Journal of Artificial Intelligence Research 76 (2023) 959-1018 Submitted 08/2022; published 04/2023

Optimal and Efficient Auctions for the Gradual Procurement of Strategic Service Provider Agents

Farzaneh Farhadi F.FARHADI@ASTON.AC.UK
Maria Chli M.CHLI@ASTON.AC.UK
Department of Computer Science, Aston University, UK

Nicholas R. Jennings N.R.JENNINGS@LBORO.AC.UK
Loughborough University, UK

Abstract

We consider an outsourcing problem where a software agent procures multiple services from providers with uncertain reliabilities to complete a computational task before a strict deadline. The service consumer's goal is to design an outsourcing strategy (defining which services to procure and when) so as to maximize a specific objective function. This objective function can be different based on the consumer's nature; a socially-focused consumer often aims to maximize social welfare, while a self-interested consumer often aims to maximize its own utility. However, in both cases, the objective function depends on the providers' execution costs, which are privately held by the self-interested providers and hence may be misreported to influence the consumer's decisions. For such settings, we develop a unified approach to design truthful procurement auctions that can be used by both socially-focused and, separately, self-interested consumers. This approach benefits from our proposed weighted threshold payment scheme which pays the provably minimum amount to make an auction with a monotone outsourcing strategy incentive compatible. This payment scheme can handle contingent outsourcing plans, where additional procurement happens gradually over time and only if the success probability of the already hired providers drops below a time-dependent threshold. Using a weighted threshold payment scheme, we design two procurement auctions that maximize, as well as two low-complexity heuristic-based auctions that approximately maximize, the consumer's expected utility and expected social welfare, respectively. We demonstrate the effectiveness and strength of our proposed auctions through both game-theoretical and empirical analysis.

1. Introduction

There are many reasons why a business may choose to outsource a particular task, job, or process. Some of the recognized benefits of outsourcing include an improved focus on core business activities, increased efficiency due to the task being performed by specialists, and reduced costs. Outsourcing has gained wide-spread popularity in a wide range of application areas, including finance (Reddy & Ramachandran, 2008), IT (Lacity et al., 2009), cloud computing (Motahari Nezhad et al., 2009), supply chain management (Winsper & Chli, 2013; Chli & Winsper, 2015), and marketing (Kotabe et al., 2011). The focus of this paper is on outsourcing applications where both the outsourcer and the service providers are automated software agents and the task can be procured and delivered by an automated computer system. Examples of such settings can be found in (Stanford-Smith, 2000; Grefen, 2006; Tai et al., 2010; Gahletia, 2021).

Although there are many benefits, outsourcing presents a number of challenges that must be addressed. First, when a task is outsourced, the outsourcing agent, the consumer, has no direct control over the process. Therefore, it may face some uncertainties in the task's delivery time and hence be in danger of task failure if a strict deadline must be met[1]. In such settings, the consumer can outsource the task to multiple providers to allow for extra redundancy and failure protection (Barbour & Wojcik, 1989). However, redundancy can be very costly if not designed optimally, as it increases both the execution cost (i.e., the cost imposed on the providers to perform the task) and the invocation cost (i.e., the cost paid by the consumer for recruitment).

The consumer will often wish to balance the tradeoff between the success probability and either of these costs, based on its ultimate goal (i.e., prosocial vs. self-interested). A prosocial (i.e., socially-focused) consumer often aims to maximize the expected social welfare, which is equivalent to balancing the tradeoff between success and execution cost, while a self-interested consumer often aims to maximize its own expected utility, which is equivalent to balancing the tradeoff between success and invocation cost. However, some of the information needed to find the tipping points of these tradeoffs, such as the providers' execution costs, is often unavailable to the consumer and must be extracted from the providers. In real-world applications, service providers are often self-interested and may misrepresent their private information upon request, if this promises to increase their profits. For instance, a self-interested provider, which aims only to maximize its own profit, may have an incentive to inflate its cost to earn higher revenue. Eliciting truthful information from self-interested providers is the second challenge faced by an outsourcing agent.

The main goal of this paper is to design an outsourcing mechanism that optimally addresses the two challenges mentioned above. To address the challenge of uncertain delivery times, we introduce the method of *contingent outsourcing planning* that enables the consumer to build more efficient redundancy-based outsourcing strategies by making its decisions contingent on past observations. In this method, unlike the conventional simultaneous outsourcing techniques, where all the selected providers are simultaneously asked to do the task (Zhang, 2018; Feldman et al., 2020), the outsourcing process is spread over time to make room for observing the past decisions' outcomes and making more informed future decisions. In a contingent outsourcing plan, the consumer continuously updates its belief about the providers' service duration times based on its observations of the invoked providers' performance. The consumer may then decide to invoke some new providers if the belief that the invoked providers can complete the task on time falls below a certain threshold.

The parameters of the optimal contingent outsourcing plan, such as the time and order of invocations, depend on the providers' execution costs, which are unknown to the consumer and must be elicited from self-interested providers. One way to address this challenge is to design incentive-based outsourcing mechanisms, often called procurement auctions, that ask providers to submit "bids" and then determine the outsourcing plan and the payments based on these bids. For an incentive mechanism to be effective, the allocation function and payment scheme, which map the bids to outsourcing plans and payments, respectively,

1. We consider the set of domains where a strict deadline is imposed for each task. A treatment for settings when this is not the case (e.g., when the deadlines are slightly flexible or can be postponed by paying penalties) is left for future work.

must be designed such that the providers find it beneficial to (1) participate in the outsourcing process (*individual rationality (IR)*), and (2) bid their execution costs truthfully (*incentive compatibility (IC)*). The available incentive mechanisms can be categorized based on whether they establish these properties weakly (i.e., in expectation when other bidders are truthful), namely interim IR and Bayesian IC (BIC), or strongly (i.e., at every single outcome and irrespective of what others do), namely ex-post IR and dominant strategy IC (DSIC). However, most existing mechanisms in any of these categories apply only for simultaneous outsourcing techniques and hence cannot be directly used for designing a contingent planning-based incentive mechanism.

