Shortest Path to Mechanism Design

Rudolf Müller and Marc Uetz

Abstract Mechanism design is concerned with computing desired outcomes in situations where data is distributed among selfish agents. We discuss some of the most fundamental questions in the design of mechanisms, and derive simple answers by interpreting the problem in graph-theoretic terms. Specifically, much of mechanism design is thereby reformulated as shortest path problems.

1 Terminology in Mechanism Design

The basic premise in the design of mechanisms is the presence of a set of noncooperative agents, each of which is equipped with a piece of private information. This private information is called the agent's *type*. The set of all types, one for each agent, effectively defines the input of some optimization problem. There is a principal, henceforth called the *mechanism designer*, in charge of selecting or computing a solution for the optimization problem, referred to as the *outcome*. Each agent has a certain *valuation* for the chosen outcome, which in general depends on her type. The mechanism designer aims to choose an outcome so as to optimize some global objective function, which typically, but not necessarily, depends on the agents' types. A classical example for such a global objective is the so-called utilitarian maximizer, which is the outcome that maximizes the total valuations of all agents. However, the fundamental problem in choosing that outcome is that the mechanism designer does not know the types of the agents. In other words, the (true) input for the problem to be solved is not known a priori. In such situations, the mechanism designer needs to create incentives so that the agents share their private information.

R. Müller
Department of Quantitative Economics, Maastricht University, P.O. Box 616, 6200MD Maastricht, The Netherlands
e-mail: r.muller@maastrichtuniversity.nl

M. Uetz (✉)
Department of Applied Mathematics, University of Twente, P.O. Box 217, 7500AE Enschede, The Netherlands
e-mail: m.uetz@utwente.nl

A.S. Schulz et al. (eds.), *Gems of Combinatorial Optimization and Graph Algorithms*, DOI 10.1007/978-3-319-24971-1_8

The best known example of a mechanism is probably *Vickrey's second price auction*: When auctioning a single, indivisible item, it gives the item to the bidder with the highest bid and sets the price for the winner to the second highest bid. This makes truthful bidding a dominant strategy. Like in Vickrey's auction, mechanism design generally allows for payments, also called transfers. Central questions are: Is there a payment rule that incentivizes agents to tell the truth? And which flexibility is there in choosing such payment? Indeed, if payments are taxes, we might want to reduce tax load and make them as small as possible, while in auction settings we might want to increase the revenue for the seller, hence maximize payments. We will answer these questions using basic graph-theoretic concepts.

1.1 Types, Outcomes and Valuations

To fix some notation, we denote by $\{1, \dots, n\}$ the set of agents and let A be the set of possible outcomes. Outcome space A is allowed to have infinitely many, even uncountably many, elements. Denote the *type* of agent $i \in \{1, \dots, n\}$ by t_i. Let T_i be the *type space* of agent i; that is, the possible types that agent i might have. Type spaces T_i can be arbitrary sets. Agent i's preferences over outcomes are modeled by the *valuation function* $v_i\colon A \times T_i \to \mathbb{R}$, where $v_i(a, t_i)$ is the valuation of agent i for outcome a when she has type t_i. Let us give three illustrative examples of this rather abstract and general definition.

Example 1 Consider the classical situation where the agents are n bidders in a *private value single-item auction*. Here, the mechanism designer is the auctioneer, and each agent i's type is described by the maximum amount she would be willing to pay for the indivisible item, say $t_i \in \mathbb{R}_{\geq 0}$. The $n + 1$ possible outcomes A are then all possible assignments of the item to the agents, namely, $a \in \{0, 1\}^n$ so that $\sum_{i=1}^{n} a_i \leq 1$, meaning that agent i gets the item iff $a_i = 1$. The valuation functions are $v_i(a, t_i) = t_i$ if $a_i = 1$ and $v_i(a, t_i) = 0$ otherwise. If randomized allocations are allowed, we get $a \in [0, 1]^n$ instead of $a \in \{0, 1\}^n$, and as a general expression for the valuations, $v_i(a, t_i) = a_i t_i$. Note that the underlying combinatorial optimization problem to maximize the total valuation is trivial: choose the maximum among all t_i's and give the item to that bidder. The problem owes its interest only to the fact that the types t_i are private information of the bidders.

