Journal of Scheduling manuscript No.
(will be inserted by the editor)

On Optimal Mechanism Design for a Sequencing Problem

Jelle Duives · Birgit Heydenreich · Debasis Mishra · Rudolf Müller · Marc Uetz

Received: date / Accepted: date

Abstract We study mechanism design for a single server setting where jobs require compensation for waiting, while waiting cost is private information to the jobs. With given priors on the private information of jobs, we aim to find a Bayes-Nash incentive compatible mechanism that minimizes the total expected payments to the jobs. Following earlier work in the auction literature, we show that this problem is solved, in polynomial time, by a version of Smith's rule. When both waiting cost and processing times are private, we show that optimal mechanisms generally do not satisfy an independence condition known as IIA, and conclude that a closed form for optimal mechanisms is generally not conceivable.

Keywords Auction · Mechanism · Scheduling · Economics · Combinatorial Optimization

1 Introduction

The design of optimal mechanisms was first studied by Myerson [22] for single item auctions, and constitutes an intriguing problem in auction theory. We here study the design of optimal mechanisms for one of the classic machine scheduling problems, namely single machine scheduling to minimize the total weighted completion time. The setting comprises job-agents $j \in J = \{1, \dots, n\}$ that compete for being processed by a service provider who can handle one job at a time. No job can be interrupted once started, and each job is characterized by processing time p_j and weight w_j. The latter represents job j's disutility for waiting one unit of time. It is well known that the total disutility of the jobs is minimized by a schedule known as Smith's rule: sequence jobs in order of non-increasing ratios of weight over processing time w_j/p_j [26]. As usual in service or maintenance contracts for example, we assume that the service provider needs to compensate jobs for waiting, while the job data is private and only known to the jobs themselves. What is assumed to be known are publicly known priors, that is, probability distributions for the private job data. The effect of the private information setting is that jobs may have incentives to be dishonest: For instance, they may want to pretend to have higher waiting cost in order to receive higher compensation payments. The problem that jobs are dishonest can be overcome by appropriately adjusting the compensation payment, e.g. by paying more than the actual disutility for waiting, but that could have the effect of excessive compensation payments. The optimal mechanism design problem is then to find a mechanism in which jobs are incentivized to behave truthfully using an appropriate scheduling rule and payment scheme, but at the same time minimizing the total (expected) payments that are made to the jobs.

Contribution. We consider two cases. In the *single dimensional* case, processing times of jobs p_j are public information and only the jobs' weights w_j are private. In the *two dimensional* case, both weight w_j and processing time p_j are private. The private information of a job is called a jobs type,

Birgit Heydenreich and Marc Uetz acknowledge support by the Dutch research foundation through NWO grant 2004/03545/MaGW 'Local Decisions in Decentralised Planning Environments'. An extended abstract of this paper appeared in the proceedings of the 4th International Workshop on Internet and Network Economics [10].

Jelle Duives, Marc Uetz (corresponding author)
University of Twente, Dept. Applied Mathematics, P.O. Box 217, 7500 AE Enschede, The Netherlands. E-mail: jelleduives@gmail.com, m.uetz@utwente.nl

Birgit Heydenreich, Rudolf Müller
Maastricht University, Quantitative Economics, P.O.Box 616, 6200 MD Maastricht, The Netherlands. E-mail: birgit.heydenreich@gmail.com, r.muller@maastrichtuniversity.nl

Debasis Mishra
Indian Statistical Institute, Planning Unit, 7, S.J.S. Sansanwal Marg, New Delhi - 110 016, India. E-mail: dmishra@isid.ac.in

denoted t_j, so that for the single dimensional case $t_j = w_j$ and in the two dimensional case $t_j = (w_j, p_j)$. In either of the two cases, we assume the type spaces of jobs to be discrete. This is a departure from the traditional literature on auctions [22], but it is not uncommon to work with discrete type spaces: Some recent progress in deriving optimal mechanisms for multi-dimensional settings assume the type space to be discrete [1,16,23,2,12]. The assumption of discrete type spaces allows us to formulate optimal mechanism design problems as integer linear programs, while "nothing of qualitative significance is lost in moving from a continuous to a discrete type space" [29].

For the single dimensional case, we largely follow earlier work in auction theory. More specifically, we use a graph theoretic interpretation of the so-called incentive compatibility constraints - as used e.g. by Rochet [25], Malakhov and Vohra [16], Müller, Perea, and Wolf [21], Lavi and Swamy [15], Heydenreich et al. [11] and others. Doing so, we can derive a closed formula for the optimal mechanism. The result is in line with known results for single dimensional mechanism design, e.g. Hartline and Karlin [9], namely that serving the jobs in the order of non-increasing ratios of 'virtual' job weights over processing times is optimal. This mechanism minimizes expected total payments in the standard Bayes-Nash setting, which is formally defined later. We also show that the same mechanism can even be implemented such that truthfulness is even a dominant strategy for all jobs. By simple instances, we further show that the optimal mechanism is not necessarily efficient, that is, in general it does not coincide with Smith's rule, but it does so if jobs are symmetric. Our instances also show that the relative difference in total payments of an optimal mechanism and an efficient mechanism can be arbitrarily high. In this part of the paper we show how to put techniques from the mechanism design literature to work for the specific scheduling problem at hand, and thereby obtain some qualitative insights into the nature of optimal mechanisms.

For the two dimensional case, our main result is that the optimal mechanism does not satisfy an independence condition which is known as 'independence of irrelevant alternatives (IIA)'. This we show by analyzing a corresponding instance, which has been found with integer linear programming techniques. From that instance we conclude that the optimal mechanism cannot be expressed in terms of virtual weights along the lines of the single dimensional case. In fact, any kind of priority based scheduling algorithm, e.g., scheduling using Smith's rule with virtual weights, where the virtual weights of a job depend only on the characteristics of that job itself, cannot be an optimal mechanism in general. We conclude that optimal mechanism design for the two-dimensional case is in fact more involved than optimal mechanism design for certain, two dimensional auction settings, as studied for instance by Malakhov and Vohra [16] or Pai and Vohra [23]. We also show that the optimal mechanism for the two dimensional case is not efficient, not even for symmetric jobs.

Related Work. Optimal mechanism design goes back to a seminal paper by Myerson [22]. He studies optimal mechanism design for single item auctions and continuous, single dimensional type spaces. The optimal auction in his setup is to award the object to a bidder who has the highest *virtual valuation*, provided this virtual valuation is non-negative. In the symmetric case, this turns out to be the celebrated Vickrey auction, but augmented with a so-called reserve price. More generally, for single parameter agents the optimal auction is the one that maximizes the total virtual surplus [9].

Our work can be seen as analyzing how far the scheduling problem parallels the auction case. In that sense we follow Myerson's approach, but with discrete type spaces and using a graph-theoretic approach to compute the optimal mechanism. Hence it is no surprise that we observe close similarities to the auction case, up to some differences which are mainly due to the specific problem setting and discrete type spaces.

Concerning multi-dimensional mechanism design, Malakhov and Vohra [16] derive optimal mechanisms for an auction setting with discrete two dimensional type spaces. They consider a multi-unit model where every bidder has a capacity constraint, and the marginal value per unit and capacity are the private type of the bidder. The derived optimal mechanism also employs the efficient allocation rule with respect to 'virtual' types. Contrasting their work, we show that for two dimensional type spaces, the same graph-theoretic approach must fail to determine an optimal mechanism for the scheduling problem. This follows from the fact that an optimal mechanism is in general not IIA.

The fact that optimal mechanism design with multi-dimensional types is harder than with single dimensional types is not new. For example, Armstrong [1] studies a multi-object auction model where valuations are additive and drawn from a binary distribution (i.e., high or low). He gives optimal auctions under specific conditions. It becomes evident from his work that optimal mechanism design with multi-dimensional, discrete types is indeed difficult.

For makespan scheduling on unrelated machines, also Lavi and Swamy [15] show how to exploit binary distributions (i.e., high or low) for a multi dimensional mechanism design problem, and using Rochet's graph-theoretic characterization of implementability, there called cycle monotonicity, obtain mechanisms with constant factor approximation guarantees. Recent work has also shown that optimal multi-dimensional mechanism design can in some cases be done computationally efficient, yet at the cost of allowing randomization, e.g. [2,12]. In particular, inspired by the conference publication of this work [10], Hoeksma and Uetz [12] show how to compute an optimal randomized mechanism for the two dimensional problem considered in Section 4 of this paper, using linear programming techniques.

More generally, scheduling models have been looked at from different game theoretic perspectives, both in the Economic and Operations Research literature. There are some papers which are closely related to ours with respect to the model considered, but each with a different flavor when it comes to the game theoretic models. For example, Mitra [18] analyzes efficient and budget balanced mechanism design in a single dimensional queueing model, and Kittsteiner and Moldovanu [13] consider a model in which jobs arrive stochastically, and processing time is private information. Moulin [20] derives mechanisms that prevent merging and splitting of jobs. Suijs [27] discusses the same sequencing model as ours, and derives results on the existence of payment schemes that are required to be budget balanced. The same problem is discussed from the perspective of cost sharing by Curiel et al. [4], and later for m machines by Hamers et al. [8]. Also Hain and Mitra [7] analyze a sequencing problem with private information on processing times, give conditions for cost functions that allow implementability, and characterize optimal mechanisms as generalized VCG mechanisms. This list could be continued, yet none of the papers addresses the problem setting discussed here.

2 Notation and Preliminaries

Recall that $J=\{1,\dots,n\}$ is the set of jobs with weights w_j and processing times p_j. The private information of a job j is denoted its type t_j, so that $t_j=w_j$ in the single dimensional case and $t_j=(w_j,p_j)$ in the two dimensional case. Jobs as well as the mechanism designer share a common belief about other jobs' types in terms of probability distributions. We assume $T_j=\{t_j^1,\dots,t_j^{m_j}\}$ are the possible discrete types of job j. For the single dimensional case, we assume types are numbered so that $t_j^1=w_j^1<\dots<t_j^{m_j}=w_j^{m_j}$. Let $\phi_j(t_j)$ denote the probability for job j having type $t_j\in T_j$. Probability distributions $(\phi_j)_{j\in J}$ and type spaces $(T_j)_{j\in J}$ are public information. We assume that types are independent across jobs. Let us denote by $T=T_1\times\dots\times T_n$ the set of all type profiles. Let ϕ be the joint probability distribution of $t=(t_1,\dots,t_n)\in T$. Then by independence, $\phi(t)=\Pi_{j\in J}\phi_j(t_j)$. Let t_{-j}, T_{-j} and ϕ_{-j} be defined as usual to denote type profile, type space and type distribution for all jobs except j, so that (t_j,t_{-j}) is the type profile where job j has type t_j and the types of other jobs are t_{-j}.