To bridge this gap, we first construct a mapping which transforms each contingent outsourcing plan into a randomized simultaneous outsourcing strategy. We then use Myerson's fundamental results for randomized simultaneous mechanisms (Myerson, 1981) to design a *weighted* threshold payment scheme that guarantees the weaker versions of IC and IR properties (i.e., BIC and interim IR) for a contingent planning-based incentive mechanism, at the minimum cost. We show later that the auctions we design by employing this payment scheme satisfy the stronger versions of both properties. The weighted threshold payment scheme, which determines the price of recruiting each provider based on a weighted integral over the provider's possible bid values, is also highly compatible with approximation algorithms, meaning that it ensures the nice features mentioned above even if the allocation function is suboptimal. We take advantage of this property and design not only optimal but also suboptimal low-complexity auctions, which are of practical importance.

In more detail, first, using the weighted threshold payment scheme, we design two novel incentive mechanisms, in the form of procurement auctions, namely OCPA and ECPA, that can be used to maximize the consumer's expected utility and the expected social welfare, respectively. These auctions do not only establish BIC and interim IR, but also exhibit the stronger notions of DSIC and ex-post IR. As an additional interesting feature, the OCPA auction ensures that the consumer will never run into a deficit. To test the effectiveness of the proposed auctions, several benchmarks are adopted to demonstrate the performance in terms of the consumer's utility and social welfare. The results show that contingent planning can improve these two metrics by up to 120% and 123%, compared to the current state of the art, respectively.

Note that both OCPA and ECPA auctions, each seeking a different objective, use very similar algorithms for selecting their contingent outsourcing plans. This algorithm is a branch-and-bound-based method for a mixture of continuous and combinatorial optimization. Determining the best contingent outsourcing plan for a specific bid vector falls within this domain as decisions need to be made on both the sequence of services to be invoked (discrete variables) and their invocation times (continuous variables). Our proposed branch-and-bound algorithm solves the contingent planning problems optimally, but at the expense of high time complexity. To overcome this issue, we have also developed a low-complexity heuristic algorithm that can produce close-to-optimal solutions at reduced complexity.

Against this background, we advance the state-of-the-art in the following ways.

- We propose the first payment scheme that can be used to design both optimal and suboptimal contingent planning-based incentive compatible mechanisms.

- We are the first to develop a dominant strategy incentive compatible and ex-post individually rational procurement auction that achieves the optimal tradeoff between success probability and invocation cost. It achieves this by implementing contingent action plans.
- We propose a novel social welfare maximizing contingent planning-based outsourcing mechanism that is both dominant strategy incentive compatible and ex-post individually rational.
- We also present low-complexity versions of the above-mentioned auctions, called SOCPA and SECPA, that reduce the runtime of the original auctions by 99%, while preserving the DSIC and ex-post IR properties and not compromising the performance (optimality gaps of less than 1%).

The rest of the paper is organized as follows. After a review of the main relevant literature (Section 2), a specification of our outsourcing problem is given in Section 3. In Section 4, we formally define contingent outsourcing planning as the main outsourcing method used in this paper. In Section 5, we formulate the outsourcing problem with self-interested agents as an auction design problem and derive conditions that guarantee our desirable properties. Section 6 is devoted to solving the optimal (i.e., utility maximizing) auction design problem introduced in Section 5. The theoretical properties of the designed auction as well as a low-complexity version of it are presented in Section 7. Section 8 discusses how an approach similar to that used in Sections 5 and 6 can be used to design a social welfare maximizing procurement auction. In Section 9, we evaluate our proposed auctions by simulations compared to several benchmarks. We conclude our paper in Section 10.

2. Related Work

Reaching a desired outcome when agents have private information is often achieved through some form of negotiation process (Jennings et al., 2001; Fatima et al., 2004; Bartolini et al., 2004; Zheng et al., 2016). In (Yao et al., 2010; Wang et al., 2021), the authors propose protocols that can be used for negotiation between the consumer and the service providers in an outsourcing application. These protocols provide a desired outcome when agents are truthful in information sharing. However, their performance significantly degrades when providers are selfish and may misreport their private information if it benefits them.

Determining appropriate rules to achieve a desired outcome when providers are selfish is the subject of incentive mechanism design (Myerson, 1988; Börgers, 2015; Farhadi & Jennings, 2021; Farhadi et al., 2019). Incentive mechanisms can be very diverse. However, according to the direct revelation principle (Myerson, 1979; Maskin et al., 1979; Holmström, 1977), the consumer can restrict attention to direct mechanisms where the providers are asked to reveal their private information in terms of bids, and the outsourcing strategy and payments are determined based on these bids. Such mechanisms are often known as auctions and these are the basis for our work.

Auctions are categorized into two groups based on the auction's primary goal: *efficient auctions* or *optimal auctions*. The former are designed to maximize the social welfare, while the latter aim to maximize the auctioneer's expected utility (also called revenue). Most of the early works in the field have dealt with efficient auction design (Vickrey, 1961; Clarke, 1971; Groves, 1973). The first insights into the design of optimal auctions have started with the works of Myerson (Myerson, 1981) and Riley and Samuelson (Riley & Samuelson, 1981) in 1981. Since then, much effort has been devoted to both issues in auction theory. In the next two subsections, we review the related works in each of these fields. We then briefly review the other researches related to our study in Section 2.3.

