(\mathcal{I}_T)) = \frac{Pr[\text{no link in } \pi \text{ fails}]}{Pr[\text{no link in } \pi_{-e} \text{ fails}]},$$

where π is the best st-path and π_{-e} denotes the best st-path not containing e.

[4] The existence of truthful mechanisms easily extends to a more general setting where each agent owns multiple edges.

[5] If the graph is not 2-connected then the problem breaks down to independent subproblems (2-connected components). In this case, it is easy to see that the VP condition cannot be fulfilled.

The ARBITRAGE **problem.** Arbitrage is the trading of one currency for another with the hopes of gaining on the exchange by taking advantage of small differences in the exchange rates among several currencies. Given n currencies $c_1, \ldots, c_n$, there exists an "exchange relation" $\langle i,j \rangle$ with a currency exchange rate $c_{i,j}$ between currencies c_i and c_j if there is an agent/stock-broker $a_{i,j}$ who buys $c_{i,j}$ units of currency c_j per units of currency c_i. An operation in exchange of a currency s for another t should be made by selecting a sequence of currencies $\langle s = c_{i_1}, \ldots, c_{i_k} = t \rangle$ with maximum derived exchange rate $r_{i_1,i_2} \cdot \ldots \cdot r_{i_{k-1},i_k}$.

We can formalize the Arbitrage problem as a LMP$[\Theta]$ problem where $\Theta = \mathbb{R}^+$. Corollary 3.1 implies that the ARBITRAGE problem admits a truthful mechanism $(\mathcal{A}, \mathcal{P})$ which also satisfies VP.

Let π be a best st-path and $\pi_{-\langle i,j \rangle}$ be a best st-path not containing edge $\langle i,j \rangle$. The payment $p_{i,j}(\mathcal{I}_T)$ due to each agent $a_{i,j}$ is equal to the amount of units of currency c_i that a stock-broker could buy per units of currency c_j by converting: (i) one currency c_j to $\mu(\pi_{j,t}, \mathcal{I}_T)$ units of currency t; (ii) each unit of currency t to $\mu(\pi_{-\langle i,j \rangle}, (\mathcal{I}_T)_{-\langle i,j \rangle})^{-1}$ units of currency s and, finally, (iii) each unit of currency t to $\mu(\pi_{s,i}, \mathcal{I}_T)$ units of currency c_i. Indeed:

$$p_{i,j}(\mathcal{I}_T) = \frac{\mu(\pi_{s,i}, \mathcal{I}_T) \cdot \mu(\pi_{j,t}, \mathcal{I}_T)}{\mu(\pi_{-\langle i,j \rangle}, (\mathcal{I}_T)_{-\langle i,j \rangle})},$$

where $\pi_{s,i}$ and $\pi_{j,t}$ denote the sub-paths of π from s to i and from j to t, respectively.

The utility $u_{i,j}(\pi, c_{i,j}, p_{i,j}(\mathcal{I}_T)) = v_{i,j}(\pi, c_{i,j}) \cdot p_{i,j}(\mathcal{I}_T)$ of each agent $a_{i,j}$ is the amount of units of currency c_i which can be gained starting from one unit of the same currency, by converting it to $v_{i,j}(\pi, c_{i,j})$ units of currency c_j, and then by buying $p_{i,j}$ units of the initial currency per units of c_j.

Clearly, VP guarantees that each agent $a_{i,j}$ involved in the solution actually achieves a profit as she earns $u_{i,j}(\pi, c_{i,j}, p_{i,j}(\mathcal{I}_T)) \geq 1$ units of c_i per units of the same currency.

3.2.2 The α-RENT TASK SCHEDULING problem

We are given k tasks which need to be allocated to n machines, each of them corresponding to one agent. Let t^i_j denote the minimum amount of time machine i is capable of performing task j and let X_i be the the set of tasks allocated to agent a_i. The goal is to minimize the makespan, that is, the maximum, over all machines, completion time.

The type of agent i is given by $t_i = \langle t^i_1, \ldots, t^i_k \rangle$, thus implying $I_T = \langle t_1, \ldots, t_n \rangle$, $I_P = \langle k, n \rangle$ and $\mathcal{I}_T = (I_P, I_T)$. The set of feasible solutions $\Phi(\mathcal{I})$ is the set of all partitions $X = X_1, \ldots, X_n$ of the tasks, where X_i denotes the tasks allocated to agent a_i. For any $\mathcal{I}$, we define $v_i(X, t_i) = -\sum_{j \in X_i} t^i_j$, that is, the completion time of machine i. Agent a_i is not involved in the solution X if $X_i = \emptyset$. In this case, $v_i(X, \cdot) = 0 = v^0_i$.

We consider the following variant of the problem defined in [NR99]. An assignment has to be computed according to the reported types. Each machine i that has been selected (i.e., $X_i \neq \emptyset$) is *rented* for the duration required to perform the tasks assigned to it. The corresponding agent must then receive an amount of money *not larger* than $\alpha - \sum_{j \in X_i} t^i_j = \alpha + v_i(X, t_i)$, where α is a fixed constant equal for all machines.

Incentives are provided by defining, for each machine/agent, a *maximum* payment M_i that the machine i will receive if used. In particular, each rented machine is then payed the *minimum* between M_i and $\alpha + v_i(X, t_i)$.