Example 2 An extension of Example 1 is an auction with a finite set of m heterogenous and indivisible items. Agent types $t_i \in \mathbb{R}^m_{\geq 0}$ denote values for each item, with the value for bundle $B \subseteq \{1, \dots, m\}$ being equal to $\sum_{j \in B} t_{ij}$. The outcome space is then all partitions of m items into $n + 1$ bundles, including bundles that stay at the auctioneer. Allowing even for arbitrary values for bundles rather than additive valuations, the problem is called a *combinatorial auction*. It has the same outcome space, but a richer set of types. For combinatorial auctions, the underlying optimization problem is the set packing problem, and generally NP-hard.

Example 3 Consider the following *demand rationing problem*. The mechanism designer is asked to distribute one unit of a divisible good amongst the agents, and the type of each of the n agents is her minimal demand $t_i \in (0, 1]$ for the good. In this setting, the outcomes A are all possible fractional distributions of the good over the agents. That is, all vectors $a \in [0, 1]^n$ so that $\sum_{i=1}^{n} a_i \leq 1$. The valuation functions are $v_i(a, t_i) = \min\{0, a_i - t_i\}$. Note that the valuations are either zero or negative and express the amount of demand rationing that an agent suffers. Here again, the underlying optimization problem to maximize the total valuation is almost trivial: if the demand exceeds one unit, any allocation of one unit is optimal, as long as no agent i receives more than her demand t_i. Otherwise, assign to every agent at least her demand. Again, the problem gets interesting as soon as the demands are private information.

Observe that the outcome spaces of the discrete version of the first and second example are finite, while it is infinite in the third. Also observe that in the first and the third example, the types of an agent i are single dimensional in the sense that $t_i \in \mathbb{R}$, while in the second example it is multi-dimensional. The above examples address utility maximization, but of course other objectives are possible as well. All that follows is general enough to cover these cases, too.

1.2 Mechanisms and Incentive Compatibility

The mechanism design problem we want to address here involves money to set incentives. More specifically, the task will be to find a mechanism that defines which allocation to choose and how much the agents need to pay for it. Payments could be both positive or negative. Formally, a *mechanism* (f, π) consists of an *allocation rule* $f : \Pi_{i=1}^{n} T_i \to A$ and a *payment rule* $\pi : \Pi_{i=1}^{n} T_i \to \mathbb{R}^n$. In fact what we describe here is a *direct revelation mechanism*, meaning that the only action of any agent is to reveal her type t_i. The allocation rule chooses for a vector t of type reports an outcome $f(t)$, and the payment rule assigns a payment $\pi_i(t)$ to be made by agent i.

On the agents' side we assume *quasi-linear utilities*, that is, the *utility* of agent i when the reported type vector is t, and the outcome is $a = f(t)$, equals valuation minus payment, that is,

$$v_i(f(t), t_i) - \pi_i(t).$$

Agents are assumed to be utility maximizers. Hence an agent, once asked by the mechanism designer about her type t_i, could decide to misreport her true type and pretend to have another type $s_i \in T_i$ instead. Indeed, if that false report s_i would result in an outcome and payment that yields higher utility than reporting true type t_i, agent i would not be truthful about her type. Once a mechanism (f, π) is fixed, the agents effectively play a noncooperative game of incomplete information, in which each agent i's strategy is to choose a reported type s_i, given true type t_i, so as to maximize her utility. Incomplete information refers to the fact that neither the mechanism designer nor any agent knows the types of the (other) agents. This implies for agents that their actions in equilibrium should ideally be robust against this uncertainty about other agents' types.

For what follows, let us denote by (t_i, t_{-i}) the vector of type reports when i reports t_i and the other agents' reports are t_{-i}. The following definition expresses the requirement that truthfulness yields the maximal utility for any agent, independent of the reports of other agents. This exactly corresponds to a dominant strategy equilibrium in the just mentioned noncooperative game.

Definition 4 (*Dominant strategy incentive compatible*) A mechanism (f, π) is called *dominant strategy incentive compatible*, or *truthful* for short, if for every agent i, every type $t_i \in T_i$, all type vectors t_{-i} that the other agents could report and every type $s_i \in T_i$ that i could report instead of t_i:

$$v_i(f(t_i, t_{-i}), t_i) - \pi_i(t_i, t_{-i}) \geq v_i(f(s_i, t_{-i}), t_i) - \pi_i(s_i, t_{-i}).$$

If for allocation rule f there exists a payment rule π such that (f, π) is a dominant strategy incentive compatible mechanism, then f is called *implementable in dominant strategies*, in short *implementable*.