2.1 Social Welfare Maximizing (Efficient) Auctions

The efficiency problem is theoretically solved by the Vickrey-Clarke-Groves (VCG) mechanism (Clarke, 1971; Groves, 1973) and its variants, including execution–contingent VCG (Ramchurn et al., 2009; Gerding et al., 2010), dynamic VCG (Bergemann & Valimaki, 2006), multi-stage VCG (Zhang & Verwer, 2012), ad/position dependent cascade VCG (APDC-VCG) (Farina & Gatti, 2017), and online VCG (Parkes & Singh, 2003). All of these VCG-based mechanisms guarantee DSIC, mainly due to their specific payment scheme called VCG payments. However, VCG payments guarantee DSIC only if the allocation function is the exact solution (and not an approximated solution) of an optimization problem, which is generally NP-hard. This incompatibility of the VCG payments with approximation algorithms is recognized as one of their main drawbacks (Kraft et al., 2014).

In order to reduce complexity, some researchers focus on developing heuristic non-VCG-based mechanisms that approximate the optimal social welfare. Some of these mechanisms satisfy DSIC (Babaioff et al., 2009; Huang & Kannan, 2012; Gerding et al., 2010; Stein et al., 2011), however, some others turn to BIC, which is a weaker notion of incentive compatibility (Hartline et al., 2011; Mansour et al., 2020; Dughmi et al., 2021). Among these mechanisms, some satisfy ex-post IR (Babaioff et al., 2009; Gerding et al., 2010; Stein et al., 2011), some satisfy interim IR (Huang & Kannan, 2012), and some do not satisfy any notion of individual rationality (Mansour et al., 2020).

In the context of service procurement with uncertainty, the most efficient ex-post IR mechanism available so far is a pairing mechanism proposed in (Stein et al., 2011). This mechanism first pairs providers randomly. Then, for each pair, it puts the provider with the lowest bid into a candidate set $\mathcal{K}$ and assigns it a virtual cost equal to its pair's bid. When this procedure ends, the mechanism restricts itself to providers in $\mathcal{K}$ and computes the social welfare maximizing recruitment strategy by assuming that the providers' costs are equal to their virtual costs. Each provider will receive a payment equal to its virtual cost upon recruitment. The pairing mechanism is proved to satisfy DSIC and ex-post IR and is currently the state-of-the-art in the class of approximate social welfare maximizing service procurement auctions. However, we will show in the numerical results section (Section 9) that the inefficiency of the pairing mechanism could be as high as 58%. We will also demonstrate that our proposed heuristic-based low-complexity SECPA auction can outperform the pairing mechanism by up to 137%.

2.2 Utility Maximizing (Optimal) Auctions

The study of optimal auctions started with the seminal work of Myerson in 1981 (Myerson, 1981). In this work, Myerson introduced the first BIC optimal auction for single-object problems, which has been proven later to also satisfy DSIC (Manelli & Vincent, 2010; Lee et al., 2021). Since then, there has been some progress in extending Myerson's fundamental result to multi-object environments, however, a general and analytical optimal auction in this framework has yet to be found (Nedelec et al., 2021).

The available literature on multi-object auction design can be categorized based on whether the objects are identical or not (homogeneous (Maskin & Riley, 2000; Malakhov & Vohra, 2009; Pycia & Woodward, 2021) vs. heterogeneous auctions (Vries & Vohra, 2003; Ledyard, 2007; Xu et al., 2020)). The problem we investigate in this paper is a homogeneous multi-object auction design, as the auctioneer can create multiple copies of the task and assign them to different providers. However, there are two fundamental differences between our problem and the standard multi-object optimal auction design studied in the literature.

1. In available multi-object auctions, the decision is on the number or the set of objects (goods or tasks) assigned to each bidder. There is no time element in such auctions and it is often assumed that all assignments are made simultaneously. These auctions are often called *simultaneous auctions* (Zhang, 2018; Feldman et al., 2020). There is a set of auctions called sequential auctions where the decisions about distinct sets of objects are made separately and sequentially in time (Leme et al., 2012; Hosseinalipour & Dai, 2017; Donna & Espin-Sanchez, 2018; Narayan et al., 2022; Kong, 2021). The goal of this approach is to simplify the optimal auction design problem by restricting attention to a subset of objects at each round. However, none of these works consider *time* as a decision factor. Our paper is the first work that considers *allocation time* as a deciding factor in the optimal auction design. Time adds a continuous aspect to the allocation design part of the problem and hence significantly increases its complexity.

2. In available utility maximizing auctions for outsourcing problems, whether the invocations are deterministic (Iyengar & Kumar, 2008) or randomized (Alaei et al., 2012; Celis et al., 2014), they happen right after bid collection. However, in our auction, for each set of bids, the auctioneer chooses a contingent plan for how different services should be invoked over time based on whether the previously hired services are successful or not. Making contingent and interdependent decisions over time is a unique feature of our auction that has not been previously exploited for utility maximization in outsourcing problem.

The features mentioned above differentiate our work from the existing literature, making it the first work to address designing utility maximizing auctions when service procurement can take place at arbitrary points in time.

2.3 Other Related Studies

Our work is also related to the areas of automated mechanism design, robust service procurement, and contingent planning. We review the related works in these areas in Sections 2.3.1-2.3.3, respectively. We then distinguish the outsourcing problem studied in this paper from some classic control problems (such as bandit problem, prophet inequality problem, secretary problem, and Pandora's box problem) in Section 2.3.4.