The utility of an agent i is naturally defined as the amount of money derived from the renting of her machine, that is, $\min\{\alpha + v_i(X, t_i), M_i\}$. By letting $P_i := M_i - \alpha$, the previous quantity can be rewritten as

$$\min\{\alpha + v_i(X, t_i), M_i\} = \alpha + \min\{v_i(X, t_i), P_i\}.$$

To formalize the problem as a consistent problem with operator '$\oplus$'='min' it suffices to define $u_i(X, t_i, P_i) = \min(v_i(X, t_i), P_i)$, and to observe that $\mu(X, \mathcal{I}) = \max_{i=1}^n -v_i(X, y_i) = \min_{i=1}^n v_i(X, y_i)$. Theorem 3.1 thus implies the following:

Corollary 3.2 *The* α-RENT TASK SCHEDULING *problem is consistent. Hence, It admits a truthful mechanism.*

Proof. Notice that the operator under consideration is '$\oplus$'='min', which enjoys the following properties: associativity, commutativity, monotonicity in its arguments. It is also easy to see that the objective and the utility functions satisfy the definition of consistent problem (see Def. 3.1), thus implying that the α-Rent Task Scheduling problem admits a truthful VCG-consistent mechanism (see Theorem 3.1). □

Observe that, the only difference between the α-Rent Task Scheduling problem and the Task Scheduling problem in [NR99] is on the utility function. This provides an interesting comparison since in [NR99] the authors proved that no exact (or even 2-approximate non-polynomial-time) truthful mechanism exists. Corollary 3.2 shows that this is due to the fact that the utility functions are quasi-linear.

Remark 3.1 (on the voluntary participation) *Observe that no mechanism for the α-Rent Task Scheduling problem can guarantee the VP condition. Indeed, it suffices to consider instances for which $\min\{t^i_j\} > \alpha$, in which case the utilities are always negative. Hence, α-Rent Task Scheduling $\notin \mathsf{C}^{\mathsf{vcgc}}_{\mathsf{vp}}$.*

4 Impossibility results

In this section we investigate extensions of our positive result (Theorem 3.1) to problems obtained by removing Constraint (1) in the definition of consistent:

Definition 4.1 (relaxed consistent problem) *A problem is a* relaxed consistent problem *if it satisfies all constraints of Def. 2.1 except for Constraint (1), as well as the two items in Def. 3.1. The class of all relaxed consistent problems is denoted as* relaxed consistent.

In Sect.s 4.1 and 4.2 we define two subclasses of relaxed consistent and show that problems in these two classes do not admit truthful VCG-consistent mechanisms (Theorem 4.1) and truthful mechanisms (Theorem 4.6), respectively. We also prove that the latter class in included in the former (Theorem 4.5).

4.1 A class with no truthful VCG-consistent mechanisms

Intuitively speaking, we next consider a class of problems for which some non-feasible solution $\hat{X}$ has a measure strictly better than any feasible solution. Moreover, such an unfeasible solution can be output when reporting a false input $\hat{\mathcal{I}}$, that is, $\hat{X} = \mathcal{A}(\hat{I}) \in \Phi(\hat{\mathcal{I}})$. Formally, we have the following:

Definition 4.2 (the class $\mathsf{C}^{\mathsf{vcgc}}_{\mathsf{none}}$.) *A problem Π is said to be in the class $\mathsf{C}^{\mathsf{vcgc}}_{\mathsf{none}}$ if it is relaxed consistent and the following holds:*

1. *the operator '$\oplus$' satisfies* strict monotonicity*;*
2. *there exist i, $\widetilde{I} = \langle \widetilde{y}_1, \ldots, \widetilde{y}_k \rangle$ and $\widehat{y}_i \in \Theta_i$ ($y_i \neq \widehat{y}_i$) such that, for $\widetilde{\mathcal{I}} = (I_P, \widetilde{I})$ and $\widehat{\mathcal{I}} = \langle \mathcal{I}_{-i}, \widehat{y}_i \rangle$, it holds that*
$$\mathcal{A}(\widehat{\mathcal{I}}) \notin \Phi(\widetilde{\mathcal{I}}) \text{ and } \mu(\mathcal{A}(\widehat{\mathcal{I}}), \widetilde{\mathcal{I}}) > \mu(\mathcal{A}(\widetilde{\mathcal{I}}), \widetilde{\mathcal{I}}). \tag{17}$$

Theorem 4.1 *No problem $\Pi \in \mathsf{C}^{\mathsf{vcgc}}_{\mathsf{none}}$ admits a truthful VCG-consistent mechanism.*

Proof. By contradiction, let $(\mathcal{A}, \mathcal{P})$ be a VCG-consistent truthful mechanism for Π and be $\widetilde{X} = \mathcal{A}(\widetilde{\mathcal{I}})$ and $\widehat{X} = \mathcal{A}(\widehat{\mathcal{I}})$. We then have that

$$\begin{aligned}
u_i(\widetilde{X}, \widetilde{y}_i, p_i(\widetilde{\mathcal{I}})) = v_i(\widetilde{X}, \widetilde{y}_i) \oplus (\mu_{-i}(\widetilde{X}, \widetilde{\mathcal{I}}) \oplus h_i(\widetilde{\mathcal{I}}_{-i})) & \quad \text{(by Def.s 3.1 and 3.2)} \\
= \mu(\widetilde{X}, \widetilde{\mathcal{I}}) \oplus h_i(\widetilde{\mathcal{I}}_{-i}) & \quad \text{(by associativity of '}\oplus\text{' and by Def. 3.1)} \\
< \mu(\widehat{X}, \widetilde{\mathcal{I}}) \oplus h_i(\widetilde{\mathcal{I}}_{-i}) & \quad \text{(by Eq. 17 and strict monotonicity of '}\oplus\text{')} \\
= u_i(\widehat{X}, \widetilde{y}_i, p_i(\widehat{\mathcal{I}})) & \quad \text{(by Def.s 4.1 and 4.2)}
\end{aligned}$$

thus contradicting the truthfulness of $(\mathcal{A}, \mathcal{P})$. This completes the proof. □

In the following we provide two examples of problems in the class $\mathsf{C}^{\mathsf{vcgc}}_{\mathsf{none}}$ which, by Theorem 4.1, do not admit a truthful VCG-consistent mechanism: KNAPSACK and the 2ND SHORTEST PATH.