This is a strong requirement. We can motivate it as follows: when the mechanism designer's goal is to optimize some objective function that depends on the true types of agents, he should better know these types. From the agents' perspectives the equilibrium is desirable as it provides a high degree of robustness: independent of the other agents' types, truthfulness is a best action. Even more, being truthful is optimal independent of the other agents' types *and* reports. As a matter of fact, this restriction to truthful mechanisms is much less restrictive than it seems: The celebrated *revelation principle* states that *any* mechanism with corresponding strategies of the agents that form an ex-post equilibrium can be equivalently replaced by a direct revelation mechanism in which truthfulness is a dominant strategy equilibrium. Here an ex-post equilibrium refers to a strategy profile in an incomplete information game, in which no agent would have liked to deviate after he has learned the true type of the other agents, and thus is ex-post satisfied with her action. In that sense truthfulness is a requirement that can be made *without loss of generality*.

1.3 Agenda

A first question that comes up is this: Which of all possible allocation rules are actually implementable? In other words, is it possible to augment all possible allocation rules f by appropriate payments π so the result is a truthful mechanism?

A second question is this: Assume that we have an implementable allocation rule f, how much flexibility does the mechanism designer have with respect to the payments π? For example, which are the minimal or maximal payments that implement that allocation rule?

The second question hints to an important property of mechanisms that is known as *revenue* or *payoff equivalence*. It describes situations where the payment rule π of a mechanism is already uniquely defined by the choice of an allocation rule f, up to adding constants. Let us formally define what it is.

Definition 5 (*Revenue equivalence*) An allocation rule f that is implementable satisfies the *revenue equivalence* property if for any two dominant strategy incentive compatible mechanisms (f, π) and (f, π') and any agent i there exist constants $h_i(t_{-i})$ that only depend on the reported types of the other agents, t_{-i}, such that

$$\forall\, t_i \in T_i : \pi_i(t_i, t_{-i}) = \pi_i'(t_i, t_{-i}) + h_i(t_{-i}).$$

The main purpose of this chapter is to show how these two fundamental questions can be rather elegantly addressed by reformulating them in terms of shortest path problems in graphs. Moreover, we hope that the reader who finds the previous concepts hard to digest, will find the graph-theoretic approach helpful and may even start to appreciate some of these concepts.

2 The Type Graph

Let us fix agent i and the reports, t_{-i}, of the other agents. For simplicity of notation we write $f(t_i)$ instead of $f(t_i, t_{-i})$, and $\pi(t_i)$ instead of $\pi_i(t_i, t_{-i})$, whenever t_{-i} is clear from the context.

For any allocation rule f, agent i and types of the other agents, t_{-i}, we define a *type graph* $G_f = G_f(i, t_{-i})$ as follows.[1] It has node set T_i and contains an arc from any node $s_i \in T_i$ to any other node $t_i \in T_i$ of length

$$\ell(s_i, t_i) = v_i(f(t_i), t_i) - v_i(f(s_i), t_i).$$

Note that there is an arc between any two nodes, as depicted in Fig. 1. The type graph is a complete, directed, and possibly infinite graph.

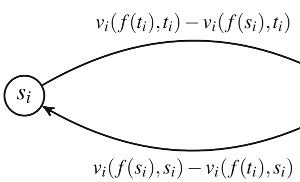

Fig. 1 Arc lengths of the type graph G_f between any two types $s_i, t_i \in T_i$ for agent i

[1] The type graph depends on the agent i and reports t_{-i} of the other agents. In order to keep notation simple, we will suppress the dependence on i and t_{-i}, and simply write G_f.

Here, $\ell(s_i, t_i)$ represents the gain in valuation for an agent truthfully reporting type t_i instead of lying some other type s_i. This gain could be positive or negative. A *path* from node s_i to node t_i in G_f, or (s_i, t_i)-path for short, is defined as $P = (s_i = t_{i_0}, t_{i_1}, \dots, t_{i_k} = t_i)$ such that $t_{i_j} \in T_i$ for $j = 0, \dots, k$. Denote by $length(P)$ the length of this path. A *cycle* is a path with $s_i = t_i$. We assume that each node represents a path from itself to itself and define the length of this (t_i, t_i)-path to be 0. Define $\mathscr{P}(s_i, t_i)$ to be the set of all (s_i, t_i)-paths in G_f.