In mechanism design, a direct revelation mechanism consists of an allocation rule f and a payment scheme π^1. In our setting, jobs report their (possibly false) types $t=(t_1,\dots,t_n)$, and depending on those reported types t, the allocation rule is nothing but a schedule $f(t)$. Next to that there is a vector of payments $\pi(t)$ that assigns a payment to every job in order to compensate them for their waiting. The restriction to direct revelation mechanisms, that is mechanisms in which the only action of the jobs is to report their types, is not arbitrary: By Myerson's revelation principle [22], the restriction to direct revelation mechanisms is no loss of generality.

If in a given schedule job j has waiting time S_j and actual weight w_j, it encounters a valuation for the schedule which equals the actual cost for waiting, $-w_jS_j$. On top of that, job j receives payment π_j, and therefore it's total utility equals $\pi_j-w_jS_j$. This is a standard model known as quasilinear utilities. Given a mechanism (f,π), in the Bayes-Nash model the assumption is that each job's goal is to maximize its expected utility, where the expectation is to be taken over all possible types of other jobs. To that end, we need to express the expected valuation of a job when reporting to be of type t_j, which is determined by the expected waiting time when reporting to be of type t_j. Formally, the expected waiting time of job j if it reports type t_j, and allocation rule f is in place, equals

$$ES_j(f,t_j):=\sum_{t_{-j}\in T_{-j}} S_j(f(t_j,t_{-j}))\phi_{-j}(t_{-j})\,.$$

Here, $S_j(f(t))$ denotes the start time of job j in schedule $f(t)$. Also, let $E\pi_j(t_j):=\sum_{t_{-j}\in T_{-j}}\pi_j(t_j,t_{-j})\phi_{-j}(t_{-j})$ be the expected payment to j if it reports type t_j. We are now prepared to give the standard definitions.

Definition 1 A mechanism (f,π) is *Bayes-Nash incentive compatible* (BIC) if truth telling is a weakly dominant strategy in expectation, so for every job j and every two types $t_j^i,t_j^k\in T_j$,

$$E\pi_j(t_j^i)-w_j^iES_j(f,t_j^i)\geq E\pi_j(t_j^k)-w_j^iES_j(f,t_j^k)\,. \tag{1}$$

Note that the expectation is taken under the assumption that all agents apart from j report truthfully. Indeed, this inequality exactly expresses the (Bayes-)Nash equilibrium concept: Under the assumption that no other job deviates from being truthful, job j's expected utility is maximal when being truthful, too. If for a given allocation rule f there exists a payment scheme π such that (f,π) is BIC, then f is called *Bayes-Nash implementable*. The payment scheme π is referred to as an *incentive compatible* payment scheme.

Next to Bayes-Nash incentive compatibility, a standard constraint that we impose as well is individual rationality, expressing the fact that expected utilities should be nonnegative. In auctions it can be interpreted as voluntary participation constraint, as with negative expected utility, a rational agent would not want to participate at all. A subtle difference in our setting is that we assume a priori that all

[1] In fact, the payment scheme π is dependent on the choice of f, so we should write π^f to indicate this dependence. But in order to avoid excessive notation, wherever possible we use π instead, keeping in mind that π is clearly not independent of f.

jobs must be scheduled. That means that individual rationality should be motivated from economic applications, like for example in services contracts where delays need to be reimbursed. Mathematically, individual rationality makes sure that the optimal mechanism design problem is bounded.

Definition 2 A mechanism (f,π) is (interim) *individually rational* (IR) if for every agent j and every type $t_j \in T_j$,

$$E\pi_j(t_j) - w_j ES_j(f,t_j) \geq 0\,. \tag{2}$$

Note that individual rationality only makes a claim about truthful jobs. For what follows, it will be convenient to ensure individual rationality by introducing a dummy type t_j^d for each job j, with probability $\phi(t_j^d) = 0$, and making sure that the dummy type gives zero utility to the job, by defining $ES_j(f,t_j^d) := 0$ and $E\pi_j(t_j^d) := 0$ for all jobs $j \in J$. Now, we impose constraints (1) also for $k = d$, which then implies (2). Therefore, the dummy types together with the mentioned assumptions guarantee that individual rationality is satisfied along with the incentive compatibility constraints. Sometimes, it will be convenient to use index $m_j + 1$ instead of d for the dummy type.

Our goal is to set up a mechanism that fulfills (1) and (2), and among all such mechanisms minimizes the expected total payment that has to be made to the jobs.

3 The Single Dimensional Setting

In this section, the processing times p_j of jobs are fixed and public. For convenience of notation, we suppress t_j and write w_j for the type of job j, W_j to denote its type space, and $W = W_1 \times \cdots \times W_n$. We first introduce some basic graph theoretic concepts that come handy in determining the optimal mechanism in the discrete setting considered here. The basic idea for this approach goes back to a paper by Rochet [25].

Define the *type graph* $T_j(f)$ for job j as a complete directed graph with node set W_j and an arc from any node w_j^i to any other node w_j^k of length

$$\ell_{ik} = w_j^i[ES_j(f,w_j^k) - ES_j(f,w_j^i)].$$

Note that ℓ_{ik} represents the gain in expected valuation for agent j by truthfully reporting type w_j^i instead of lying type w_j^k; it can be both positive or negative. Each node w_j^i also has one arc towards the dummy node, but the dummy node has no outgoing arcs.

The incentive constraints for a BIC mechanism (f,π) and job j can then be written as

$$\begin{aligned} E\pi_j(w_j^k) &\leq E\pi_j(w_j^i) + w_j^i[ES_j(f,w_j^k) - ES_j(f,w_j^i)] \\ &= E\pi_j(w_j^i) + \ell_{ik}\,. \end{aligned}$$

That is, the expected payments $E\pi_j(\cdot)$ constitute a node potential in digraph $T_j(f)$, and therefore Bayes-Nash implementability of an allocation rule f is equivalent to $T_j(f)$ having no negative length directed cycle; this observation implicitly appears already in [25], see also [21]. Furthermore, consider the length of a directed cycle consisting of only two arcs in $T_j(f)$, it equals

$$\begin{aligned} \ell_{ik} + \ell_{ki} = {} & w_j^i[ES_j(f,w_j^k) - ES_j(f,w_j^i)] \\ & + w_j^k[ES_j(f,w_j^i) - ES_j(f,w_j^k)] \\ = {} & (w_j^i - w_j^k)[ES_j(f,w_j^k) - ES_j(f,w_j^i)]\,. \end{aligned} \tag{3}$$

Next observe that the last term in (3) is non-negative for all jobs j and any two types w_j^i and w_j^k if and only if f fulfills the following monotonicity condition.

Definition 3 An allocation rule f satisfies monotonicity with respect to weights, or short *monotonicity*, if for every job j, $w_j^i < w_j^k$ implies that $ES_j(f,w_j^i) \geq ES_j(f,w_j^k)$.

Given this definition, the following characterization of Bayes-Nash implementability of an allocation rule f is a standard result in mechanism design with single dimensional types.

Theorem 1 *An allocation rule f is Bayes-Nash implementable if and only if it satisfies monotonicity with respect to weights.*

As to the proof, recall that implementability of f is equivalent to the fact that there are no negative length directed cycles in $T_j(f)$. Hence implementability implies that there are no negative length directed cycles with only two arcs, and monotonicity follows from (3). For the other direction, via (3) it follows from monotonicity that there are no negative length directed cycles with only two arcs in $T_j(f)$. Left to show is that there are no negative length directed cycles in $T_j(f)$. This follows from the fact that any cycle can be decomposed into two-cycles, and this only decreases the total length; the proof of this so-called *decomposition monotonicity* is given in the appendix.

3.1 An Illustrative Example

For those less familiar with the given notions in mechanism design, we briefly go through a small example to illustrate the preceding definitions. There are two jobs. Job 1 has processing time $p_1 = 1$ and two possible weights $w_1^1 = 1$ and $w_1^2 = 2$, both with probability $\phi(w_1^1) = \phi(w_1^2) = 1/2$. Job 2 has processing time $p_2 = 3$, and only one possible type, $w_2 = 4$. Assume the allocation rule f is Smith's rule [26], that is, sequencing the jobs in non-increasing order of ratios w_j/p_j. Since with Smith'r rule the start time of any job can only decrease with increasing weight, it is implementable by Theorem 1.

That said, if Job 1 has weight 2, Job 1 is scheduled first with waiting times $S_1 = 0$ and $S_2 = 1$, and if Job 1 has weight 1, it is scheduled second with waiting times $S_1 = 3$

and $S_2 = 0$. Therefore, given that Job 1 is truthful, the expected waiting time for Job 2 equals $ES_2 = 0.5$. For Job 2, since it has only one possible type, there are no incentive compatibility constraints, and individual rationality for Job 2 requires that

$$E\pi_2 - 4 \cdot \underbrace{ES_2}_{=0.5} \geq 0\,.$$

Hence any payment scheme with $E\pi_2 \geq 2$ is (interim) individual rational for Job 2; for instance one could define the payment scheme $\pi_2 = 0$ in case $w_1 = 1$ and $\pi_2 = 4$ in case $w_1 = 2$, which leaves Job 2 with nonnegative utility in all possible outcomes. But note that also the payment scheme $\pi_2 = 2$ in case $w_1 = 1$ and $\pi_2 = 2$ in case $w_1 = 2$ fulfills (interim) individual rationality, since we only require it to hold in expectation.