4.1.1 The KNAPSACK problem

We consider a variant of the classical optimization problem called 0-1 KNAPSACK which can be described as follows. We are given a set of n items $\{1, \ldots, n\}$, each one characterized by a *profit* π_i and a *size* σ_i. The goal is to find a set of items which maximize the total profit and such that its total occupancy does not exceed a given capacity B. Each item i is associated with an agent a_i that holds a part of the instance. Depending on how the private part of the instance is defined we distinguish the following three problems:

- KNAPSACK$[\pi]$, where each agent a_i only holds the profit $\pi_i = t_i$ associated with each item i, whereas every size σ_i is public knowledge.
- KNAPSACK$[\sigma]$, where each agent a_i only holds the size $\sigma_i = t_i$ associated with each item i, whereas every profit π_i is public knowledge.
- KNAPSACK$[\pi, \sigma]$ where each agent a_i holds both the profit π_i and the size σ_i associated with each item i, that is, $t_i = \langle \pi_i, \sigma_i \rangle$.

For every problem variant, we let $\Phi(\mathcal{I}) = \{X \in \{0,1\}^n \mid \sum_{i=1}^n X_i \sigma_i \leq B\}$. Moreover, the total profit of a solution $X \in \Phi(\mathcal{I})$ is given by $\mu(X, \mathcal{I}) = \sum_{i=1}^n X_i \pi_i$ is the total profit. Finally, $u_i(X, t_i, P_i) = P_i + v_i(X, t_i)$, where $v_i(X, y_i) = X_i \pi_i$.

It is worth noticing that only KNAPSACK$[\pi]$ meets Constraint (1), as sizes are public knowledge and $\Phi(\mathcal{I})$ is constant. Then, it is immediate to prove that:

Theorem 4.2 KNAPSACK$[\pi] \in \mathsf{C}^{\mathsf{vcgc}}_{\mathsf{vp}}$. *Hence, it admits a truthful mechanism which also meets VP.*

On the contrary, KNAPSACK$[\sigma]$ and KNAPSACK$[\pi, \sigma]$ satisfy Def. 3.1 except for Constraint (1). In these case we can state the following:

Theorem 4.3 KNAPSACK$[\sigma]$, KNAPSACK$[\pi, \sigma] \in \mathsf{C}^{\mathsf{vcgc}}_{\mathsf{none}}$. *Hence, they do not admit a truthful VCG-consistent mechanism.*

The three problem versions above have a natural application to the use of a shared communication channel of limited capacity and to a problem of "selling" part of a web page (typically, a marginal strip of fixed width/height) for putting some advertisements (see [DG03] for a description of the model).

4.1.2 The 2ND SHORTEST PATH problem

Let us consider an undirected weighted graph $G = (V, E, w)$ and two nodes $s, t \in V$. The objective is to find a path whose length is minimal among all st-paths that have no minimal length in G. More formally, for any instance $\mathcal{I} = G$, if Φ_{st} is the set of all st-paths in (V, E) and $X_1^*(\mathcal{I}) \subseteq \Phi_{st}$ is the subset of the shortest st-paths, $\Phi(\mathcal{I}) = \Pi_{st}(\mathcal{I}) \setminus X_1^*(\mathcal{I})$. Similarly to the SHORTEST PATH problem mentioned in [NR99], the valuation function of the agent owing edge e is equal to

$$v_e(\pi, \mathcal{I}_R) = \begin{cases} -r_e & \text{if } e \in \pi, \\ 0 & \text{otherwise.} \end{cases}$$

Utilities are quasi-linear and the objective function is the total weight of the path, that is, $\sum_{e \in \pi} r_e$. By letting $\mu(\pi, \mathcal{I}_R) = \sum_{e \in \pi} -r_e$, and by observing that $\mu(\pi, \mathcal{I}_R) = \sum_{e \in \pi} v_e(\pi, \mathcal{I}_R)$, we can easily prove the following result:

Theorem 4.4 *The* 2ND SHORTEST PATH *problem is in* $\mathsf{C}^{\mathsf{vcgc}}_{\mathsf{none}}$. *Hence, it does not admit a truthful VCG-consistent mechanism.*

In the next section we will strengthen the results of Theorem 4.3 and of Theorem 4.4.

4.2 A class with no truthful mechanisms

We next provide a general technique to prove the non-existence of truthful mechanisms for a given problem. We will then apply this result to the KNAPSACK$[\pi, \sigma]$ and to the 2ND SHORTEST PATH problems and show that the reason why VCG-consistent mechanisms fail is not due to its weakness.