In order to rephrase the question of implementability of an allocation rule in graph-theoretic terms, we need to define distances in the type graph. We let

$$dist_{G_f}(s_i, t_i) := \inf_{P \in \mathscr{P}(s_i, t_i)} length(P).$$

Observe that we here take the infimum over a possibly infinite set of (s_i, t_i)-paths in case the type graph has infinitely many nodes. If type spaces are finite, this is just the ordinary definition of shortest paths. In general, $dist_{G_f}(s_i, t_i)$ could be unbounded. However, if G_f does not contain a cycle of negative length then $dist_{G_f}(s_i, t_i)$ is finite, because the length of any (s_i, t_i)-path is bounded from below by $-\ell(t_i, s_i)$. This *nonnegative cycle* property of type graphs G_f turns out to be necessary and sufficient for the implementability of an allocation rule f.

3 Implementability and Payment Rules

A characterization of implementable allocation rules follows from the fact that incentive compatible payment rules exactly correspond to *node potentials* in the type graphs G_f. Recall that a node potential in a directed graph is a node labeling that satisfies the triangle inequality for each arc. Node potentials can be defined if and only if a graph has no cycle of negative length. In the mechanism design literature, the resulting characterization is known as *cyclical monotonicity*.

Theorem 6 (Cyclical monotonicity) *An allocation rule f is implementable if and only if none of the type graphs G_f contains a cycle of negative length.*

Proof Fix agent i and any type vector of the other agents t_{-i}. Assume f is implementable, then incentive compatibility translates into $\pi(t_i) - \pi(s_i) \leq \ell(s_i, t_i)$, for any $s_i, t_i \in T_i$. Adding up these inequalities along the arcs of any cycle shows that the cycle has nonnegative length.

On the other hand, assume that G_f has no cycle of negative length. Then fix some $t_{i_0} \in T_i$, and observe that the distance function $dist_{G_f}(t_{i_0}, t_i)$ is well defined for all $t_i \in T_i$. Moreover, $\pi(t_i) := dist_{G_f}(t_{i_0}, t_i)$ defines an incentive compatible payment rule for f, since $dist_{G_f}(t_{i_0}, t_i) \leq dist_{G_f}(t_{i_0}, s_i) + \ell(s_i, t_i)$ for all $s_i, t_i \in T_i$. □

An immediate consequence of Theorem 6 is that an implementable allocation rule must satisfy the nonnegative cycle property for all cycles of length two, that is,

$\ell(s_i, t_i) + \ell(t_i, s_i) \geq 0$. This is known in the mechanism design literature as *(weak) monotonicity* of an implementable allocation rule. Due to its importance, we explicitly state this consequence as a separate theorem.

Theorem 7 (Weak monotonicity) *If an allocation rule f is implementable, then for every i and t_{-i}, the following (weak) monotonicity condition must be satisfied: For all $s_i, t_i \in T_i$*

$$(v_i(f(t_i), t_i) - v_i(f(t_i), s_i)) + (v_i(f(s_i), s_i) - v_i(f(s_i), t_i)) \geq 0. \tag{1}$$

For example, in the single-item auction setting from Example 1 where $t_i \in \mathbb{R}_{\geq 0}$ and $v_i(f(t_i), t_i) = t_i \cdot f_i(t_i)$, inequality (1) is equivalent to $(f_i(t_i) - f_i(s_i))(t_i - s_i) \geq 0$. In other words, for any implementable mechanism the probability of allocating the item to a bidder, expressed by $f_i(t_i)$, must be monotonically non-decreasing in the valuation that the bidder has for the item.

The proof of Theorem 6 suggests even more: incentive compatible payment rules for an implementable allocation rule f can be defined by computing shortest paths in type graphs. Let us push this idea a little further. For each agent i we choose the payment at some type $t_{i_0} \in T_i$ arbitrarily,[2] say $\pi_i(t_{i_0}) = 0$. Then, in i's type graph G_f, we compute shortest path lengths $dist_{G_f}(t_{i_0}, t_i)$ to all types $t_i \in T_i$. This is well defined by Theorem 6. From the proof of Theorem 6, it follows that an incentive compatible payment rule can be defined by letting

$$\pi^+(t_i) := dist_{G_f}(t_{i_0}, t_i)$$

for all $t_i \in T_i$. But we can also compute shortest path lengths from all types $t_i \in T_i$ to t_{i_0}, which yields another incentive compatible payment rule, namely

$$\pi^-(t_i) := -dist_{G_f}(t_i, t_{i_0})$$

for all $t_i \in T_i$. Indeed, incentive compatibility for the latter follows because for all $s_i, t_i \in T_i$, $dist_{G_f}(s_i, t_{i_0}) \leq \ell(s_i, t_i) + dist_{G_f}(t_i, t_{i_0})$, and thus $-dist_{G_f}(t_i, t_{i_0}) \leq -dist_{G_f}(s_i, t_{i_0}) + \ell(s_i, t_i)$. The following lemma relates these payment rules to *any* incentive compatible payment rule π.