2.3.1 Automated Mechanism Design

Automated mechanism design (AMD) is a process for automatically constructing mechanisms via optimization (Conitzer & Sandholm, 2003). The idea is to build a constrained optimization problem, where the desired properties (e.g., DSIC, ex-post IR) correspond to the constraints and the designer's goal (e.g., maximizing social welfare or revenue) corresponds to the objective function, and then computationally search through the space of feasible mechanisms, rather than to design them analytically by hand. The variables of such optimization problems are functions, so they are generally NP-hard and difficult to cope with (Guo & Conitzer, 2010). There is a recent approach to AMD that aims to reduce the complexity by first analyzing the domain in order to refine the formulation over a smaller feasible space, and thereby improving the scale of problems that could be solved (Guo & Conitzer, 2010; Guo et al., 2015). In this approach, optimization is often done not over all feasible mechanisms, but rather over a parameterized subfamily of mechanisms that satisfy the constraints.

Using theory to narrow the setting, then using search via a math-programming formulation, is the strategy followed in our approach, too. However, our work is a modern evolution of AMD, as unlike the previous works (Guo & Conitzer, 2010; Guo et al., 2015), it ensures optimality not over a specific class of mechanisms but rather over all feasible mechanisms.

2.3.2 Robust Service Procurement

There is a body of work that suggests the use of redundancy to overcome uncertainty. This is based on techniques that duplicate the critical components of a system in order to increase its reliability (Tillman et al., 1977; Coit & Smith, 1996). Reliability and cost are two important metrics in this area and tools are mainly focused on optimizing one of them under constraints on the other. In this paper, however, we do not impose any constraints on either of these two metrics and instead, find a redundancy structure that achieves an optimal balance between them.

Redundancy can be achieved with either a parallel or a serial configuration. The parallel redundancy, where the providers attempt the task concurrently, has seen a large amount of research (Huhns et al., 2003; Koide & Sandoh, 2009; Zhang et al., 2009). The serial redundancy, where a new service is invoked when the previous service fails or takes too long, has also been studied in (Friese et al., 2005; Oinn et al., 2006; Erradi et al., 2006). The parallel and serial redundancies are also known as redundant allocation and gradual recruitment, respectively (You & Chen, 2005). Protocols that employ gradual recruitment often use pre-defined deadlines to determine when to switch to an alternative provider (Oinn et al., 2006; Erradi et al., 2006).

A major drawback of the works within this area is that they rely on simple heuristic techniques that result in satisfactory, but far-from-optimal, redundancy structures. The works of (Lukose & Huberman, 2000) and (Glatard et al., 2007) tackle this drawback by studying when the current service should be optimally timed out to invoke a new one. However, these studies assume that only one service provider could be active at any time. This shortcoming is addressed by (Stein et al., 2011), where an algorithm for deriving the efficient combination of parallel and serial redundancy has been proposed. This algorithm gives a contingent plan for how different services should be invoked over time to maximize the social welfare, when the providers' cost information is publicly known. The authors also propose an incentive compatible and individually rational heuristic mechanism, called pairing, for settings with unknown cost information and selfish providers. However, as we show in Section 9, this heuristic mechanism is far from efficient.

Against the existing literature, our proposed approach is not heuristic, but rather systematic and guaranteed to produce optimal and efficient redundancy-based procurement auctions. Our proposed auctions are also different from the well-known first-past-the-post auctions, where the providers work on the task in parallel and then the first one that delivers the task is the only one that gets paid (Pandichi & Leon, 2011). Such auctions are not well-suited to our context because they cause regret to those that attempt the task but do not win and hence violate ex-post IR, which is one of the main properties we are interested in.

2.3.3 Contingent Planning

Contingent planning is concerned with the problem of generating contingent plans that achieve a goal in the presence of incomplete information and sensing actions (Peot & Smith, 1992; Pryor & Collins, 1996). This is one of the most general and hardest problems considered in the area of planning (Rintanen, 2004). Since the 2000s, significant progress has been made in the area, resulting in a variety of contingent planning algorithms that can solve large-scale problems (Bonet & Geffner, 2000; Hoffmann & Brafman, 2005; Bertoli et al., 2006; Albore et al., 2009; Shmaryahu et al., 2019). However, these algorithms are not able to deal with strategic agents trying to manipulate the system to their own benefit.

There are a few works that take the agents' strategic behavior into account when analyzing or designing contingent plans. Some studies have focused on specific applications such as matching (Ergin & Sarver, 2015) and exchange economies (Angeloni & Martins-da Rocha, 2009). Some others, however, provide some theoretical results that can be applied to a range of problems (Forges, 2013). For example, the work of (Forges, 2013) characterizes the set of all payoffs that can be achieved by contingent plans in a Bayesian game. However, prior to our work, there has been no unified approach for designing efficient and optimal incentive compatible contingent planning-based mechanisms for outsourcing application.

2.3.4 Distinction from Other Classic Control Problems

The outsourcing problem studied in this paper is similar in some aspects to some classic problems such as the "multi-armed bandit", "prophet inequality", "secretary", and "Pandora's box" problems), but differs in fundamental ways. Below, we will briefly discuss the main differences between our outsourcing problem and each of these problems.

1. Multi-armed bandit (MAB) problem (Katehakis & Veinott Jr, 1987): In our outsourcing problem, invoking any service provider can be interpreted as pulling an arm in a MAB problem. However, there are fundamental differences between the two problems: (1) In the MAB problem, the arms' success probability distributions are unknown, while these distributions are known to the decision maker in our problem. (2) MAB algorithms often seek to minimize the regret, while the mechanisms proposed in this work follow different objectives, such as utility or social-welfare maximizations.

Note also that the following two assumptions made in early studies on the MAB are also absent in our problem: (i) at most one arm can be active at each instant of time, and (ii) the rewards of pulling different arms at different times are independent random variables. However, these assumptions are lifted in more recent works in this area (Gupta et al., 2021; Pike-Burke et al., 2017).