Definition 4.3 (the class $\mathsf{C}_{\mathsf{none}}$.) *A problem Π is said to be in the class $\mathsf{C}_{\mathsf{none}}$ if it relaxed consistent and the following holds:*

1. *the operator '$\oplus$' satisfies strict monotonicity;*
2. *there exist i, ν, $\widetilde{I} = \langle \widetilde{y}_1, \ldots, \widetilde{y}_k \rangle$ and $\widehat{y}_i \in \Theta_i$ ($y_i \neq \widehat{y}_i$) such that, for $\widetilde{\mathcal{I}} = (I_P, \widetilde{I})$ and $\widehat{\mathcal{I}} = \langle \widetilde{\mathcal{I}}_{-i}, \widehat{y}_i \rangle$, it holds that*

$$\mathcal{A}(\widehat{\mathcal{I}}) \notin \Phi(\widetilde{\mathcal{I}}) \ \wedge \ v_i(\mathcal{A}(\widehat{\mathcal{I}}), \widetilde{y}_i) > v_i(\mathcal{A}(\widehat{\mathcal{I}}), \widehat{y}_i) \ \wedge \ v_i(\mathcal{A}(\widehat{\mathcal{I}}), \cdot) \neq \nu \ \wedge \ v_i(\mathcal{A}(\widetilde{\mathcal{I}}), \cdot) = \nu. \quad (18)$$

Theorem 4.5 *The class $\mathsf{C}_{\mathsf{none}}$ is included in $\mathsf{C}^{\mathsf{vcgc}}_{\mathsf{none}}$.*

Proof. Let $\Pi \in \mathsf{C}_{\mathsf{none}}$. By Def. 4.3 there must exist i, $\widetilde{I} = \langle \widetilde{y}_1, \ldots, \widetilde{y}_k \rangle$ and $\widehat{y}_i \neq \widetilde{y}_i$ such that Eq. 18 is satisfied, for $\widetilde{\mathcal{I}} = (I_P, \widetilde{I})$ and $\widehat{\mathcal{I}} = \langle \widetilde{\mathcal{I}}_{-i}, \widehat{y}_i \rangle$. Let $\widetilde{X} = \mathcal{A}(\widetilde{\mathcal{I}})$ and $\widehat{X} = \mathcal{A}(\widehat{\mathcal{I}})$.

In order to prove that $\Pi \in \mathsf{C}^{\mathsf{vcgc}}_{\mathsf{none}}$ let us first suppose that $\mu(\widehat{X}, \widehat{\mathcal{I}}) \geq \mu(\widetilde{X}, \widehat{\mathcal{I}})$. In such a case the claim directly follows by the following three inequalities which show that Eq. 17 is also satisfied:

$$\begin{array}{ll} \mu(\widehat{X}, \widetilde{\mathcal{I}}) > \mu(\widehat{X}, \widehat{\mathcal{I}}) & \text{(by Eq. 18, as } v_i(\widehat{X}, \widetilde{y}_i) > v_i(\widehat{X}, \widehat{y}_i)) \\ \mu(\widehat{X}, \widehat{\mathcal{I}}) \geq \mu(\widetilde{X}, \widehat{\mathcal{I}}) & \text{(by hypothesis)} \\ \mu(\widetilde{X}, \widehat{\mathcal{I}}) = \mu(\widetilde{X}, \widetilde{\mathcal{I}}) & \text{(by Eq. 18, as } v_i(\widetilde{X}, \widetilde{y}_i) = v_i(\widetilde{X}, \widehat{y}_i) = \nu) \end{array}$$

Now, suppose that $\mu(\widehat{X}, \widehat{\mathcal{I}}) < \mu(\widetilde{X}, \widehat{\mathcal{I}})$. If $\widetilde{X}$ were a feasible solution for $\widehat{\mathcal{I}}$, then it would be a better solution than the optimum $\widehat{X}$. Hence, it must be $\widetilde{X} \notin \Phi(\widehat{\mathcal{I}})$, which implies that $\Pi \in \mathsf{C}^{\mathsf{vcgc}}_{\mathsf{none}}$. □

Theorem 4.6 *No problem $\Pi \in \mathsf{C}_{\mathsf{none}}$ admits a truthful mechanism.*

Proof. By contradiction, let $(\mathcal{A}, \mathcal{P})$ be a truthful mechanism for Π and let us denote $\widetilde{X} = \mathcal{A}(\widetilde{\mathcal{I}})$, and $\widehat{X} = \mathcal{A}(\widehat{\mathcal{I}})$. By the truthfulness we then have that both the following two inequalities hold:

$$u_i(\widehat{X}, \widehat{y}_i, p_i(\widehat{\mathcal{I}})) \quad \geq \quad u_i(\widetilde{X}, \widehat{y}_i, p_i(\widetilde{\mathcal{I}})), \quad (19)$$

$$u_i(\widetilde{X}, \widetilde{y}_i, p_i(\widetilde{\mathcal{I}})) \quad \geq \quad u_i(\widehat{X}, \widetilde{y}_i, p_i(\widehat{\mathcal{I}})). \quad (20)$$

By unfolding the definition of $u_i(\cdot)$ in the two equations above, and since $v_i(\widetilde{X}, \cdot) = \nu$, we obtain

$$p_i(\widehat{\mathcal{I}}) \oplus v_i(\widehat{X}, \widehat{y}_i) \geq p_i(\widetilde{\mathcal{I}}) \oplus v_i(\widetilde{X}, \widehat{y}_i) = p_i(\widetilde{\mathcal{I}}) \oplus \nu \quad \text{(by Eq. 19)} \quad (21)$$

$$p_i(\widetilde{\mathcal{I}}) \oplus v_i(\widetilde{X}, \widetilde{y}_i) = p_i(\widetilde{\mathcal{I}}) \oplus \nu \geq p_i(\widehat{\mathcal{I}}) \oplus v_i(\widehat{X}, \widetilde{y}_i) \quad \text{(by Eq. 20)} \quad (22)$$

These two inequalities, the hypothesis $v_i(\widehat{X}, \widetilde{y}_i) > v_i(\widehat{X}, \widehat{y}_i)$, and the strict monotonicity of '$\oplus$' imply the following contradiction

$$p_i(\widetilde{\mathcal{I}}) \oplus \nu \geq p_i(\widehat{\mathcal{I}}) \oplus v_i(\widehat{X}, \widetilde{y}_i) > p_i(\widehat{\mathcal{I}}) \oplus v_i(\widehat{X}, \widehat{y}_i) \geq p_i(\widetilde{\mathcal{I}}) \oplus \nu,$$

where the last inequality follows from Eq. 21. This completes the proof. □

The next results show that, in the case of the 2ND SHORTEST PATH and KNAPSACK$[\pi, \sigma]$ problems, VCG-consistent mechanisms do not fail because inappropriate.