For Job 1 we get as expected waiting times $ES_1(w_1 = 1) = 3$ and $ES_1(w_1 = 2) = 0$. The Bayes-Nash incentive compatibility constraints for Job 1 are then

$$E\pi_1(w_1 = 1) - 1 \cdot \underbrace{ES_1(w_1 = 1)}_{=3} \geq E\pi_1(w_1 = 2) - 1 \cdot \underbrace{ES_1(w_1 = 2)}_{=0}, \quad \text{and}$$

$$E\pi_1(w_1 = 2) - 2 \cdot \underbrace{ES_1(w_1 = 2)}_{=0} \geq E\pi_1(w_1 = 1) - 2 \cdot \underbrace{ES_1(w_1 = 1)}_{=3}\,.$$

The individual rationality constraints are

$$E\pi_1(w_1 = 1) - 1 \cdot \underbrace{ES_1(w_1 = 1)}_{=3} \geq 0 \quad \text{and}$$

$$E\pi_1(w_1 = 2) - 2 \cdot \underbrace{ES_1(w_1 = 2)}_{=0} \geq 0\,.$$

Then $E\pi_1(w_1 = 1) = 3$ and $E\pi_1(w_1 = 2) = 0$ is a possible payment scheme that is both Bayes-Nash incentive compatible and individual rational for Job 1. (Note, here we have the special case that $E\pi_1(\cdot) = \pi_1(\cdot)$ for Job 1, because Job 2 has just one possible type, hence the expectation over the types of other jobs is trivial for Job 1.)

Finally, the type graph $T_1(f)$ for Job 1 for the given allocation rule f is depicted in Figure 1. Note that, indeed, it has no negative length directed cycle. Payment schemes that implement f are exactly node potentials in $T_1(f)$, that is, expected payments for the possible types of Job 1 that fulfill the triangle inequality along all arcs of $T_1(f)$. As explained in the next section, the minimal incentive compatible payments for a given allocation rule f can easily be computed via shortest paths in the type graphs $T_j(f)$.

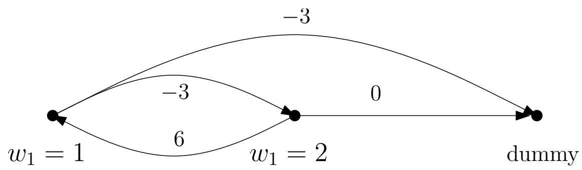

Fig. 1 Type graph $T_1(f)$ for Job 1 with two possible weights; f being Smith's rule.

3.2 Optimal Mechanisms

Knowing that incentive compatible payment schemes must yield node potentials in the type graphs $T_j(f)$, we seek to find the node potentials that minimize the total expected payments. In fact, a lower bound for the expected payment $E\pi_j(w_j^i)$ for type w_j^i is found by taking the negative of the shortest path length from node w_j^i to dummy node w_j^d in the type graph $T_j(f)$. To see why, let $P = (w_j^i = a_0, a_1, \ldots, a_m = w_j^d)$ denote some directed path from w_j^i to w_j^d in the type graph $T_j(f)$ for job j. Denote by $\ell(P)$ its length. Let (f,π) be a Bayes-Nash incentive compatible mechanism. Adding up the incentive constraints

$$\begin{aligned} E\pi_j(a_i) &\leq E\pi_j(a_{i-1}) + a_{i-1}[ES_j(f,a_i) - ES_j(f,a_{i-1})] \\ &= E\pi_j(a_{i-1}) + \ell_{a_{i-1}a_i} \end{aligned}$$

for $i = 1,\ldots,m$ yields $E\pi_j(w_j^d) \leq E\pi_j(w_j^i) + \ell(P)$. By definition, we have $E\pi_j(w_j^d) = 0$, so this is equivalent to

$$E\pi_j(w_j^i) \geq -\ell(P)\,. \tag{3}$$

Now recall that for any implementable allocation rule f, $T_j(f)$ has no negative length directed cycle. Hence, by (3), for given f we can compute minimal incentive compatible payments $\pi_j(w_j^i)$ by computing shortest paths in $T_j(f)$. The following lemma specifies the resulting minimal payments $\pi (= \pi^f)$ for any implementable allocation rule f.

Lemma 1 *For a Bayes-Nash implementable allocation rule f, the payment scheme defined by $\pi_j(w_j^d) = 0$, and for $i = 1,\ldots,m_j$,*

$$\pi_j(w_j^i) = \sum_{k=i}^{m_j} w_j^k [ES_j(f,w_j^k) - ES_j(f,w_j^{k+1})] \tag{4}$$

is incentive compatible, (interim) individually rational and minimizes the expected total payment made to the jobs. Here, recall that $m_j + 1 = d$. The corresponding expected total payment is given by

$$\begin{aligned} P^{min}(f) &= \sum_{j\in J}\sum_{i=1}^{m_j} \phi_j(w_j^i) \sum_{k=i}^{m_j} w_j^k \left[ES_j(f,w_j^k) - ES_j(f,w_j^{k+1})\right] \\ &= \sum_{j\in J}\sum_{w_j\in W_j} \phi_j(w_j)\overline{w}_j ES_j(f,w_j), \end{aligned}$$

where the virtual weights $\overline{w}_j$ are defined by $\overline{w}_j^1 = w_j^1$ and

$$\overline{w}_j^i = w_j^i + (w_j^i - w_j^{i-1}) \frac{\sum_{k=1}^{i-1} \phi(w_j^k)}{\phi_j(w_j^i)} \quad \textit{for } i = 2,\ldots,m_j\,.$$

Observe that for each job j and type w_j^i we define one payment $\pi_j(w_j^i)$. In particular this is independent of the reports of other jobs w_{-j}. It is one and the same payment that job j receives when reporting w_j^i, independent of w_{-j}, so that $\pi_j(w_j^i, w_{-j}) = \pi_j(w_j^i) = E\pi_j(w_j^i)$, for all w_{-j}. Incentive compatibility and individual rationality hold in expectation, but not necessarily for all w_{-j}. In Section 3.5 we will argue that the same expected payments can be achieved by defining a payment scheme that is even incentive compatible and indinvidually rational for any type report of other jobs w_{-j}.

The formal proof of the above lemma and the corresponding formulae is contained in the appendix. That proof can be seen as an instantiation of the standard payment scheme for single dimensional mechanism design problems, here for a scheduling setting and with discrete types[2]. We remark that that shortest paths in the type graphs $T_j(f)$ are in fact independent of the (implementable) allocation rule f, which allows us to obtain the above closed formulae for minimal (expected) payments. This property generally gets lost once we move to multidimensional types in Section 4, where we indeed show that this approach fails.

An intuitive interpretation for the virtual weights in (4) is not that straightforward. We have that $\overline{w}_j \geq w_j$, and roughly speaking, the virtual weight $\overline{w}_j$ is large if the probability $\phi(w_j)$ is small. For example, consider a job with weights $w_j = 1,2,3$, or 4 with probabilities 1/4 each, then virtual weights are $\overline{w}_j = 1,3,5$, and 7. When the probabilities are $1/2, 1/4, 3/16$ and $1/16$, virtual weights become $\overline{w}_j = 1,4,7$, and 19. In both cases the mapping $w_j \mapsto \overline{w}_j$ is monotone, but this is not true in general: When the probabilities are $1/8, 1/8, 1/8$, and $5/8$, virtual weights become $\overline{w}_j = 1,3,5$, and 4.6.

Given the minimum payments for every (implementable) allocation rule f, we next want to identify an allocation rule f that indeed minimizes $P^{min}(f)$ among all Bayes-Nash implementable and individually rational allocation rules. For the time being, let us impose the following regularity condition that ensures Bayes-Nash implementability of the allocation rule in our candidate mechanism. We will get rid of it afterwards using standard techniques.

Definition 4 *Regularity* is satisfied if for every job j and $i = 1,\dots,m_j - 1$, we have $\overline{w}_j^i < \overline{w}_j^k$ whenever $w_j^i < w_j^k$.

Recall that, by the above example, regularity need not be satisfied in general. But it is always satisfied, for instance if the differences $w_j^i - w_j^{i-1}$ are non-decreasing and the probabilities are non-increasing in i, so especially for uniform distributions over a contiguous domain of weights.

Theorem 2 *Let the virtual weights $\overline{w}_j$, $j \in J$, and payments π be defined as in Lemma 1. Let f be the allocation rule that schedules jobs in non-increasing order of ratios $\overline{w}_j/p_j$. If regularity holds, then (f,π) is a mechanism that minimizes the total expected costs among all individual rational, Bayes-Nash implementable mechanisms.*

Proof We show that f is Bayes-Nash implementable and minimizes $P^{min}(f)$ among all Bayes-Nash implementable allocation rules. For any allocation rule f, we can rewrite $P^{min}(f)$ as follows, using independence of weights.

$$\begin{aligned}
P^{min}(f) &= \sum_{j\in J}\sum_{w_j\in W_j} \phi_j(w_j)\overline{w}_j ES_j(f,w_j) \\
&= \sum_{j\in J}\sum_{w_j\in W_j} \phi_j(w_j)\overline{w}_j \sum_{w_{-j}\in W_{-j}} S_j(f(w_j,w_{-j}))\phi_{-j}(w_{-j}) \\
&= \sum_{j\in J}\sum_{(w_j,w_{-j})\in W} \phi(w_j,\, w_{-j})\overline{w}_j S_j(f(w_j,w_{-j})) \\
&= \sum_{w\in W}\phi(w)\sum_{j\in J}\overline{w}_j S_j(f(w)).
\end{aligned}$$

Thus, $P^{min}(f)$ can be minimized by point wise minimizing $\sum_{j\in J}\overline{w}_j S_j(f(w))$ for every reported type profile w. This is achieved by scheduling the jobs in order of non-increasing ratios $\overline{w}_j/p_j$. But in order for this proof to hold, we need to argue that the so-defined allocation rule is indeed monotone. To that end, observe that the expected start time $ES_j(w_j)$ is clearly non-increasing in the virtual weight $\overline{w}_j$. The regularity condition ensures that it is non-increasing also in the original weights w_j. □

3.3 Discussion of the Result

It is well known that scheduling in order of non-increasing weight over processing time ratios w_j/p_j minimizes the sum of weighted start times $\sum_{j=1}^n w_j S_j$, and therefore maximizes the total valuation of all jobs. This allocation rule, known as Smith's rule [26], is therefore the efficient allocation rule in our setting. We next compare the optimal allocation rule with the efficient allocation rule.

Definition 5 Jobs are called *symmetric* if $W_1 = \ldots = W_n$, $\phi_1 = \ldots = \phi_n$ and $p_1 = \ldots = p_n$.