2. Prophet inequality (PI) problem (Lucier, 2017): In this problem, the decision maker is faced with some random rewards, whose distributions are known upfront, but not their realizations. These realizations are revealed one-by-one in an arbitrary (i.e., adversarial (Krengel & Sucheston, 1977), random (Esfandiari et al., 2017), or best (Yan, 2011; Abolhassani et al., 2017)) order, and the goal is to give a reward-maximizing strategy that, upon seeing the value of each reward, decides either to choose or leave it. Our outsourcing problem looks a bit similar to the PI problem, as the times needed by different providers to complete the task are random variables whose distributions, and not their realizations, are known in advance. However, there is a fundamental difference between the two problems. In the PI problem, the realizations of the rewards are revealed sequentially over time and the agent can use this information to make a more informed decision. However, in our problem, even when we invoke a service provider, the realization of its service time is not revealed until the task is completed. So, the consumer cannot base its decisions on the realization of any random variable.

3. Secretary problem (Freeman, 1983): This problem is very similar to the prophet inequality problem except that the distributions of the random rewards are not known upfront. Therefore, the difference we mentioned in the previous section exists here as well. In addition, in the secretary problem, the distributions are unknown, but in our problem, they are known.

4. Pandora's box problem (Weitzman, 1979): In this problem, a decision-maker faces n boxes with known, independent distributions of their hidden rewards. To learn the reward of a box, the decision maker must pay an inspection cost, and it can choose the order of inspection. The objective is to maximize the collected reward from an inspected box minus the costs paid. This problem is different from the PI problem as (1) the information revelation is costly, and (2) the decisions are not taken in a take-it-or-leave-it fashion. The Pandora's box problem is also different from ours as invoking our providers, which is equivalent to inspecting boxes, does not reveal their exact value after a possible delay, but rather reveals some information about them gradually over time. So, how long to wait for the revealed information to lead to an optimal decision is an extra tradeoff that the decision maker faces in our problem.

3. Formal Model

A consumer C would like a task to be completed before a deadline D. The task has a value V for the consumer if it is executed before the deadline. The consumer is no longer interested in the task when the deadline is passed.[2] The consumer cannot accomplish the

2. We consider the set of domains where there is a binary model for the success or failure. A treatment for settings where the partial completion is valuable is left for future work (see Section 10).

task itself and hence needs to outsource it to other agents that are capable of performing it. There are n service providers (SPs) that can perform the task for the consumer. We denote the set of available SPs by $\mathcal{N} = \{1, \dots, n\}$.[3] The consumer may be aware of this set in advance or obtain this information when it designs an auction and calls for bids.

The time that each SP needs for executing the task is uncertain to the consumer, due to concurrent orders from other consumers, hardware or network problems and the provider's scheduling policies. The consumer often has some prior knowledge regarding the providers' service duration distributions (but not regarding their specific realizations in a given situation)[4]. However, the consumer can make this knowledge more accurate by incorporating the new information it obtains during the outsourcing process. For example, when a SP has not completed the task by a specific time, the consumer can perform a Bayesian update to obtain a posterior belief on its service duration distribution. To benefit from this gradual information revelation, the consumer may prefer to distribute the outsourcing process over time and make its decisions contingent on the information that arrives later.

The design of a contingent outsourcing plan can follow different objectives. For example, a prosocial (i.e., socially-focused) consumer might wish to maximize the sum of all participants' profits, this sum is called the social welfare, while a self-interested consumer often aims to maximize its own individual utility. The problem of finding the best contingent outsourcing plan that maximizes a specific objective would be a standard optimization problem if the costs were known to the consumer, as discussed in Section 1. However, in our case, each provider's cost is its own private information and cannot be observed by either the consumer or other providers. The consumer cannot extract the providers' costs by simply asking them, since the providers are strategic and have incentive to misreport their costs.

Dealing with intelligent and strategic providers holding private information adds a new dimension to the problem, as the consumer needs to pay an extra cost to incentivize the providers to reveal truthful information. Our goal in this paper is to propose a unified approach to design truthful contingent planning-based outsourcing mechanisms that can be used by both prosocial and self-interested consumers. In the rest of this section, we model the main components of our problem precisely. A schematic of the problem is depicted in Fig. 1. We then use these components to build up a full design in Section 4.

3.1 Uncertainty in Service Provisioning

Providers may be different from each other in terms of the speed of performing the task. The time T_i that each SP i requires to perform the task is uncertain. This uncertainty can be due to different reasons, e.g., production lines of a SP might be less or more congested at different times; resources required for performing the task could be temporarily unavailable. We model this uncertainty by assuming that the time T_i required by provider i to complete the task is not a deterministic, but a random variable obtained from a distribution with cumulative density function $G_i(t) = Prob(T_i \leq t)$. We refer to function $G_i(.)$ as provider i's duration function and assume that $G_i(0) = 0$ for all $i \in \mathcal{N}$.