Lemma 8 *Let π be any payment rule that implements allocation rule f, and assume w.l.o.g.*[3] *that $\pi(t_{i_0}) = 0$ for $t_{i_0} \in T_i$ and all i, then*

$$\pi^- \leq \pi \leq \pi^+.$$

[2] For example in auction settings $\pi(t_{i_0}) = 0$ is a natural choice for any type t_{i_0} for which bidder i does not win anything in the auction.

[3] This is indeed no loss of generality as we can replace $\pi_i(\cdot)$ by $\pi_i(\cdot) - \pi_i(t_{i_0})$ otherwise, for all i.

Proof Consider agent i. Observe that, for any $s_i, t_i \in T_i$, by adding up the incentive compatibility constraints of π along any (s_i, t_i)-path P in G_f, we have $\pi(t_i) \leq \pi(s_i) + \mathit{length}(P)$. Taking the infimum over all such paths in $\mathscr{P}(s_i, t_i)$, we get

$$\pi(t_i) \leq \pi(s_i) + \mathit{dist}_{G_f}(s_i, t_i). \tag{2}$$

Therefore, letting $s_i = t_{i_0}$ in (2) we get $\pi(t_i) = \pi(t_i) - \pi(t_{i_0}) \leq \mathit{dist}_{G_f}(t_{i_0}, t_i) = \pi^+(t_i)$. On the other hand, letting $t_i = t_{i_0}$ and $s_i = t_i$ in (2), we see that $\pi(t_i) = \pi(t_i) - \pi(t_{i_0}) \geq -\mathit{dist}_{G_f}(t_i, t_{i_0}) = \pi^-(t_i)$. $\square$

Equipped with Lemma 8 we get a characterization of implementable allocation rules that satisfy revenue equivalence.

Theorem 9 (Characterization of revenue equivalence) *Let f be an allocation rule that is implementable. Then f satisfies revenue equivalence if and only if in all type graphs G_f the distances are anti-symmetric, that is, $\mathit{dist}_{G_f}(s_i, t_i) = -\mathit{dist}_{G_f}(t_i, s_i)$ for all $s_i, t_i \in T_i$.*

Proof Let f satisfy revenue equivalence. Fix an agent i and a report vector t_{-i} of the other agents, and let G_f be i's type graph. Let $s_i, t_i \in T_i$. Then the functions $\mathit{dist}_{G_f}(s_i, \cdot)$ and $\mathit{dist}_{G_f}(t_i, \cdot)$ define both incentive compatible payment rules for f. By revenue equivalence they differ only by a constant $h(t_{-i})$. Hence we have that $\mathit{dist}_{G_f}(s_i, \cdot) - \mathit{dist}_{G_f}(t_i, \cdot) = h(t_{-i})$. Especially, if we plug s_i and t_i into this equation we get that $\mathit{dist}_{G_f}(s_i, s_i) - \mathit{dist}_{G_f}(t_i, s_i) = \mathit{dist}_{G_f}(s_i, t_i) - \mathit{dist}_{G_f}(t_i, t_i)$. As $\mathit{dist}_{G_f}(s_i, s_i) = \mathit{dist}_{G_f}(t_i, t_i) = 0$, we see $\mathit{dist}_{G_f}(s_i, t_i) = -\mathit{dist}_{G_f}(t_i, s_i)$.

For the reverse direction, take an implementable allocation rule f and suppose that $\mathit{dist}_{G_f}(s_i, t_i) = -\mathit{dist}_{G_f}(t_i, s_i)$ for all $s_i, t_i \in T_i$. Take any incentive compatible payment rule π and a type $t_{i_0} \in T_i$. By $\mathit{dist}_{G_f}(t_{i_0}, \cdot) = -\mathit{dist}_{G_f}(\cdot, t_{i_0})$ and Lemma 8 we get $\pi(\cdot) - \pi(t_{i_0}) = \pi^-(\cdot) = \pi^+(\cdot) = \mathit{dist}_{G_f}(t_{i_0}, \cdot)$. Thus for any two payment rules π and π' we get $\pi(\cdot) - \pi(t_{i_0}) = \pi'(\cdot) - \pi'(t_{i_0})$, which proves revenue equivalence by letting $h(t_{-i}) := \pi(t_{i_0}) - \pi'(t_{i_0})$. $\square$