Theorem 3 *If jobs are symmetric and regularity holds, then the optimal mechanism is the efficient one, that is, Smith's rule.*

Proof By symmetry, for any two jobs j and k the virtual weights are equal, i.e. $\overline{w}_j^i = \overline{w}_k^i$. By regularity, virtual weights are non-decreasing in the original weights, and as all p_j are

[2] For the setting considered here, due to the discrete type space, the revenue equivalence theorem does not hold, which means there are different payment schemes that implement a given allocation rule f. Yet if we were to approximate a continuous type distribution T by a convergent series of distributions on discrete type spaces T_k, the minimal and maximal payments that implement a given f for T_k will converge to the unique payment scheme of f for T. Uniqueness of the latter follows by the revenue equivalence theorem, e.g. using [11], for the setting considered.

equal, scheduling jobs in order of their non-increasing ratios w_j/p_j is equivalent to scheduling them in order of non-increasing ratios $\overline{w}_j/p_j$. □

Notice that this is no longer true in the non-regular case, as there is positive probability that the optimal mechanism does not schedule the jobs in order of non-increasing ratios w_j/p_j. If weight distributions may differ across jobs, or if jobs have different processing times, then the optimal mechanism is in general not efficient either. In fact, the minimal total expected payment for implementing Smith's rule can be arbitrarily bad in comparison to the optimal allocation rule. This is illustrated by the following two examples.

Example 1 Let there be two jobs 1 and 2 with $W_1 = \{M+1\}$ and $W_2 = \{1,M\}$ for some constant $M > 2$. Let $\phi_2(1) = 1-1/M$, $\phi_2(M) = 1/M$ and $p_1 = p_2 = 1$. Let f^e be the efficient and f^* be the optimal allocation rule. Then the ratio $P^{min}(f^e)/P^{min}(f^*)$ goes to infinity as M goes to infinity.

Proof The efficient allocation rule, Smith's rule, always allocates job 1 first. So the optimal payment for Smith's rule is to pay 0 to job 1 and to pay M to job 2, irrespective of its type. The minimum expected total payment is hence $P^{min}(f^e) = M$. For the optimal allocation, we compute virtual weights after Lemma 1: $\overline{w}_1^1 = w_1^1 = M+1$, $\overline{w}_2^1 = w_2^1 = 1$ and $\overline{w}_2^2 = M+(M-1)(1-1/M)/(1/M) = M^2-M+1$. The latter is larger than $M+1$ if $M>2$. Therefore, job 2 is scheduled in front of job 1 if it has weight M and behind it otherwise. The expected start times for job 2 are $ES_2(f^*,1) = 1$ and $ES_2(f^*,M) = 0$, respectively. Optimal payments according to Lemma 1 are $\pi_2^{f^*}(1) = 1$ and $\pi_2^{f^*}(M) = 0$. For job 1, the expected start time is $ES_1(f^*,M+1) = 1/M$ and the expected payment $\pi_1^{f^*}(M+1) = 1+1/M$. Hence, $P^{min}(f^*) = 1+1/M+1\cdot(1-1/M) = 2$, and $P^{min}(f^e)/P^{min}(f^*) = M/2 \to \infty$ for $M \to \infty$. ◁

Example 2 Let there be two jobs 1 and 2 with the same weight distribution $W_1 = W_2 = \{1,M\}$, $\phi_j(1) = 1-1/M$, $\phi_j(M) = 1/M$ for $j = 1,2$. Let $p_1 = 1/2$ and $p_2 = M/2+1$ for some $M > 2$. Let f^e be the efficient and f^* be the optimal allocation rule. Then the ratio $P^{min}(f^e)/P^{min}(f^*)$ goes to infinity as M goes to infinity.

Proof The efficient allocation rule always schedules job 1 first, since $1/(1/2) = 2 > 2M/(M+2) = M/(M/2+1)$. Therefore, the expected start time of job 1 is 0 and that of job 2 is 1/2. Optimal payments according to Lemma 1 are $\pi_1^{f^e}(1) = \pi_1^{f^e}(M) = 0$ and $\pi_2^{f^e}(1) = \pi_2^{f^e}(M) = M/2$. Hence, $P^{min}(f^e) = M/2$. For the optimal mechanism, we compute virtual weights as $\overline{w}_1^1 = \overline{w}_2^1 = 1$ and $\overline{w}_1^2 = \overline{w}_2^2 = M^2-M+1$. The schedule in the optimal mechanism is determined by taking the ratio of virtual weight and processing time of each job, and scheduling the job with the larger weight first (breaking ties in favor of job 1). The resulting expected start times and payments are given below:

$$\begin{aligned} &ES_1(f^*,1) = 1/2+1/M && \pi_1^{f^*}(1) = 1/2+1/M \\ &ES_1(f^*,M) = 0 && \pi_1^{f^*}(M) = 0 \\ &ES_2(f^*,1) = 1/2 && \pi_2^{f^*}(1) = 1-1/(2M) \\ &ES_2(f^*,M) = 1/(2M) && \pi_2^{f^*}(M) = 1/2. \end{aligned}$$

Hence,

$$\begin{aligned} P^{min}(f^*) &= (\frac{1}{2}+\frac{1}{M})(1-\frac{1}{M}) + (1-\frac{1}{2M})(1-\frac{1}{M}) + \frac{1}{2}\cdot\frac{1}{M} \\ &= (1-\frac{1}{M})(\frac{3}{2}+\frac{1}{2M}) + \frac{1}{2}\cdot\frac{1}{M}. \end{aligned}$$

Thus, $P^{min}(f^e)/P^{min}(f^*) \to \infty$ as $M \to \infty$. ◁

It is finally instructive to comment on the comparison of the optimal mechanism with the VCG mechanism. The VCG mechanism, named after Vickrey [28], Clarke [3] and Groves [6], is the following. The allocation rule is the efficient one, that is scheduling in order of non-increasing ratios w_j/p_j. Moreover, the incentive compatible payment scheme is based on computing the "harm" that a job's presence causes to other jobs. Taking into account individual rationality, for a reported type profile $w = (w_1,\dots,w_n)$, one calculates the minimal VCG payments as

$$\pi_j^{VCG}(w) = p_j \sum_{k\in J:k<j} w_k\,,$$

where we assume w.l.o.g. that $w_1/p_1 \le \cdots \le w_n/p_n$. As illustrated by Examples 1 and 2, the allocation rule of the VCG mechanism can generally differ from the allocation of the optimal mechanism. Moreover, the following example shows that, even for symmetric jobs where the allocation rule is the same, the payments for the VCG mechanism can be a factor 3/2 off the payments for the optimal mechanism, even with only two jobs[3].

Example 3 There are two symmetric jobs with $W_1 = W_2 = \{w^1,w^2\}$, $w^1 < w^2$, and $\phi_j(w^1) = \phi_j(w^2) = 1/2$ for $j = 1,2$. Processing times are $p_1 = p_2 = 1$. The total expected payment for the VCG mechanism can be a factor 3/2 higher compared to the optimal mechanism.

Proof First observe that by symmetry of the jobs, and since regularity holds, the allocation rule of both mechanisms is Smith's rule. There are four possible type profiles, each occurring with probability 1/4, namely (w^1,w^1), (w^1,w^2), (w^2,w^1), and (w^2,w^2). The resulting schedules are the same for the VCG and the optimal mechanism (breaking ties arbitrarily). The VCG mechanism pays to the job that is scheduled

[3] The primary driver for this example is the discrete type space. If type spaces were continuous, in the symmetric case the revenue equivalence theorem yields that payments in the VCG mechanism and the optimal mechanism are identical, because the allocation rule is the same.

last the weight of the job that is scheduled before. Thus, the VCG mechanism has to spend w^1 for type profile (w^1, w^1) and w^2 for the three remaining type profiles. The total expected payment for the VCG mechanism is thus $(3w^2 + w^1)/4$. Let (f, π^f) denote the optimal mechanism from Section3. In the optimal mechanism, the expected payment to a job with weight w^1 is equal to $E\pi_j^f(w^1) = w^1[ES_j(f,w^1) - ES_j(f,w^2)] + w^2 ES_j(f,w^2) = w^1[3/4-1/4] + w^2[1/4] = w^1/2 + w^2/4$. The expected payment to a job with weight w^2 is $E\pi_j^f(w^2) = w^2 ES_j(f,w^2) = w^2/4$. The total expected payment for the optimal mechanism is thus $2 \cdot 1/2 \cdot (w^1/2 + w^2/4 + w^2/4) = (w^1 + w^2)/2$. The claim follows when letting $w^1 = 0$. $\triangleleft$

3.4 The Non-Regular Case

We needed the regularity condition because we require the allocation rule of Theorem 2 to be monotone. In order to extend the optimal mechanism to the non-regular case, we can apply a standard procedure known as 'ironing' which was already proposed by Myerson in [22]. Applied to the scheduling problem it means that we 'iron' the possibly non-monotone mapping $w_j \mapsto \overline{w}_j$ at any interval of non-monotonicity by 'flattening' the mapping through adapting some of the virtual weights $\overline{w}_j$. This is explained in the following.

Indeed, we simply do the following, if necessary recursively: Consider any non-monotone subsequence of virtual weights $I_j^{qr} := \{\overline{w}_j^q, \ldots, \overline{w}_j^r\}$. Define shorthand notation $\phi_j^i := \phi_j(w_j^i)$ and let $\overline{w}_j^{qr} := (\sum_{i=q}^r \phi_j^i \overline{w}_j^i)/\sum_{i=q}^r \phi_j^i$ be a new virtual weight that replaces the virtual weights $\overline{w}_j^i$ with $i \in I_j^{qr}$. The new allocation rule assigns to job j with a report w_j^i, $q \le i \le r$, the virtual weight $\overline{w}_j^{qr}$ (equivalently, $\overline{w}_j^l$, $q \le l \le r$, with probability $\phi_j^l/\sum_{i=q}^r \phi_j^i$). Note that this does not change the expected start time for all other jobs, while the (expected) virtual weights, and hence expected start time of job j is now monotone in w_j^i – in fact constant for all $i \in I_j^{qr}$. This way we obtain a monotone allocation rule. The payments from Lemma 1 are still optimal incentive compatible payments for the new allocation rule f, and that the formula for $P^{min}(f)$ in the proof of Theorem 2 remains valid, too. Thereby, Smith's rule based on ironed virtual weights yields again the optimal mechanism. We do not go into further technical details, and state the corresponding result somewhat informally.