3. Extending the results of this paper to settings with a dynamically-changing set of providers is an interesting direction for future work.
4. Such information can be obtained from past or shared experiences, for example from using a trust or reputation system (Stein et al., 2011)

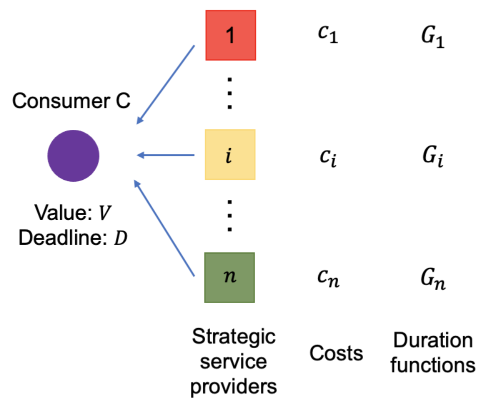


Figure 1: Schematic of the problem

Remark 1 *We make two assumptions that are commonly made about the providers' duration functions: 1) they are independent across providers, and 2) they are commonly known to both the consumer and providers. The first assumption covers settings where, for example, the computational requirements for the task are known but there is uncertainty about the load on the service provider's resources at the time of execution. The second assumption is also reasonable in settings where the duration functions can be obtained from past or shared experiences, for example from using a trust or reputation system (Stein et al., 2011). We will discuss in Section 10 what will happen if these assumptions are relaxed.*

3.2 Providers' Private Costs

Executing the task has a cost c_i for each SP i. The cost of each provider i is drawn from a regular distribution with support on the interval $\mathcal{C} = [0, c_{max}]$. The cost distribution of each SP i is said to be regular if $c_i + \frac{F_i(c_i)}{f_i(c_i)}$ is non-decreasing in c_i, where $f_i(.)$ and $F_i(.)$ are density and cumulative distribution functions of the provider i's cost. The regularity condition is satisfied by many distributions such as the uniform, normal, and Pareto distributions and is a common assumption in the auction literature (Salek & Kempe, 2008; Feng et al., 2019)[5]. The cost of each provider i is only known to itself and is considered as provider i's private information. The providers' private information is not observed by either the other providers or the consumer. However, the consumer knows the distribution F_i from which each c_i is drawn.

We assume that the costs of n providers are stochastically independent random variables.[6] Thus, the joint density function for the cost vector $\boldsymbol{c} = (c_1, \ldots, c_n)$ is $f(\boldsymbol{c}) = \prod_{i \in \mathcal{N}} f_i(c_i)$. Of course, each provider i considers its own cost c_i to be a known quantity, not a random variable. However, we assume that SP i assesses the probability distribution for the other providers' costs in the same way as the consumer does. That is, both

5. Note that as discussed in (Iyengar & Kumar, 2008), the regularity condition for procurement auctions is slightly different from that for forward auctions (Myerson, 1981). In each setting, the regularity condition is defined to facilitate the optimal auction design.
6. Extension of our results to the settings where the providers' costs are positively correlated is an interesting direction for future research (see Section 10).

the consumer and provider i assess the joint distribution function for the vector $\boldsymbol{c}_{-i} = (c_1, \ldots, c_{i-1}, c_{i+1}, \ldots, c_n)$ of costs of all providers other than i to be $f_{-i}(\boldsymbol{c}_{-i}) = \prod_{j \neq i} f_j(c_j)$.

3.3 Providers' Strategic Behavior

In order to invoke a service, the consumer has to make a payment to the provider at least equal to its cost. However, as we discussed in Section 3.2, the consumer is not aware of the providers' costs. In this paper, the providers are assumed to be self-interested and will act selfishly to maximize their own utilities. The utility of each provider i is the difference between the payment π_i it receives from the consumer and the cost c_i it incurs for performing the task. If a self-interested provider is asked to declare a cost, it may have an incentive to either "mark up" the price to earn more profit or "mark down" the price to attract the consumer.

In this situation, the consumer should design its outsourcing strategy and the payments such that the providers cannot benefit by misreporting their costs. This can be done by designing an incentive mechanism[7](Brubaker, 1980). The consumer announces the mechanism's rules in advance and then each provider seeks an action that maximizes its own utility. The mechanism must be designed such that truth telling is the best strategy for each provider.

3.4 Consumer's Objective

The consumer may seek to achieve a number of different objectives. However, two of the most common are: (1) maximizing the consumer's expected utility, and (2) maximizing the expected social welfare (Zhan, 2008). The first objective is often pursued by self-interested consumers while the second objective is more suitable for socially-focused consumers.

Utility Maximization: The consumer's utility is the difference between the value it gets from the task (if any) and all the payments it should make. The consumer gets a value V from the task if it is executed successfully before the deadline. Faster and more reliable SPs, which can increase the chance of getting the task done before the deadline, are often more expensive to hire. Therefore, the goal of a self-interested consumer that seeks to maximize its own utility is to balance the tradeoff between the success value and the invocation cost.

Social Welfare Maximization: Social welfare is defined as the total utility of the consumer and all service providers. Each service provider's utility is the difference between the payment it receives and its incurred cost. Summing up the utilities of all providers and the consumer (discussed in the previous paragraph), we obtain the social welfare as the difference between the value of task completion and the cost of task execution. Thus, the goal of the consumer that seeks to maximize the social welfare is to balance the tradeoff between the success value and the execution (not invocation) cost.

At first glance the difference between the two above-mentioned objectives might seem to be small, as they only differ in the cost that they assign to the providers' invocations. However, upon careful examination, it can be seen that the difference is important and may have significant effect on the mechanism design process, as the invocation cost (which

7. We define the concept of incentive mechanisms formally in Section 5.

appears in the first objective) is a function of the payments, which are known to and designed by the consumer, while the execution costs (which appear in the second objective) are unknown to the consumer and are privately held by the providers.

In this paper, we propose a unified approach to contingent planning-based incentive mechanism design that can be used to maximize either of the above-mentioned objectives. However, before that, we must formalize the concept of contingent outsourcing planning in Section 4. We then are ready to begin our incentive mechanism design process from Section 5. In Section 5, constructing a mapping from contingent outsourcing plans to randomized simultaneous outsourcing strategies, we propose a payment scheme that guarantees truthful information elicitation from self-interested providers at the minimum cost. Using this payment scheme, we design a utility maximizing contingent planning-based outsourcing incentive mechanism in Section 6. Analyzing the mechanism's properties in Section 7, we then discuss how the same approach can be used to design a social welfare maximizing mechanism in Section 8.