Theorem 4.7 *The* 2ND SHORTEST PATH *problem is in* $\mathsf{C}_{\mathsf{none}}$. *Hence, it does not admit a truthful mechanism.*

Theorem 4.8 *The* KNAPSACK$[\pi, \sigma]$ *problem is in* $\mathsf{C}_{\mathsf{none}}$. *Hence, it does not admit a truthful mechanism.*

Proof. By Def. 4.3 and by Theorem 4.6, we only need to show that Condition 18 is satisfied. We consider any instance $\mathcal{I}$ with a single optimal solution $X = \mathcal{A}(\mathcal{I})$ and such that all the profits associated with items are different and positive. If i is the item with minimum profit in X, we define an instance $\mathcal{I}' = (\mathcal{I}_{-i}, \langle \pi'_i, \sigma'_i \rangle)$ in such a way that the optimal solution $X' = \mathcal{A}(\mathcal{I}')$ cannot include the i-th item, i.e., $v_i(X', \cdot) = 0$ Indeed:

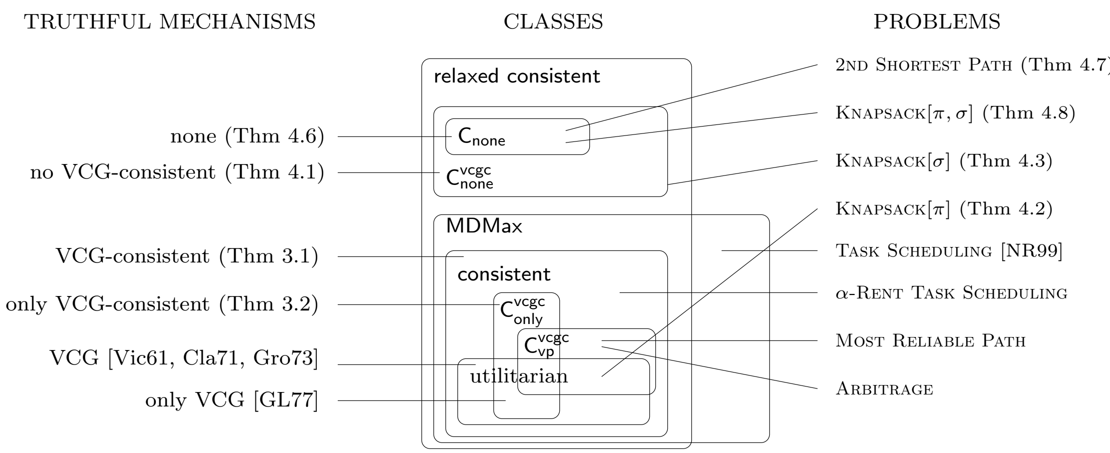
- $\pi'_i = \pi_i + \varepsilon$, for any $\varepsilon > 0$ such that $\pi_i + \varepsilon < \min_{j \neq i : X_j = 1}(\pi_j)$;
- $\sigma'_i = 1 + \max(\sigma_i + B - \sum_{i=1}^{n} X_i \sigma_i,\ \max_{j : X_j = 1}(\sigma_j))$.

Since $\sigma'_i > \sigma_i + B - \sum_{i=1}^{n} X_i \sigma_i$, it must be $X \notin \Phi(\mathcal{I}')$. Moreover, since $v_i(X, \langle \pi'_i, \sigma'_i \rangle) = \pi'_i > v_i(X, \langle \pi_i, \sigma_i \rangle) = \pi_i > 0$ and $v_i(\mathcal{A}(\mathcal{I}'), \cdot) = 0$, Condition 18 is met and, by Theorem 4.6, no truthful mechanism exists. □

Remark 4.1 (necessity of Constraint (1)) *Observe that if we remove Constraint (1) from the definition of consistent problems, then we obtain the class* relaxed consistent *(Def. 4.1). Theorems 4.7-4.8 imply that* $\emptyset \neq \mathsf{C}_{\mathsf{none}} \subseteq$ relaxed consistent. *Hence, Constraint (1) is necessary for guaranteeing the existence of truthful mechanisms.*

5 Conclusions and open problems

In the following figure we summarize the results obtained in this work. In particular, we have isolated several classes of problems involving selfish agents which are defined according to some mathematical properties. The inclusions mostly follow from the definitions, except for the result of Theorem 4.5.