Theorem 4 *Let the virtual weights $\overline{w}_j$, $j \in J$, and payments π be as defined in Lemma 1. Let f be the allocation rule that first irons the mappings $w_j \mapsto \overline{w}_j$ as suggested by the ironing procedure described above, and then schedules jobs in non-increasing order of ratios $\overline{w}_j/p_j$. Then (f, π) is an optimal mechanism.*

3.5 Implementation in Dominant Strategies

So far, we have discussed implementability according to Bayes-Nash equilibrium from Definition 1, and for a given report w_j^i of job j, waiting time as well as payment are expected values, the expectation taken over truthful reports of the other jobs. It is an important question in mechanism design to ask if a given allocation rule can also be implemented with respect to the stronger, dominant strategy equilibrium, with the same expected utilities for jobs. We refer for example to [19, 17, 5] for results in that direction. Here we briefly argue that this is indeed possible for our setting.

Definition 6 A mechanism (f, π) is *dominant strategy incentive compatible* (DSIC) if for every job j and every two types w_j^i, $w_j^k \in W_j$, and any report w_{-j} of other jobs,

$$\pi_j(w_j^i) - w_j^i S_j(f,(w_j^i, w_{-j})) \ge \pi_j(w_j^k) - w_j^i S_j(f,(w_j^k, w_{-j})) \,. \tag{5}$$

If for allocation rule f there exists a payment scheme π such that (f, π) is DSIC, then f is called dominant strategy implementable. Mechanism (f, π) is ex-post individual rational if

$$\pi_j(w_j^i) - w_j^i S_j(f,(w_j^i, w_{-j})) \ge 0$$

for any report w_{-j} of other jobs.

Clearly, dominant strategy implementability implies Bayes-Nash implementability. The definition of monotonicity, and the fact that implementability is equivalent with monotonicity, translate correspondingly, only replacing the expected waiting time $ES_j(f, w_j)$ by the waiting time $S_j(f,(w_j, w_{-j}))$, for all w_{-j}. Mookherjee and Reichelstein [19] identify a "single crossing" property for agents' valuation functions that guarantees any Bayes-Nash implementable allocation rule f to be implementable in dominant strategies at the same expected payment per agent. Except for the fact that we have discrete and not continuous type spaces, and therefore do not have uniqueness of the payment scheme as they do, the scheduling problem considered here can be shown to fall into the category of problems they address, and therefore their proof could be adapted. More recently, based on Manelli and Vincent's [17] idea to ask for equivalence of only expected allocations and payments, Gershkov et al. derive a more general result on equivalence of Bayes Nash and dominant strategy implementation that can be used to show that there exists an equivalent dominant strategy implementation of our optimal Bayes Nash mechanism; see [5, Thm. 2]. For the sake of completeness, we nevertheless give the simple, direct proof of Theorem 5 in the appendix.

Theorem 5 (See also [19] and [5]) *The allocation rule of the optimal mechanism from Theorem 2 is implementable in dominant strategies with the same expected utility per job and the same total expected cost.*

4 The Two Dimensional Setting

Now weight w_j and processing time p_j of a job are private information of the job. Recall that job j's type is then $t_j = (w_j, p_j)$. For convenience, let us assume the types $t_j \in T_j$ are as follows. $t_j = (w_j, p_j) \in W_j \times P_j$, where $W_j = \{w_j^1, \dots, w_j^{m_j}\}$ with $w_j^1 < \dots < w_j^{m_j}$ and $P_j = \{p_j^1, \dots, p_j^{q_j}\}$ with $p_j^1 < \dots < p_j^{q_j}$, and $\phi(t_j)$ being the probability of type $t_j = (w_j, p_j)$. It will also be convenient to identify $(w_j^{m_j+1}, p_j^k)$ with the dummy type t_j^d for any $k = 1, \dots, q_j$. We assume that a job can only report a processing time that is not lower than the true processing time, and that a job is processed for his reported processing time. We believe this is a natural assumption, as reporting a shorter processing time can be easily punished by preempting the job after the declared processing time, before it is actually finished. This is sometimes referred to as verifiability, e.g. [24]. By this assumption, the incentive compatibility constraints (1) need to hold only for types t_j^i and t_j^k with $p_j^i \leq p_j^k$, as jobs cannot understate their true processing time.

4.1 Bayes-Nash Implementability

In the two dimensional setting, the type graph $T_j(f)$ of job j has nodes $T_j = W_j \times P_j$, dummy node $t_j^d = t_j^{m_j+1}$, and contains an arc from any node $(w_j^{i_1}, p_j^{k_1})$ to every other node $(w_j^{i_2}, p_j^{k_2})$ with $k_1 \leq k_2$, with length

$$\ell_{(i_1k_1)(i_2k_2)} = w_j^{i_1}[ES_j(f, w_j^{i_2}, p_j^{k_2}) - ES_j(f, w_j^{i_1}, p_j^{k_1})].$$

Note that we have arcs only in direction of increasing processing times, since jobs can only overstate their processing time. Furthermore, every node has an arc to the dummy type, but there are no outgoing arcs from the dummy type.

Similar as in [16], one can show that for monotone allocation rules f, some arcs in the type graph are not necessary since the corresponding incentive constraints are implied by others. We first give the definition of monotonicity in the two dimensional setting and then formulate a lemma which reduces the set of necessary incentive constraints.

Definition 7 An allocation rule f satisfies *monotonicity with respect to weights* if for every job j and fixed $p_j \in P_j$, $w_j^i < w_j^k$ implies that $ES_j(f, w_j^i, p_j) \geq ES_j(f, w_j^k, p_j)$.

Lemma 2 *Let f be an allocation rule satisfying monotonicity with respect to weights. For any job j, the following constraints imply all other incentive constraints*

$$E\pi_j(w_j^i, p_j^k) - w_j^i ES_j(f, w_j^i, p_j^k) \geq E\pi_j(w_j^{i+1}, p_j^k) - w_j^i ES_j(f, w_j^{i+1}, p_j^k) \quad \forall\, i,k,$$

$$E\pi_j(w_j^{i+1}, p_j^k) - w_j^{i+1} ES_j(f, w_j^{i+1}, p_j^k) \geq E\pi_j(w_j^i, p_j^k) - w_j^{i+1} ES_j(f, w_j^i, p_j^k) \quad \forall\, i,k,$$

$$E\pi_j(w_j^i, p_j^k) - w_j^i ES_j(f, w_j^i, p_j^k) \geq E\pi_j(w_j^i, p_j^{k+1}) - w_j^i ES_j(f, w_j^i, p_j^{k+1}) \quad \forall\, i,k.$$

The proof is rather straightforward and therefore given in the appendix. Lemma 2 can be seen as a generalization of decomposition monotonicity as discussed in the proof of Theorem 1. We can now define the reduced type graph of job j, which contains only arcs that are necessary in the sense of Lemma 2. These are arcs

- from type (w_j^i, p_j^k) to (w_j^{i+1}, p_j^k) for all $i \in \{1, \dots, m_j\}$ and $k \in \{1, \dots, q_j\}$,
- from type (w_j^i, p_j^k) to (w_j^{i-1}, p_j^k) for all $i \in \{2, \dots, m_j\}$ and $k \in \{1, \dots, q_j\}$,
- from type (w_j^i, p_j^k) to (w_j^i, p_j^{k+1}) for all $i \in \{1, \dots, m_j\}$ and $k \in \{1, \dots, q_j - 1\}$.

A sketch of the reduced type graph is given in Figure 2. Minimal expected payments again correspond to negative of shortest paths in the reduced type graph. We finally give

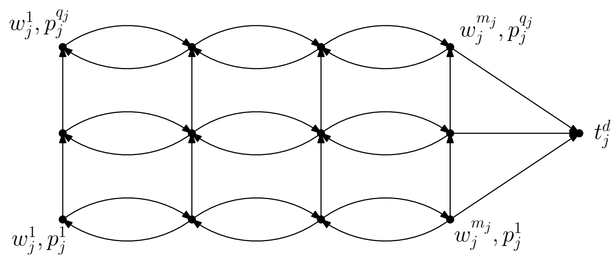

Fig. 2 Reduced type graph $T_j(f)$.

the characterization of Bayes-Nash implementable allocation rules for the two dimensional setting, which is a consequence of our restriction of the strategy space for each job, i.e., the assumption that no job can understate its required processing time.

Theorem 6 *Allocation rule f is Bayes-Nash implementable in the given two dimensional setting if and only if it satisfies monotonicity with respect to weights.*

Proof Recalling that implementability of f is equivalent with the fact that the type graph as no negative length directed cycle, monotonicity with respect to weights follows via (3), just as in the single dimensional setting. Now consider any allocation rule f that is monotone with respect to weights, and consider the corresponding type graph $T_j(f)$. We need to show that it has no negative length directed cycle. By Lemma 2, it suffices to consider the reduced type graph. Again via (3), monotonicity with respect to weights implies that there are no negative length directed cycles consisting of only two arcs. But observe that every cycle in the reduced type graph $T_j(f)$ consists of a finite number of two-cycles. The claim follows. □

4.2 On Optimal Mechanisms

Given the elegant approach by Malakhov and Vohra [16] for an auction setting with two dimensional type spaces, it is tempting to try and use the network approach also in the two dimensional setting for the scheduling problem. In this section, we show that this won't work. A first and crucial problem is that, in contrast to the single dimensional case, the shortest paths in the type graph may now depend on the allocation rule f. Hence, we cannot express minimum payments in a closed formula. Exactly this problem is ruled out in [16], but it turns out that the situation here is worse.

Definition 8 We say that an allocation rule f satisfies independence of irrelevant alternatives (IIA) if the relative order of any two jobs j_1 and j_2 is the same in the schedules $f(s)$ and $f(t)$ for any two type profiles $s,t \in T$ that differ only in the types of jobs from $J \setminus \{j_1, j_2\}$.

Note that the IIA axiom has no bite if the number of jobs is two. Another way to state the IIA axiom is that the relative order of two jobs is independent of all other jobs.