We have seen until here that, up to normalization at some type t_{i_0}, there is a minimal and a maximal payment rule, and revenue equivalence holds when these two coincide. But there is even more we can say about the space of payment rules, and it is surprisingly simple. We need two more definitions. For two payment rules π and π' that implement an allocation rule f, define

$$\hat{\pi}(t_i) := \max\{\pi(t_i), \pi'(t_i)\} \quad \text{and} \quad \check{\pi}(t_i) := \min\{\pi(t_i), \pi'(t_i)\}$$

as the *join* and *meet* of π and π'. Then these payment rules are incentive compatible, too. To conveniently formulate this, let us again assume w.l.o.g. that all payment rules are normalized at some t_{i_0} for all agents i, so that $\pi(t_{i_0}) = 0$.

Theorem 10 (Payment rules form a lattice) *Consider an implementable allocation rule f. Then the set of all payment rules that implement f define a bounded lattice* *with respect to the meet and join definition. The payment rules π^- and π^+ are the minimal, respectively maximal elements of that lattice.*

Proof We are only left to prove that join $\hat{\pi}$ and meet $\check{\pi}$ are both incentive compatible payments for f. This follows immediately from the fact that payment rules correspond to node potentials in the type graphs, and it is well known that the set of node potentials in a directed graph forms a lattice. Let us give the simple proof here: Take any two $s_i, t_i \in T_i$, and first consider $\hat{\pi}$. Say w.l.o.g. that $\hat{\pi}(t_i) = \pi(t_i)$. Then, as π is incentive compatible, $\hat{\pi}(t_i) = \pi(t_i) \leq \pi(s_i) + \ell(s_i, t_i) \leq \hat{\pi}(s_i) + \ell(s_i, t_i)$. Likewise, for $\check{\pi}$, assume w.l.o.g. that $\check{\pi}(s_i) = \pi(s_i)$, then, as π is incentive compatible, $\check{\pi}(s_i) + \ell(s_i, t_i) = \pi(s_i) + \ell(s_i, t_i) \geq \pi(t_i) \geq \check{\pi}(t_i)$. □

4 Theorems at Work

We here sketch how the theorems just derived can help to derive qualitative insights into some relevant problems.

4.1 Implementations for Demand Rationing

Recall the demand rationing problem introduced in Example 3. One possible allocation rule f would be the *dictatorial rule*: Choose some agent, say agent 1 and give her whatever she demands, $a_1 = f_1(t) := t_1$. Divide the remainder equally over the remaining agents, $a_i = f_i(t) := (1 - t_1)/(n - 1)$. This allocation rule does not maximize total utility when $f_i(t) > t_i$ for some i, but this is not essential for what follows. We can use Theorems 6 and 9 to address implementability and revenue equivalence. First, since the outcome is independent of the reports of agents $2, \ldots, n$, their type graphs have arcs of length zero, and truthfulness is trivially a dominant strategy for them as long as their payment is constant over all types. Now focus on agent 1, and let us drop index 1 in the following. Denote by $s < t$ two possible types in T_1, then the arc lengths in the type graph G_f for agent 1 are

$$\ell(s, t) = v(f(t), t) - v(f(s), t) = v(t, t) - v(s, t) = 0 - (s - t) = t - s > 0, \text{ and}$$
$$\ell(t, s) = v(f(s), s) - v(f(t), s) = v(s, s) - v(t, s) = 0 - 0 = 0.$$

As all arc lengths in all type graphs have nonnegative length, f is implementable by Theorem 6. Moreover, for any three types $s < q < t$ we have that $\ell(s, q) + \ell(q, t) = (q - s) + (t - q) = t - s = \ell(s, t)$; see Fig. 2.