4. Contingent Outsourcing Planning

4.1 Informal Definition and Importance

Outsourcing is a partially observable problem: the consumer can observe whether the task has been delivered or not, but cannot observe how much time each SP needs to complete the task. Operating in this partially observable environment, the consumer can gain important information by direct experience, i.e., by recruiting a provider i and waiting for a period of time τ to see how it acts. If the SP does not complete the task within the given time, the consumer can perform a Bayesian update and obtain the distribution of T_i conditioned on $T_i > \tau$. Using this posterior distribution, the consumer can then decide whether this SP is likely to complete the task on time or it is better to assign the task to a new SP. In the latter case, both SPs will attempt the task in parallel.

Making outsourcing plans contingent on the information revealed over time is called the contingent outsourcing planning. This method takes advantage of the following two outsourcing techniques: redundant allocation and gradual recruitment. Redundant allocation is an approach to increase the task success probability by procuring multiple SPs to attempt the task in parallel (Ha & Kuo, 2006). Gradual recruitment is an extension to redundant allocation which lets the consumer distribute the recruitment process over time (Shen & Xie, 1990). In both of these techniques, the consumer obtains value V if at least one of the providers completes the task before the deadline D. Completing the task by more than one SP does not provide any additional value to the consumer. However, the consumer cannot stop hired providers from performing the task when the task is completed by one SP or the deadline is reached. This means that the task is assumed to be non-interruptible and needs to run to completion once it started.

Recruiting more providers increases the success probability, but it also increases both (i) the execution cost, as more resources are allocated to task execution, and (ii) the invocation cost, as the consumer should make a payment to each hired provider to incentivize it to perform the task upon recruitment. Therefore, as shown in the example below, the consumer should decide wisely which outsourcing plans lead to a better balance between the success probability and any of the costs.

Example 1 *Suppose that a high-valued task can be performed by two SPs. Provider* 1 *is cheap to hire, but its delivery time varies based on its workload. Under a low workload (prob.* α*), provider* 1 *delivers the task after* $D/2$ *units of time. However, it needs at least* $2D$ *time-units to deliver the task when working at a high workload (prob.* $1-\alpha$*). The second provider is expensive to hire, but it is totally reliable and always delivers the task by time* $D/2$*.*

In this situation, if the consumer knew provider 1*'s workload, the optimal decision would be to recruit provider* 1 *when the workload is low and to recruit provider* 2 *when the workload is high. This information is not available to the consumer beforehand. However, the consumer can build a cheap experiment based on the gradual recruitment technique to first find out provider* 1*'s workload and then make the decision accordingly. Specifically, the consumer can first recruit provider* 1 *at a low cost and give it* $D/2$ *units of time to perform the task. Whether or not provider* 1 *delivers the task by* $D/2$ *serves as a signal to the consumer to discover provider* 1*'s workload. That is, provider* 1*'s workload is low if and only if the job is delivered by time* $D/2$*. If based on this information, the workload is revealed to be high, the consumer can spend more money to guarantee success by recruiting provider* 2*. This contingent outsourcing plan provides the same value to the consumer as the one that recruits provider* 2 *from the beginning (i.e., the probability of success is* 1 *in both cases), however, it yields a lower execution cost if* $c_1/c_2 < \alpha$*, i.e., the ratio of the providers' costs is less than the probability of provider* 1*'s workload being low. Therefore, this contingent outsourcing plan is efficient for a consumer whose goal is to balance the tradeoff between success value and execution cost.* □

We formalize the idea of a contingent outsourcing plan in the next subsection. We then start developing a technique to find optimal contingent outsourcing plans that maximize a specific goal when providers are self-interested in Section 5.

4.2 Formal Definition

Definition 1 (Contingent outsourcing plan) *A contingent outsourcing plan is a vector* $\rho = ((s_1, \tau_1), \ldots, (s_m, \tau_m))$*, where each element represents the time* $\tau_k \in [0, D]$ *that provider* $s_k \in \mathcal{N}$ *is recruited if and only if the task has not been completed so far. The elements of this vector are all different in their first component (i.e.,* $s_k \neq s_{k'}$ *if* $k \neq k'$*) and ordered non-decreasingly based on their second component* τ_k*, i.e.,* $\tau_{k+1} \geq \tau_k$*, for all* $k \in \{1, \ldots, m-1\}$*. The vector size* m *can be any arbitrary integer between* 0 *and* n*.*

A contingent outsourcing plan ρ specifies an ordered set of SPs $\rho_s = (s_1, \ldots, s_m)$ that are candidates to be invoked, as well as their invocation time $\rho_\tau = (\tau_1, \ldots, \tau_m)$. The consumer starts by recruiting the provider with the first turn, i.e., s_1, at time τ_1 and waits up to time τ_2 to see if the task is completed. If the task is not completed, the consumer increases the success probability by invoking provider s_2, which is second in the line, to attempt the task in parallel with s_1. This gradual recruitment continues until either the task is completed or all the candidate providers have been invoked.

Remark 2 *In this paper, for ease of presentation, we restrict our attention to contingent plans with "pure" actions, where the consumer invokes a specific provider* s_k *at each decision* *point τ_k if the task has not been delivered thus far. However, our analysis can be extended to randomized settings, where the invoked provider is selected from a predetermined distribution on $\mathcal{N}$, in a straightforward manner.*

We denote the set of all contingent outsourcing plans with pure actions by $\mathcal{H}$. This set is general enough to include the non-contingent plans (also known as "simultaneous recruitment strategies"), where the recruitments happen simultaneously at an arbitrary time τ (i.e. $\rho = ((s_1, \tau), \ldots, (s_m, \tau))$) without being contingent on a specific event. A special case of such non-contingent plans where the consumer recruits only one single provider (i.e., $\rho = ((s_1, \tau_1))$ is often called a"single recruitment strategy" (Stein et al., 2011).