Theorem 7 *The allocation rule of the optimal mechanism for the two dimensional setting does not generally satisfy IIA.*

Proof The following (minimal) instance has been found by using an ILP formulation for the optimal mechanism design problem, in which the IIA condition can be cast as a linear constraint. There are 3 jobs. Both job 1 and job 2 have a type space containing only one type, type $(w_1, p_1) = (2,1)$ and $(w_2, p_2) = (9,8)$ respectively. Job 3 has type space $(w_3, p_3) \in \{1,3,5\} \times \{5,7\}$ and the corresponding probabilities for its types are listed below.

$$\phi_3(1,7) = 0.24 \qquad \phi_3(3,7) = 0.24 \qquad \phi_3(5,7) = 0.10$$
$$\phi_3(1,5) = 0.24 \qquad \phi_3(3,5) = 0.02 \qquad \phi_3(5,5) = 0.16$$

A graphical representation of the instance is sketched in Figure 3. We show that for this instance the unique allocation rule that is Bayes-Nash implementable, individually rational and minimizes the expected total payments, does not satisfy independence of irrelevant alternatives. The instance has three jobs and therefore we have $3! = 6$ different schedules. We denote by schedule 213 the schedule where job 2 is scheduled first and job 3 is scheduled last. Job 1 and job 2 both have only one possible type, whereas job 3 can have 6 types. Therefore the type profile is only dependent on the type of job 3. We therefore have six type profiles. For this

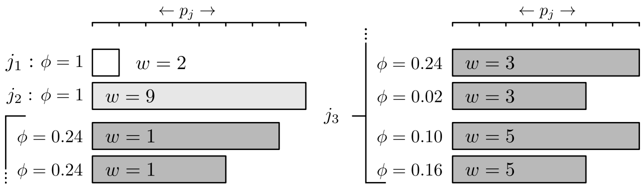

Fig. 3 Instance with three jobs for which the optimal allocation rule is not IIA.

instance IIA implies that for all six type profiles, the allocation rule must choose a schedule in which the relative order of job 1 and job 2 is the same. Therefore the allocation rule must choose schedules from either $\{123, 132, 312\}$ or $\{213, 231, 321\}$ for all six type profiles.

As an example consider allocation rule f, assigning the following schedules to reported types of job 3.

$$(1,7) \to 123 \qquad (3,7) \to 123 \qquad (5,7) \to 312$$
$$(1,5) \to 123 \qquad (3,5) \to 132 \qquad (5,5) \to 132$$

For job 1 and 2 we do not need to take into account incentive constraints as they only have one type. In order to evaluate the individual rationality constraints we need to compute the expected start times for job 1 and 2. We get

$$\begin{aligned}\pi_1^f(2,1) &= w_1 \cdot ES_1(f,2,1)\\ &= 2 \cdot (0.24 \cdot 0 + 0.24 \cdot 0 + 0.02 \cdot 0\\ &\qquad + 0.24 \cdot 0 + 0.16 \cdot 0 + 0.10 \cdot 7)\\ &= 1.40\end{aligned}$$

whereas for job 2 we have

$$\begin{aligned}\pi_2^f(9,8) &= w_2 \cdot ES_2(f,9,8)\\ &= 9 \cdot (0.24 \cdot 1 + 0.24 \cdot 1 + 0.02 \cdot 6\\ &\qquad + 0.24 \cdot 1 + 0.16 \cdot 6 + 0.10 \cdot 8)\\ &= 23.40\end{aligned}$$

For job 3 we have to take into account both the incentive and the individual rationality constrains. The type graph for job 3 is depicted in Figure 4.

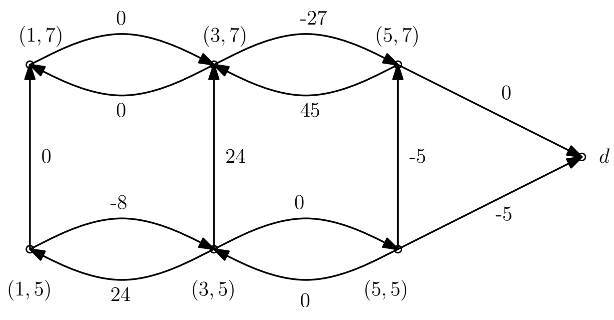

Fig. 4 Type graph corresponding to allocation rule f.

We conclude that the minimal payments to job 3 are

$$\pi_3^f(1,5) = \pi_3^f(1,7) = \pi_3^f(3,7) = 27$$
$$\pi_3^f(3,5) = \pi_3^f(5,5) = 5$$
$$\pi_3^f(5,7) = 0$$

Now that we have computed the minimal payments to all jobs, we can compute the minimal total expected payment achieved by allocation rule f.

$$\begin{aligned} P^{min}(f) &= \pi_1^f(2,1) + \pi_2^f(9,8) + \sum_{t_3 \in T_3} \phi_3(t_3)\pi_3^f(t_3) \\ &= 1.40 + 23.40 \\ &\quad + (3 \cdot 0.24)27 + (0.02 + 0.16)5 + (0.10)0 \\ &= 45.14 \end{aligned}$$

In the same way we can compute the minimal expected total payments achieved by all other $2 \cdot 3^6 - 1 = 1457$ allocation rules that are IIA. For this instance it turns out that allocation rule f is the unique Bayes-Nash implementable allocation rule that achieves minimal expected total payments, while satisfying the IIA condition.

Now consider allocation rule g, that chooses for each type profile the following schedules.

$(1,7) \to 123$	$(3,7) \to 123$	$(5,7) \to 312$
$(1,5) \to 123$	$(3,5) \to 231$	$(5,5) \to 132$

This allocation rule does not satisfy the IIA condition: the relative order of jobs 1 and 2 for case $(3,5)$ is different from their relative order in all other cases, although the types of jobs 1 and 2 are identical. Using the same approach as for allocation rule f, we calculate $\pi_1^g(2,1) = 1.92$ and $\pi_2^g(9,8) = 22.32$, respectively. The type graph for job 3 is depicted in Figure 5. The minimal payments to job 3 are

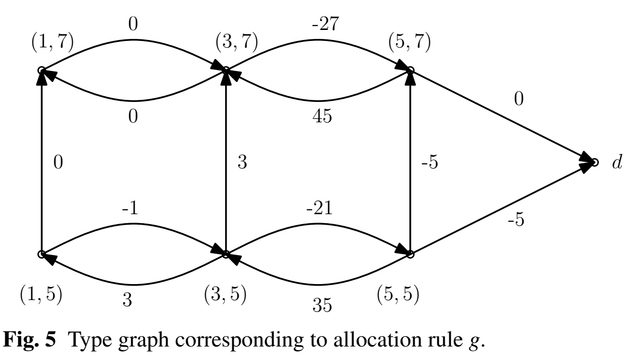

Fig. 5 Type graph corresponding to allocation rule g.

$$\pi_3^f(1,5) = \pi_3^f(1,7) = \pi_3^f(3,7) = 27$$
$$\pi_3^f(3,5) = 26$$
$$\pi_3^f(5,5) = 5$$
$$\pi_3^f(5,7) = 0$$

For allocation rule g, the minimal expected total payment is therefore

$$\begin{aligned} P^{min}(g) &= \pi_1^g(2,1) + \pi_2^g(9,8) + \sum_{t_3 \in T_3} \phi_3(t_3)\pi_3^g(t_3) \\ &= 1.92 + 22.32 \\ &\quad + (3 \cdot 0.24)27 + (0.02)26 + (0.16)5 + (0.10)0 \\ &= 45.00 \end{aligned}$$

This proves the claim. □

Theorem 7 shows that any priority based allocation rule where the priority of a job is computed from the characteristics of the job itself cannot be optimal in general. We conclude that the network approach which we used for the single dimensional case, and which is used also by [16] for example, fails in the two dimensional case we consider here[4]. When there are only two jobs present, then IIA is trivially satisfied. Recall that in the single dimensional case the optimal mechanism is efficient for symmetric jobs and regular distributions and that the uniform distribution is regular. This is contrasted by the following theorem for the two dimensional case.

Theorem 8 *Even for two symmetric jobs, 2×2-type spaces and uniform probability distributions, the optimal mechanism is not efficient.*

Proof Consider the following example with two jobs, $W_1 = W_2 = \{1,2\}$ and $P_1 = P_2 = \{1,2\}$. We assume that $\phi_1(i,k) = \phi_2(i,k) = \frac{1}{4}$ for $i,k \in \{1,2\}$. On the one hand, consider the efficient allocation rule f^e, which schedules the job with higher weight over processing time ratio first. On the other hand, regard the so-called w-rule, f^w, that schedules the job with the higher weight first. In case of ties, both rules schedule job 1 first. Note that both f^e and f^w satisfy monotonicity with respect to weights, and are therefore implementable. The expected start times for the w-rule f^w are

$$ES_1(f^w,1,1) = ES_1(f^w,1,2) = 3/4,$$
$$ES_1(f^w,2,1) = ES_1(f^w,2,2) = 0,$$
$$ES_2(f^w,1,1) = ES_2(f^w,1,2) = 3/2,$$
$$ES_2(f^w,2,1) = ES_2(f^w,2,2) = 3/4.$$

The type graphs corresponding to f^w for Jobs 1 and 2 respectively are shown in Figure 6. From this, the optimal payments can be computed as $\pi_1^{f^w}(2,1) = \pi_1^{f^w}(2,2) = 0,$

[4] Of course, for any fixed allocation rule f, we can use the network approach to compute optimal payments. But in the two dimensional setting, the approach fails for determining f itself: For an arbitrary, implementable f, the total expected payment $P^{min}(f)$ is a linear function in the values $ES_j(f,t_j)$. But in contrast to the single dimensional case, the coefficients of the values $ES_j(f,t_j)$ may depend on the allocation rule f. Hence, we see no easy way to conclude which f is the minimizer for $P^{min}(f)$.

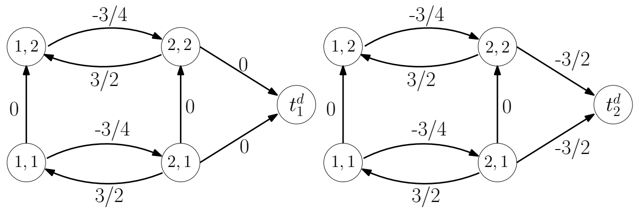

Fig. 6 Type graphs for the w-rule for Jobs 1 and 2.