Moreover, the results on the α-RENT TASK SCHEDULING problem and the fact that $\mathsf{C}_{\mathsf{none}} \neq \emptyset$ imply that $\mathsf{C}^{\mathsf{vcgc}}_{\mathsf{vp}} \subsetneq$ consistent $\subsetneq$ relaxed consistent. Since the TASK SCHEDULING problem in [NR99] can be formulated as a MDMax problem, the negative results in [NR99] also implies that consistent $\subsetneq$ MDMax. It would be interesting to prove analogous separation results among the classes. For instance, if KNAPSACK$[\sigma]$ had a truthful mechanism, then we would obtain $\mathsf{C}_{\mathsf{none}} \subsetneq \mathsf{C}^{\mathsf{vcgc}}_{\mathsf{none}}$. Combinatorial auction is a classic utilitarian problem (see e.g. [NR00]) which admits VCG mechanisms only. If would be interesting to find a *non-utilitarian* problem in $\mathsf{C}^{\mathsf{vcgc}}_{\mathsf{only}}$. Comparing $\mathsf{C}^{\mathsf{vcgc}}_{\mathsf{only}}$ and $\mathsf{C}^{\mathsf{vcgc}}_{\mathsf{vp}}$ would be also worthwhile. Investigating classes for which mechanisms that use non-optimal algorithms $\mathcal{A}$ remain truthful is an important issue. Interestingly, Theorem 3.1 also holds when algorithm $\mathcal{A}$, though non-optimal, is *maximal in its range* (see [NR00]), thus generalizing one of the results in [NR00] for utilitarian problems.

References

[ACP+03] C. Ambuehl, A. Clementi, P. Penna, G. Rossi, and R. Silvestri. Energy Consumption in Radio Networks: Selfish Agents and Rewarding Mechanisms. In *Proceedings of the 10th International Colloquium on Structural Information and Communication Complexity (SIROCCO)*, pages 1–16, 2003.

[AE03] L. Anderegg and S. Eidenbenz. Ad hoc-VCG: A Truthful and Cost-Efficient Routing Protocol for Mobile Ad Hoc Networks with Selfish Agents. In *Proc. of the Annual ACM International Conference on Mobile Computing and Networking (MOBICOM)*, 2003.

[AT01] A. Archer and E. Tardos. Truthful mechanisms for one-parameter agents. In *IEEE Symposium on Foundations of Computer Science*, pages 482–491, 2001.

[Cla71] E. Clarke. Multipart pricing of public goods. *Public Choice*, 8:17–33, 1971.

[Cra97] P. Cramton. The fcc spectrum auction: an early assessment. *Journal of Economics and Management Strategy*, 6:431–495, 1997.

[DG03] B. Dean and M. Goemans. Improved approximation algorithms for minimum-space advertisement scheduling. In *Proc. of ICALP*, number 2719 in LNCS, pages 1138–1152, 2003.

[FPS01] J. Feigenbaum, C.H. Papadimitriou, and S. Shenker. Sharing the cost of multicast transmissions. *Journal of Computer and System Sciences*, 63(1):21–41, 2001.

[FS02] J. Feigenbaum and S. Shenker. Distributed algorithmic mechanism design: Recent results and future directions. In *Proc. of the 6th International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications*, pages 1–13. ACM Press, 2002.

[GL77] J. Green and J.J. Laffont. Characterization of satisfactory mechanisms for the revelation of preferences for public goods. *Econometrica*, 45(2):727–738, 1977.

[Gro73] T. Groves. Incentives in teams. *Econometrica*, 41(4):617–631, 1973.

[McM95] K. McMillan. Selling spectrum rights. *Journal of Economic Perspectives*, pages 145–162, 1995.

[NR99] N. Nisan and A. Ronen. Algorithmic Mechanism Design. In *Proc. of the 31st Annual ACM Symposium on Theory of Computing (STOC)*, 1999.

[NR00] N. Nisan and A. Ronen. Computationally feasible VCG mechanisms. In *ACM Conference on Electronic Commerce*, pages 242–252, 2000.

[OR94] M.J. Osborne and A. Rubinstein. *A course in game theory*. MIT Press, 1994.

[Pap01] C. H. Papadimitriou. Algorithms, Games, and the Internet. In *33rd Annual ACM Symposium on Theory of Computing (STOC)*, 2001.

[PV03] P. Penna and C. Ventre. *Sharing the cost of multicast transmissions in wireless networks*. Technical report, European Project CRESCCO, available at `http://www.ceid.upatras.gr/crescco/`, 2003. Also submitted for publication.

[Vic61] W. Vickrey. Counterspeculation, auctions and competitive sealed tenders. *J. Finance*, 16:8–37, 1961.

A List of MDMax problems

A.1 The Longest Multiplicative Path Problem

- Private instance. Let $G = (V, E)$ be an directed weighted graph with edge weights $w_e \in \Theta$, and $s, t \in V$ be two special vertices. The true instance is $\mathcal{I}_T = (I_P, I_T)$ where (i) $I_P = \langle (V, E), s, t \rangle$, and (ii) $I_T = W$, that is the private part on the instance is the set of edge weights.

- Reported instance. Each agent a_e makes public an edge weight $r_e \in \Theta$, so $\mathcal{I}_R = (I_P, I_R)$, where $I_R = \{r_e \mid e \in E\}$.

- Feasible solutions. For any instance $\mathcal{I} = (I_P, I)$ specified by edge weights $y_e \in \Theta$, the set $\Phi(\mathcal{I})$ of feasible solutions, also denoted by Φ_{st}, is the set of all st-paths in (V, E). Φ_{st} is constant with respect to I, so it enjoys Constraint 1.

- Objective function. For any $\mathcal{I} = (I_P, I)$ and for any st-path $\pi \in \Phi_{st}$, $\mu(\pi, \mathcal{I}) = \prod_{e \in \pi} y_e$.

- Valuation functions. For any $\pi \in \Phi_{st}$ and for any $y_e \in \Theta$, let

$$v_e(\pi, y_e) = \begin{cases} v_e^0 = 1 & \text{if } e \text{ is not on the path } \pi \\ y_e & \text{otherwise.} \end{cases} \quad . \tag{23}$$

  Hence, $\mu(\pi, \mathcal{I}) = \prod_{e \in E} v_e(\pi, y_e)$.