It follows that the length of *any* (s, t)-path is at least $t - s$, and hence $dist_{G_f}(s, t) = \ell(s, t) = t - s$. Therefore, $dist_{G_f}(s, t) + dist_{G_f}(t, s) > 0$, and by Theorem 9 allocation rule f does not qualify for revenue equivalence. Indeed, if we normalize the payments at $\pi_i(1) = 0$ for all agents i, then the maximal incentive

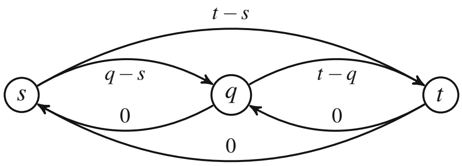

Fig. 2 Demand rationing: part of the type graph G_f for agent 1 for the dictatorial allocation rule

compatible payment for agent 1 is $\pi_1^+(t_1) = 0$ for all $t_1 \in (0, 1]$, and the minimal incentive compatible payment is $\pi_1^-(t_1) = t_1 - 1$ for all $t_1 \in (0, 1]$. These payments are exactly the negative of the shortest path lengths from type $t_{1_0} = 1$ to all other types t_1 in the former case, and from types t_1 to type $t_{1_0} = 1$ in the latter case. (Payments for all other agents being constant and equal to zero.)

Another, more reasonable allocation rule for this demand rationing game is the *proportional allocation rule*: define outcome $a_i = f_i(t) := t_i / \sum_{k=1}^{n} t_k$ for all agents i. This allocation rule is implementable as well, and it turns out to have a unique payment rule by revenue equivalence. Just like for the dictatorial allocation rule, this can be shown by verifying cyclical monotonicity as well as the antisymmetry of distances in the corresponding type graphs, using Theorems 6 and 9. Even though this is basic, it turns out to be a bit tedious and we do not work it out here. The reason is that the description of arc lengths in the type graph involves several case distinctions, such as the case where the total demand of all agents except i exceeds 1, i.e. $\sum_{k \neq i} t_k \geq 1$, and the case where it does not.

The demand rationing example owes its particular interest to the fact that many known theorems about revenue equivalence remain silent on it. The reason is that these results usually give characterizations of type spaces T and/or valuations v that allow to conclude revenue equivalence for *all* implementable allocation rules. But here we have two implementable allocation rules, one of which fulfills revenue equivalence, while the other does not.

4.2 Uniqueness of the Vickrey Auction for Single-Item Auctions

Consider the single-item auction setting from Example 1, and recall the Vickrey auction: it collects a bid from each bidder, allocates the item to a bidder with highest bid, and charges this bidder a price equal to the highest bid of a loosing bidder. We here show that the Vickrey auction is in fact the unique incentive compatible single-item auction that allocates the item to the bidder with highest valuation.

Fix agent i and t_{-i}, and again let us drop index i for simplicity. Consider *any* implementable deterministic allocation rule f. By Theorem 7 the allocation rule must be monotone. This implies that f allocates the item to agent i either never, or

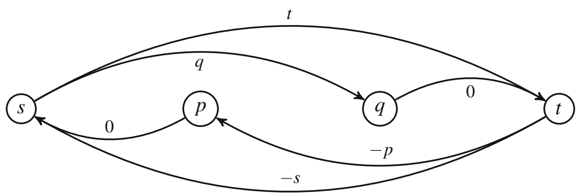

Fig. 3 Single-item auction: part of the type graph G_f for agent i, q being i's threshold value

always, or there exists a threshold type q, such that $f(t) = 0$ for $t < q$ and $f(t) = 1$ for $t > q$. Let us assume $f(q) = 1$. (The case $f(q) = 0$ works analogously.)

Let $s < t$ be two types of agent i. Note that $\ell(s, t) = 0$ if $t < q$ or $s \geq q$. Otherwise,

$$\ell(s, t) = t \quad \text{and} \quad \ell(t, s) = -s.$$

This implies that, for $s < q < t$, arc (s, t) with length t can be made shorter by replacing it by the two arcs (s, q), (q, t) with total length q. And any arc (t, s) can be made shorter by replacing it by the two arcs (t, p), (p, s), with $s < p < q$, with total length $-p$; see Fig. 3. Using this observation, we can argue that all cycles have nonnegative length. Therefore, by Theorem 6 every deterministic, monotone allocation rule f is implementable. By taking the limit $p \to q$, it also follows that

$$dist_{G_f}(s, t) = \begin{cases} 0 & \text{if } \ t < q\,, \\ q & \text{if } \ s < q \leq t\,, \\ 0 & \text{if } \ s \geq q\,, \end{cases} \quad \text{and} \quad dist_{G_f}(t, s) = \begin{cases} 0 & \text{if } \ t < q\,, \\ -q & \text{if } \ s < q \leq t\,, \\ 0 & \text{if } \ s \geq q\,. \end{cases}$$

In particular, we get $dist_{G_f}(s, t) = -dist_{G_f}(t, s)$. Hence, by Theorem 9, there is a unique payment rule for implementing f, up to constants.