$\pi_1^{f^w}(1,1) = \pi_1^{f^w}(1,2) = 3/4$, $\pi_2^{f^w}(2,1) = \pi_2^{f^w}(2,2) = 3/2$, $\pi_2^{f^w}(1,1) = \pi_2^{f^w}(1,2) = 9/4$. Hence the (minimum) total expected payment for the w-rule is:

$$P^{min}(f^w) = \frac{1}{4}\sum_j \sum_{(i,k)} \pi_j^{f^w}(i,k) = 9/4.$$

On the other hand, for the efficient allocation rule f^e, we get the expected start times

$$\begin{aligned}
&ES_1(f^e,1,1) = ES_1(f^e,2,2) = 1/4,\\
&ES_1(f^e,1,2) = 1,\\
&ES_1(f^e,2,1) = 0,\\
&ES_2(f^e,1,1) = ES_2(f^e,2,2) = 1,\\
&ES_2(f^e,1,2) = 3/2,\\
&ES_2(f^e,2,1) = 1/4.
\end{aligned}$$

The type graphs corresponding to f^e for Jobs 1 and 2 respectively are shown in Figure 7. From this, the optimal payments can be computed as $\pi_1^{f^e}(1,1) = \pi_1^{f^e}(2,2) = 1/2$, $\pi_1^{f^e}(2,1) = 0$, $\pi_1^{f^e}(1,2) = 5/4$, $\pi_2^{f^e}(1,1) = \pi_2^{f^e}(2,2) = 2$, $\pi_2^{f^e}(1,2) = 5/2$, $\pi_2^{f^e}(2,1) = 1/2$.

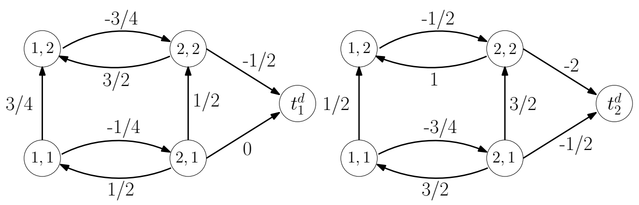

Fig. 7 Type graphs for the efficient rule for Jobs 1 and 2.

Hence the (minimum) total expected payment for the efficient rule f^e is

$$P^{min}(f^e) = \frac{1}{4}\sum_j \sum_{(i,k)} \pi_j^{f^e}(i,k) = 37/16.$$

So $P^{min}(f^e) > P^{min}(f^w)$. Thus, the efficient allocation is dominated by the w-rule, and consequently does not correspond to the optimal mechanism. □

5 Discussion

In the first part of the paper, we show that the graph theoretic approach is a simple and intuitive tool for optimal mechanism design with discrete types. It yields a closed form solution for the optimal mechanism in the single dimensional case, which can be computed in polynomial time. This result parallels Myerson's results for single item auctions, and we hope it provides insight into mechanism design methodology for the scheduling community. For the two dimensional case, in light of Theorem 7, it is conceivable that a closed form solution does not exist in general. As to computational complexity of the two dimensional problem, it has been shown recently that an optimal randomized mechanism can be computed in polynomial time [12], but it remains an open problem to determine the computational complexity for computing a deterministic optimal mechanism.

Acknowledgements Thanks to Rakesh Vohra, Jason Hartline and Ruben Hoeksma for helpful discussions, and thanks to the anonymous referees for thoughtful and constructive remarks.

References

1. M. Armstrong. Optimal multi-object auctions. *Review of Economic Studies*, 67:455–481 (2000).
2. Y. Cai, C. Daskalakis and S.M. Weinberg. Optimal Multi-Dimensional Mechanism Design: Reducing Revenue to Welfare Maximization. In: Proc. 53rd FOCS (2012).
3. E. H. Clarke. Multipart pricing of public goods. *Public Choice*, 11(1):17–33 (1971).
4. I. Curiel, G. Pederzoli, and S. Tijs. Sequencing games. *European Journal of Operational Research*, 40:344–351 (1989).
5. A. Gershkov, J. K. Goeree, A. Kushnir, B. Moldovanu and X. Shi. On the Equivalence of Bayesian and Dominant Strategy Implementation. *Econometrica* 81(1):197–220 (2013).
6. T. Groves. Incentives in teams. *Econometrica*, 41:617–631 (1973).
7. R. Hain and M. Mitra: Simple sequencing problems with interdependent costs. *Games and Economic Behavior* 48(2): 271–291 (2004).
8. H. Hamers, F. Klijn, and J. Suijs. On the balancedness of multiple machine sequencing games. *European Journal of Operational Research*, 119:678–691 (1999).
9. J. Hartline and A. Karlin. Profit maximization in mechanism design. In N. Nisan, T. Roughgarden, É. Tardos, and V. Vazirani, editors, *Algorithmic Game Theory*. Cambridge University Press (2007).
10. B. Heydenreich, D. Mishra, R. Müller, and M. Uetz. Optimal Mechanisms for Single Machine Scheduling. In: Internet and Network Economics, C. Papadimitriou and S. Zhang (eds.), Lecture Notes in Computer Science 5385:414–425 (2008).
11. B. Heydenreich, R. Müller, M. Uetz, and R. Vohra. Characterization of revenue equivalence. *Econometrica*, 77(1):307–316 (2009).
12. R. Hoeksma and M. Uetz Two-Dimensional Optimal Mechanisim Design for a Sequencing Problem, In: Integer Programming and Combinatorial Optimization, M. Goemans and J. Correa (eds.), Lecture Notes in Computer Science 7801:242–253 (2013).

13. T. Kittsteiner and B. Moldovanu. Priority auctions and queue disciplines that depend on processing time. *Management Science*, 51(2):236–248 (2005).
14. V. Krishna. *Auction Theory*. Academic Press (2002).
15. R. Lavi and C. Swamy: Truthful mechanism design for multidimensional scheduling via cycle monotonicity. *Games and Economic Behavior* 67(1):99–124 (2009).
16. A. Malakhov and R. Vohra. An optimal auction for capacity constrained bidders: a network perspective. *Economic Theory*, 39(1):113–128 (2009).
17. A. M. Manelli and D. Vincent. Bayesian and dominant strategy implementation in the independent private values model. *Econometrica*, 78(6):1905–1939 (2010).
18. M. Mitra. Mechanism design in queueing problems. *Economic Theory*, 17(2):277–305 (2001).
19. D. Mookherjee and S. Reichelstein. Dominant Strategy Implementation of Bayesian Incentive Compatible Allocation Rules. *Journal of Economic Theory*, 56:378–399 (1992).
20. H. Moulin. On scheduling fees to prevent merging, splitting, and transferring of jobs. *Mathematics of Operations Research*, 32:266–283 (2007).
21. R. Müller, A. Perea, and S. Wolf. Weak monotonicity and Bayes-Nash incentive compatibility. *Games and Economic Behavior*, 61(2):344–358 (2007).
22. R. Myerson. Optimal auction design. *Mathematics of Operations Research*, 6(1):58–73 (1981).
23. M. Pai and R. Vohra. Optimal auctions with financially constrained bidders. Discussion Paper 1471, The Center for Mathematical Studies in Economics & Management Sciences, Northwestern University, Evanston, IL (2008).
24. P. Penna and C. Ventre. Collusion-Resistant Mechanisms with Verification Yielding Optimal Solutions. *ACM Transactions on Computation Theory*, 4(2): article no. 6 (2012)
25. J.-C. Rochet. A necessary and sufficient condition for rationalizability in a quasi-linear context. *Journal of Mathematical Economics*, 16(2):191–200 (1987).
26. W. Smith. Various optimizers for single stage production. *Naval Research Logistics Quarterly*, 3:59–66 (1956).
27. J. Suijs. On incentive compatibility and budget balancedness in public decision making. *Economic Design*, 2:193–209 (1996).
28. W. Vickrey. Counterspeculation, auctions and competitive sealed tenders. *Journal of Finance*, 16:8–37 (1961).
29. R. Vohra. Optimization and mechanism design. *Mathematical Programming*, 134:283–303 (2012).

Appendix

Proof for Theorem 1 via decomposition monotonicity. We consider any monotone allocation rule f, together with type graph $T_j(f)$. We show that if there is no negative length directed cycle with only two arcs, then there is no negative length directed cycle. We first show that the arc lengths satisfy a property called *decomposition monotonicity*, i.e., whenever $i < k < l$ then $\ell_{ik} + \ell_{kl} \leq \ell_{il}$ and $\ell_{lk} + \ell_{ki} \leq \ell_{li}$. Decomposition monotonicity follows from

$$\begin{aligned}\ell_{ik} + \ell_{kl} &= w_j^i[ES_j(f,w_j^k) - ES_j(f,w_j^i)]\\ &\quad + w_j^k[ES_j(f,w_j^l) - ES_j(f,w_j^k)]\\ &\leq w_j^i[ES_j(f,w_j^k) - ES_j(f,w_j^i)]\\ &\quad + w_j^i[ES_j(f,w_j^l) - ES_j(f,w_j^k)]\\ &= w_j^i[ES_j(f,w_j^l) - ES_j(f,w_j^i)] = \ell_{il},\end{aligned}$$

where the inequality follows from monotonicity of f (recalling that $w_j^i < w_j^k < w_j^l$). Note that everything remains true if the dummy type is involved, i.e., if $l = m_j + 1$. The inequality $\ell_{lk} + \ell_{ki} \leq \ell_{li}$ follows similarly. Because of decomposition monotonicity, we conclude for any $k > i+1$ that

$$\ell_{i,i+1} + \cdots + \ell_{k-1,k} \leq \ell_{ik}$$

Similarly, $\ell_{k,k-1} + \cdots + \ell_{i+1,i} \leq \ell_{ki}$ (where k is not the dummy type, which by definition does not have outgoing arcs). Hence, for any directed cycle C, we can replace arcs (i,k) with $k > i+1$ by the chain $(i,i+1),\ldots,(k-1,k)$, thereby reducing the total length of C. The same is true for all arcs (k,i) with $k > i+1$, which can be replaced by the chain $(k,k-1),\ldots,(i+1,i)$. By 'decomposing' the cycle in this way, we see that its length can be lower bounded by the lengths of a finite number of two-cycles. Since two-cycles have non-negative length by monotonicity of f and (3), the claim is proved. □

Proof of Lemma 1. As f is Bayes-Nash implementable, $T_j(f)$ satisfies the non-negative cycle property. Consequently, we can compute shortest paths in $T_j(f)$. With $dist(w_j^i, w_j^d)$ we denote the length of a shortest path from w_j^i to w_j^d. By (3), $-dist(w_j^i, w_j^d) \leq E\pi_j(w_j^i)$. Therefore, $-dist(w_j^i, w_j^d)$ is a lower bound on the expected payment for reporting w_j^i. On the other hand, since we have $dist(w_j^i, w_j^d) \leq \ell_{ik} + dist(w_j^k, w_j^d)$ for any two types w_j^i and w_j^k, it follows that

$$-dist(w_j^k, w_j^d) \leq -dist(w_j^i, w_j^d) + \ell_{ik}.$$

Consequently, defining the payment as negative of the shortest path lengths,

$$\pi_j(w_j^i) := -dist(w_j^i, w_j^d)\,,$$

yields an incentive compatible payment scheme that minimizes the expected payment to every job for any reported type of the agent. (Recall that individual rationality is satisfied along with the incentive constraints, due to the introduction of dummy types.)