- Agent Payments and Utility functions. Each agent a_e receives a payment P_e and then achieves an utility $u_e(\pi, w_e, P_e) = v_e(\pi, w_e) \cdot P_e$

A.2 The α-Rent Task Scheduling problem

We are given k tasks which need to be allocated to n machines, each of them corresponding to one agent. Let t_j^i denote the minimum amount of time machine i is capable of performing task j and let X_i be the the set of tasks allocated to agent a_i. The goal is to minimize the makespan, that is, the maximum, over all machines, completion time. We can formalize the above problem as a consistent problem as follows:

- Private instance. The instance corresponding to the true agents' types is $\mathcal{I}_T = (I_P, I_T)$, where $I_P = \langle k, n \rangle$ and $I_T = \langle t_1, \ldots, t_n \rangle$, where $t_i = \langle t_1^i, \ldots, t_k^i \rangle$.

- Reported instance. Each agent a_i makes public a vector $r_i = \langle r_1^i, \ldots, r_k^i \rangle$ representing the (reported) task execution times on machine i, then $\mathcal{I}_R = (I_P, I_R)$ is the input of the algorithm.

- Feasible solutions. $\Phi(\mathcal{I})$ is the set of all partitions $X = X_1, \ldots, X_n$ of the tasks, where X_i denotes the tasks allocated to agent a_i. $\Phi(\mathcal{I})$ is constant with respect to the private part of the input, so it enjoys Constraint 1.

- Objective function. For any instance $\mathcal{I} = \langle I_P, I \rangle$ with $I = \langle y_1, \ldots, y_n \rangle$, we define $\mu(X, \mathcal{I}) = \max_{1 \le i \le n} \sum_{j \in X_i} y_j^i$.

- Valuation functions. For any $\mathcal{I}$, we define $v_i(X, y_i) = -\sum_{j \in X_i} y_j^i$, that is, the completion time of machine i. Hence, it holds that $\mu(X, \mathcal{I}) = \max_{i=1}^n -v_i(X, y_i) = \min_{i=1}^n v_i(X, y_i)$. Agent a_i is not involved in the solution X if $X_i = \emptyset$. In this case, $v_i(X, \cdot) = 0 = v_i^0$.

- Payments and utility function. We define the utility function as $u_i(X, t_i, P_i) = \min(v_i(X, t_i), P_i)$.

- Goal. Minimize the makespan.

A.3 The 2nd Shortest Path problem

Let us consider an undirected weighted graph $G = (V, E, W)$ and two nodes $s, t \in V$. The objective is to find a path whose length is minimal among all st-paths that have no minimal length in G. More formally:

- Private instance. Let $G = (V, E, W)$ be an undirected weighted graph $G = (V, E, W)$ with real edge weights w_e. The true instance is $\mathcal{I}_T = (I_P, I_T)$ where (i) $I_P = \langle (V, E), s \in V, t \in V \rangle$, and (ii) $I_T = W$.
- Reported instance. $\mathcal{I}_R = (I_P, I_R)$, where I_R is the set of reported types: each agent a_e makes public an edge weight r_e.
- Feasible output. For any instance $\mathcal{I}$, if Φ_{st} is the set of all st-paths in (V, E) and $X_1^*(\mathcal{I}) \subseteq \Phi_{st}$ is the subset of the shortest st-paths, $\Phi(\mathcal{I}) = \Pi_{st}(\mathcal{I}) \setminus X_1^*(\mathcal{I})$.
- Objective function. $\mu(\pi, \mathcal{I}_R) = \sum_{e \in \pi} -r_e$.
- Valuation functions. If for every edge $e \in E$ and for every st-path π

  $$v_e(\pi, \mathcal{I}_R) = \begin{cases} -r_e & \text{if } e \in \pi \\ 0 & \text{otherwise} \end{cases},$$

  then $\mu(\pi, \mathcal{I}_R) = \sum_{e \in E} v_e(\pi, r_e)$.
- Goal. Find a path whose length is minimal among all st-paths that have no minimal length in G. That is, $\mathcal{A}(\mathcal{I}_R) \in \text{argmax}\{\mu(\pi, \mathcal{I}_R) \mid \pi \in \Phi(\mathcal{I}_R)\}$.

A.4 The Knapsack problem(s)

We consider a variant of the classical optimization problem called 0-1 Knapsack which can be described as follows. We are given a set of n items $\{1, \ldots, n\}$, each one characterized by a *profit* π_i and a *size* σ_i. The goal is to find a set of items which maximize the total profit and such that its total occupancy does not exceed a given capacity B. Each item i is associated with an agent a_i that holds a part of the instance. Depending on how the private part of the instance is defined we distinguish the following three problems:

- Knapsack$[\pi]$, where each agent a_i only holds the profit π_i associated with each item i, whereas every size σ_i is public knowledge; then, $\mathcal{I}_T = (I_P, I_T)$ where $I_P = (n, B, \langle \sigma_1, \ldots, \sigma_n \rangle)$ and $I_T = \langle t_1, \ldots, t_n \rangle$, with $t_i = \pi_i$.
- Knapsack$[\sigma]$, where each agent a_i only holds the size σ_i associated with each item i, whereas every profit π_i is public knowledge; then, $\mathcal{I}_T = (I_P, I_T)$ where $I_P = (n, B, \langle \pi_1, \ldots, \pi_n \rangle)$ and $I_T = \langle t_1, \ldots, t_n \rangle$, with $t_i = \sigma_i$.
- Knapsack$[\pi, \sigma]$ where each agent a_i holds both the profit π_i and the size σ_i associated with each item i, that is, $t_i = \langle \pi_i, \sigma_i \rangle$; then $\mathcal{I}_T = (I_P, I_T)$ where $I_P = (n, B)$ and $I_T = \langle t_1, \ldots, t_n \rangle$, with $t_i = (\pi_i, \sigma_i)$.