It is a reasonable requirement that a bidder who does not win the item pays zero. This fixes the payment at bids strictly lower than q to zero. Now consider the allocation rule that assigns to the bidder with highest valuation, say i. It must have a threshold value q_i equal to the highest $t_j, j \neq i$, in other words the second highest bid. Revenue equivalence tells us that charging the winner the second highest bid is the only way of making this allocation rule dominant strategy incentive compatible.

Theorem 11 (Uniqueness of Vickrey auction) *The Vickrey auction is the unique direct, deterministic, dominant strategy incentive compatible auction that assigns the item to the bidder with highest valuation and charges losers nothing.*

5 Discussion and Literature

Characterizing implementable allocation rules by cyclical monotonicity as in Theorem 6 goes back to a paper by Rochet [10]; the term cyclical monotonicity is from convex analysis [11]. The graph-theoretic view and characterization of implementable allocation rules was first developed by Gui et al. [5], and has been fully explored by Vohra in [13]. There is a series of papers that study situations where the necessary weak monotonicity condition of Theorem 7 suffices to guarantee implementability, going far beyond the single-item auction setting illustrated in Sect. 4. Saks and Yu [12] proved it for convex type spaces, extending partial results by Bikhchandani et al. [3] and [5]. Berger et al. [2] further generalized these results. For implementation in Bayes–Nash equilibrium with single-dimensional type spaces this result goes back to Myerson [9]. Ashlagi et al. [1] have shown that convexity of the type space is essentially necessary. The demand rationing problem in Example 3 is from [4], and the valuation functions also appear in [7]. For the revelation principle in mechanism design, and also a version of the revenue equivalence theorem, see the seminal paper by Myerson on the design of optimal auctions [9]. The graph-theoretic characterization of revenue equivalence presented in Theorem 9 is taken from Heydenreich et al. [6]. Kos and Messner [8] extended this result by identifying the lattice structure of incentive compatible payments. Theorem 11 can be extended to many multi-dimensional domains, showing that any implementable allocation rule satisfies revenue equivalence and thereby proving that the multi-dimensional generalization of the Vickrey auction, called the Vickrey–Clarke–Groves (VCG) mechanism, is the only way to implement the utilitarian maximizer. The graph-theoretic approach yields elegant proofs of such results.

References

1. Ashlagi, I., Braverman, M., Hassidim, A., Monderer, D.: Monotonicity and implementability. Econometrica **78**(5), 1749–1772 (2010)
2. Berger, A., Müller, R., Naeemi, S.H.: Characterizing implementable allocation rules in multidimensional environments. Memorandum RM/14/021, Maastricht University (2014)
3. Bikhchandani, S., Chatterji, S., Lavi, R., Mu'alem, A., Nisan, N., Sen, A.: Weak monotonicity characterizes deterministic dominant-strategy implementation. Econometrica **74**(4), 1109–1132 (2006)
4. Cachon, G., Lariviere, M.: Capacity choice and allocation: strategic behavior and supply chain performance. Manag. Sci. **45**(8), 1091–1108 (1999)
5. Gui, H., Müller, R., Vohra, R.: Dominant strategy mechanisms with multidimensional types. Discussion Paper 1392, Northwestern University (2004)
6. Heydenreich, B., Müller, R., Uetz, M., Vohra, R.: Characterization of revenue equivalence. Econometrica **77**(1), 307–316 (2009)
7. Holmström, B.: Groves' scheme on restricted domains. Econometrica **47**(5), 1137–1144 (1979)
8. Kos, N., Messner, M.: Extremal incentive compatible transfers. J. Econ. Theory **148**(1), 134–164 (2012)
9. Myerson, R.B.: Optimal auction design. Math. Oper. Res. **6**(1), 58–73 (1981)
10. Rochet, J.-C.: A necessary and sufficient condition for rationalizability in a quasi-linear context. J. Math. Econ. **16**(2), 191–200 (1987)
11. Rockafellar, R.T.: Convex Analysis. Princeton University Press, Princeton (1970)
12. Saks, M., Yu, L.: Weak monotonicity suffices for truthfulness on convex domains. In: Proceedings of the 6th ACM Conference on Electronic Commerce, pp. 286–293. ACM (2005)
13. Vohra, R.: Mechanism Design: A Linear Programming Approach. Econometric Society Monographs. Cambridge University Press, Cambridge (2011)