Next, recall that arc lengths in $T_j(f)$ satisfy decomposition monotonicity. Therefore, a shortest path from w_j^i to w_j^d is exactly the path that includes all $w_j^{i+1}, \ldots, w^{m_j}$. Observing that $-dist(w_j^d, w_j^d) = 0$ and

$$-dist(w_j^i, w_j^d) = \sum_{k=i}^{m_j} w_j^k[ES_j(f,w_j^k) - ES_j(f,w_j^{k+1})]$$

for all $w_j^i \in W_j \setminus \{w_j^d\}$, we have verified the first claim of the Lemma.

We are left to verify the formulae for the minimum expected total payment for a given allocation rule f. We get for $P^{min}(f)$

$$
\begin{aligned}
&= \sum_{j\in J}\sum_{i=1}^{m_j} \phi_j(w_j^i)\pi_j(w_j^i) \\
&= \sum_{j\in J}\sum_{i=1}^{m_j} \phi_j(w_j^i)\sum_{k=i}^{m_j} w_j^k[ES_j(f,w_j^k)-ES_j(f,w_j^{k+1})] \\
&= \sum_{j\in J}\sum_{i=1}^{m_j} \phi_j(w_j^i)\left[\sum_{k=i}^{m_j} w_j^k ES_j(f,w_j^k) - \sum_{k=i+1}^{m_j} w_j^{k-1} ES_j(f,w_j^k)\right] \\
&= \sum_{j\in J}\sum_{i=1}^{m_j} \phi_j(w_j^i)\left[w_j^i ES_j(f,w_j^i)\right. \\
&\qquad \left. + \sum_{k=i+1}^{m_j} ES_j(f,w_j^k)(w_j^k - w_j^{k-1})\right] \\
&= \sum_{j\in J} ES_j(f,w_j^1)w_j^1\phi_j(w_j^1) \\
&\quad + \sum_{j\in J}\sum_{i=2}^{m_j} ES_j(f,w_j^i)\left[\phi_j(w_j^i)w_j^i + (w_j^i - w_j^{i-1})\sum_{k=1}^{i-1}\phi_j(w_j^k)\right].
\end{aligned}
$$

Therefore, defining the virtual weights $\overline{w}_j$ by setting $\overline{w}_j^1 = w_j^1$, and for $i = 2,\dots,m_j$,

$$\overline{w}_j^i = w_j^i + (w_j^i - w_j^{i-1})\frac{\sum_{k=1}^{i-1}\phi_j(w_j^k)}{\phi_j(w_j^i)}$$

yields the closed form

$$P^{min}(f) = \sum_{j\in J}\sum_{w_j\in W_j}\phi_j(w_j)\overline{w}_j ES_j(f,w_j). \qquad \square$$

Proof of Theorem 5. For any implementable f, a candidate for the optimal DSIC payment can be found in exactly the same way as in the Bayes-Nash case, namely as (negative of the length of) shortest paths in the type graphs $T_j(f)$, only that we now have $|W_{-j}|$ many type graphs for each job j, one for each possible report w_{-j} of the other jobs. The arc lengths in these type graphs are

$$\ell_{ik} = w_j^i\left[S_j(f,(w_j^k,w_{-j})) - S_j(f,(w_j^i,w_{-j}))\right].$$

The resulting payments, for given w_{-j}, are $\pi_j(w_j^d,w_{-j}) = 0$, and

$$\pi_j(w_j^i,w_{-j}) = \sum_{k=i}^{m_j} w_j^k\left[S_j(f,(w_j^k,w_{-j})) - S_j(f,(w_j^{k+1},w_{-j}))\right]$$

for $i = 1,\dots,m_j$. It is an easy exercise to verify incentive compatibility and individual rationality of these payments. If we now compute the total expected payment $P(f)$ of the resulting mechanism (f,π), we get

$$
\begin{aligned}
P(f) &= \sum_j\sum_{w_{-j}}\phi(w_{-j})\sum_i\phi_j(w_j^i)\sum_{k=i}^{m_j} w_j^k\left[S_j(f,(w_j^k,w_{-j}))\right. \\
&\qquad \left. - S_j(f,(w_j^{k+1},w_{-j}))\right] \\
&= \sum_j\sum_i\phi_j(w_j^i)\sum_{k=i}^{m_j} w_j^k\left[\sum_{w_{-j}}\phi(w_{-j})S_j(f,(w_j^k,w_{-j}))\right. \\
&\qquad \left. - \sum_{w_{-j}}\phi(w_{-j})S_j(f,(w_j^{k+1},w_{-j}))\right] \\
&= \sum_j\sum_i\phi_j(w_j^i)\sum_{k=i}^{m_j} w_j^k\left[ES_j(f,w_j^k) - ES_j(f,w_j^{k+1})\right] \\
&= P^{min}(f).
\end{aligned}
$$

Note that for each job the minimal payments for dominant strategy incentive compatibility are, in expectation, identical to the minimal payments that we computed before for Bayes-Nash implementability. Finally, note that the allocation rule that has been defined in Theorem 2 is indeed monotone in w_j for any report w_{-j}. $\square$

Proof of Lemma 2. For any $i_1,i_2,i_3 \in \{1,\dots,m_j+1\}, i_1 < i_2 < i_3$, and any $k \in \{1,\dots,q_j\}$ the constraint

$$
\begin{aligned}
&E\pi_j(w_j^{i_1},p_j^k) - w_j^{i_1}ES_j(f,w_j^{i_1},p_j^k) \geq \\
&\qquad E\pi_j(w_j^{i_3},p_j^k) - w_j^{i_1}ES_j(f,w_j^{i_3},p_j^k)
\end{aligned}
$$

is implied by

$$
\begin{aligned}
&E\pi_j(w_j^{i_1},p_j^k) - w_j^{i_1}ES_j(f,w_j^{i_1},p_j^k) \geq \\
&\qquad E\pi_j(w_j^{i_2},p_j^k) - w_j^{i_1}ES_j(f,w_j^{i_2},p_j^k), \text{ and} \\
&E\pi_j(w_j^{i_2},p_j^k) - w_j^{i_2}ES_j(f,w_j^{i_2},p_j^k) \geq \\
&\qquad E\pi_j(w_j^{i_3},p_j^k) - w_j^{i_2}ES_j(f,w_j^{i_3},p_j^k).
\end{aligned}
$$

The claim follows by adding up the latter two constraints, and using monotonicity of f together with the fact that $w_j^{i_1} < w_j^{i_2}$. Everything remains true if the dummy type is involved, i.e., if $(w_j^{i_3},p_j^k) = (w_j^{m_j+1},p_j^k) = t_j^d$. Therefore, all constraints of the type

$$
\begin{aligned}
&E\pi_j(w_j^{i_1},p_j^k) - w_j^{i_1}ES_j(f,w_j^{i_1},p_j^k) \geq \\
&\qquad E\pi_j(w_j^{i_2},p_j^k) - w_j^{i_1}ES_j(f,w_j^{i_2},p_j^k) \qquad (6)
\end{aligned}
$$

are implied by the subset of constraints where $i_2 = i_1 + 1$.

A similar effect can be shown for the "reverse" incentive constraints, i.e., the above constraints for $i_3 < i_2 < i_1$, where $i_1,i_2,i_3 \in \{1,\dots,m_j\}$. Again, out of all constraints of the type

$$E\pi_j(w_j^{i_1},p_j^k) - w_j^{i_1}ES_j(f,w_j^{i_1},p_j^k) \geq$$

$$E\pi_j(w_j^{i_2}, p_j^k) - w_j^{i_1} ES_j(f, w_j^{i_2}, p_j^k), \tag{7}$$

only those with $i_2 = i_1 - 1$ are necessary.

Similarly, out of all constraints of the type

$$E\pi_j(w_j^i, p_j^{k_1}) - w_j^i ES_j(f, w_j^i, p_j^{k_1}) \geq E\pi_j(w_j^i, p_j^{k_2}) - w_j^i ES_j(f, w_j^i, p_j^{k_2}) \tag{8}$$

for $i \in \{1, \dots, m_j\}$, $k_1, k_2 \in \{1, \dots, q_j\}$, $k_1 < k_2$ only those with $k_2 = k_1 + 1$ are necessary.

Finally, for any types $(w_j^{i_1}, p_j^{k^1}), (w_j^{i_2}, p_j^{k^2})$ with $i_1 < i_2$ and $k_1 < k_2$ the corresponding "diagonal" constraint

$$E\pi_j(w_j^{i_1}, p_j^{k_1}) - w_j^{i_1} ES_j(f, w_j^{i_1}, p_j^{k_1}) \geq E\pi_j(w_j^{i_2}, p_j^{k_2}) - w_j^{i_1} ES_j(f, w_j^{i_2}, p_j^{k_2})$$

follows by adding up the corresponding constraints of type (8) and (6),

$$E\pi_j(w_j^{i_1}, p_j^{k_1}) - w_j^{i_1} ES_j(f, w_j^{i_1}, p_j^{k_1}) \geq E\pi_j(w_j^{i_1}, p_j^{k_2}) - w_j^{i_1} ES_j(f, w_j^{i_1}, p_j^{k_2}) \quad \text{and}$$

$$E\pi_j(w_j^{i_1}, p_j^{k_2}) - w_j^{i_1} ES_j(f, w_j^{i_1}, p_j^{k_2}) \geq E\pi_j(w_j^{i_2}, p_j^{k_2}) - w_j^{i_1} ES_j(f, w_j^{i_2}, p_j^{k_2}).$$

Similarly, for any $(w_j^{i_1}, p_j^{k^1}), (w_j^{i_2}, p_j^{k^2})$ with $i_2 < i_1$ and $k_1 < k_2$, the corresponding "diagonal" constraint follows by adding up the appropriate constraints of type (8) and (7). □