Every such problem can be further specified as follows:

- Reported instance. $\mathcal{I}_R = (I_P, I_R)$ such that r_i is the type reported by agent a_i and $I_R = (r_1, \ldots, r_n)$.
- Feasible output. For any instance $\mathcal{I}$, $\Phi(\mathcal{I}) = \{X \in \{0,1\}^n \mid \sum_{i=1}^n X_i \sigma_i \leq B\}$.
- Objective function. For every $X \in \Phi(\mathcal{I})$, $\mu(X, \mathcal{I}) = \sum_{i=1}^n X_i \pi_i$ is the total profit.
- Valuation function. We define $v_i(X, y_i) = X_i \pi_i$.
- Goal. Maximize the total profit.

B Postponed proofs

B.1 Theorem 3.1

Proof. Suppose that for each i there exists a function $h_i(\mathcal{I}_{-i})$ such that the payment function $\mathcal{P}$ satisfies (4), but that $(\mathcal{A}, \mathcal{P})$ is not truthful. Then there exist an agent a_i and a reported type $r_i \neq t_i$ such that for any I_{R-i}:

$$u_i(\mathcal{A}, \mathcal{I}_T, t_i) < u_i(\mathcal{A}, \mathcal{I}_R, t_i)$$

where $\mathcal{I}_T$ denotes the instance $(I_P, \langle (I_R)_{-i}, t_i \rangle)$ defined by reporting the true type t_i and $\mathcal{I}_R$ the instance $(I_P, \langle (I_R)_{-i}, r_i \rangle)$ by reporting r_i. So, denoted $X_T = \mathcal{A}(\mathcal{I}_T)$ and $X_R = \mathcal{A}(\mathcal{I}_R)$, Equations (3) and (4) imply that

$$v_i(X_T, t_i) \oplus (\mu_{-i}(X_T, \mathcal{I}_T) \oplus h_i(\mathcal{I}_{-i})) < v_i(X_R, t_i) \oplus (\mu_{-i}(X_R, \mathcal{I}_R) \oplus h_i(\mathcal{I}_{-i}))$$

Consequently, by the associativity property:

$$(v_i(X_T, t_i) \oplus \mu_{-i}(X_T, \mathcal{I}_T)) \oplus h_i(\mathcal{I}_{-i}) < (v_i(X_R, t_i) \oplus \mu_{-i}(X_R, \mathcal{I}_R)) \oplus h_i(\mathcal{I}_{-i})$$

that is, since $\mu_{-i}(X_R, \mathcal{I}_R) = \mu_{-i}(X_R, \mathcal{I}_T)$

$$\mu(X_T, \mathcal{I}_T) \oplus h_i(\mathcal{I}_{-i}) < \mu(X_R, \mathcal{I}_T) \oplus h_i(\mathcal{I}_{-i})$$

and, by the monotonicity property

$$\mu(X_T, \mathcal{I}_T) < \mu(X_R, \mathcal{I}_T)$$

which is a contradiction, as X_T is the output which maximizes $\mu(X, \mathcal{I}_T)$, for every $X \in \Phi(\mathcal{I}_T)$, and, by constraint (1), $X_R \in \Phi(\mathcal{I}_R) = \Phi(\mathcal{I}_T)$. □

B.2 Theorem 4.7

Proof. Let $\mathcal{I} = \langle \mathcal{I}_{-1}, w_e \rangle$ and $\mathcal{I}' = \langle \mathcal{I}_{-1}, w'_e \rangle$ be the instances defined to prove Proposition **??**. It can be easily verified that:

- $X_2^*(\mathcal{I}') \notin \Phi(\mathcal{I})$ (we recall that $X_2^*(\mathcal{I}') = X_1^*(\mathcal{I})$);
- e has been chosen in such a way that $e \in X_1^*(\mathcal{I}) - X_2^*(\mathcal{I})$, so $v_e(X_1^*(\mathcal{I}), \cdot) = v_e(X_2^*(\mathcal{I}'), \cdot) \neq 0$, whereas $v_e(X_2^*(\mathcal{I}), \cdot) = 0$. Namely, $v_e(X_2^*(\mathcal{I}'), w_e) = -w_e$ and $v_e(X_2^*(\mathcal{I}'), w'_e) = -w'_e$, then, being by construction $w'_e > w_e$, it is $v_e(X_2^*(\mathcal{I}), w_e) > v_e(X_2^*(\mathcal{I}'), w'_e)$.

This completes the proof. □

B.3 Theorem 4.3

Proof. By Def. 4.2 and by Theorem 4.1, we only need to show that Condition 17 is satisfied. Towards this aim, given any instance $\mathcal{I}$ with a single optimal solution, let us consider any item i into the solution $X = \mathcal{A}(\mathcal{I})$. We can construct a new instance $\mathcal{I}'$ by only incrementing the occupancy of item i as follows: $\sigma'_i = \sigma_i + B - \sum_{i=1}^{n} X_i \sigma_i + 1$. As $\mathcal{A}(\mathcal{I}) \notin \Phi(\mathcal{I}')$, and $\mu(\mathcal{A}(\mathcal{I}), \mathcal{I}') > \mu(\mathcal{A}(\mathcal{I}'), \mathcal{I}')$, Condition 17 is met. Hence the theorem follows. □