The consumer would like to choose an outsourcing plan $\rho \in \mathcal{H}$ that optimizes its objective function, which typically is either its own expected utility or the expected social welfare. In the next two subsections, we quantify the consumer's expected utility and the expected social welfare for any arbitrary outsourcing plan $\rho = ((s_1, \tau_1), \ldots, (s_m, \tau_m))$. Then, in Sections 5-8, we design incentive mechanisms that optimize these two objective functions.

4.3 Consumer's Expected Utility from an Outsourcing Plan

We denote the consumer's expected utility when it employs the outsourcing plan ρ and makes payments $\boldsymbol{\pi} = (\pi_i)_{i \in \mathcal{N}}$ to providers upon their recruitment, by

$$U_{con}(\rho, \boldsymbol{\pi}) = V(\rho, D) - C_I(\rho, \boldsymbol{\pi}), \tag{1}$$

where $V(\rho, D)$ is the expected value the consumer gets from task completion and $C_I(\rho, \boldsymbol{\pi})$ is the expected invocation cost. All expectations are over the providers' service duration distributions T_i, $i = 1, \ldots, n$. Below, we will first quantify the expected value and the expected invocation cost for an arbitrary outsourcing plan $\rho = ((s_1, \tau_1), \ldots, (s_m, \tau_m))$. We then subtract these two quantities to derive the consumer's expected utility.

4.3.1 Expected Value of Task Completion

The consumer gets a value V if the outsourcing plan ρ is successful in delivering the task before deadline D. An outsourcing plan ρ is successful if at least one of the recruited SPs completes the task before the deadline. We define the completion time of the task when outsourcing plan ρ is employed as the first completion time for providers ρ_s and denote it by $T_\rho = \min_{k \in \{1,\ldots,m\}} (\tau_k + T_{s_k})$. The task completion time T_ρ is a random variable whose distribution depends on the duration functions of the recruited providers. Since each T_{s_k}, for $s_k \in \rho_s$, is drawn independently from distribution G_{s_k}, the cumulative distribution function of T_ρ can be derived as follows:

$$G_\rho(t) = Prob(T_\rho \leq t) = 1 - Prob(T_\rho > t) = 1 - \prod_{k=1}^{m} (1 - G_{s_k}(t - \tau_k)). \tag{2}$$

Based on (2), the success probability of outsourcing plan ρ for a task with deadline D can be formulated as

$$P_{succ}(\rho, D) = Prob(T_\rho \leq D) = 1 - \prod_{k=1}^{m} (1 - G_{s_k}(D - \tau_k)). \tag{3}$$

We can use (3) to derive the expected value of an outsourcing plan ρ as

$$V(\rho, D) = VP_{succ}(\rho, D) = V(1 - \prod_{k=1}^{m}(1 - G_{s_k}(D - \tau_k))). \tag{4}$$

4.3.2 Expected Invocation Cost

The consumer makes a payment π_i to each provider i upon recruitment. Therefore, the expected invocation cost of an outsourcing plan ρ when employed along with the payment strategy $\boldsymbol{\pi}$ can be written as

$$C_I(\rho, \boldsymbol{\pi}) = \sum_{i=1}^{n} P_i(\rho)\,\pi_i, \tag{5}$$

where $P_i(\rho)$ is the probability that provider i will be invoked in the outsourcing plan ρ. Providers out of the set of candidate providers ρ_s have no chance for being hired, i.e., $P_i(\rho) = 0$, where $i \notin \rho_s$. However, each provider $i \in \rho_s$ will be recruited at a specific time if all the providers with earlier turns fail to complete the task by then.

To formalize this, we define the order of provider i's invocation in the outsourcing plan ρ as $o_i(\rho)$, where

$$o_i(\rho) = \begin{cases} k, & \text{if} \quad s_k = i, \\ 0, & \text{if} \quad i \notin \rho_s. \end{cases} \tag{6}$$

Any provider $i \in \rho_s$ will be invoked at time $\tau_{o_i(\rho)}$ if all the providers j with $o_j(\rho) < o_i(\rho)$ fail to complete the task by $\tau_{o_i(\rho)}$. Since the providers' duration functions are independent random variables, we can derive provider i's invocation probability as

$$P_i(\rho) = \prod_{j:o_j(\rho)<o_i(\rho)} Prob(\tau_{o_j(\rho)} + T_j > \tau_{o_i(\rho)}) = \prod_{j:o_j(\rho)<o_i(\rho)} (1 - G_j(\tau_{o_i(\rho)} - \tau_{o_j(\rho)})), \tag{7}$$

where $i \in \rho_s$. It can be seen from (7) that the providers' invocation probabilities are inversely related to their invocation orders; providers with earlier turns have higher chances of being recruited.

By substituting (7) in (5), we can derive the expected invocation cost as

$$C_I(\rho, \boldsymbol{\pi}) = \sum_{i\in\rho_s} \pi_i \,(\prod_{j:o_j(\rho)<o_i(\rho)} (1 - G_j(\tau_{o_i(\rho)} - \tau_{o_j(\rho)}))). \tag{8}$$

4.3.3 Consumer's Expected Utility

Combining the results of Subsections 4.3.1 and 4.3.2, we can obtain a closed form expression for the consumer's expected utility, as follows:

$$\begin{aligned} U_{con}(\rho, \boldsymbol{\pi}) &= VP_{succ}(\rho, D) - \sum_{i=1}^{n} P_i(\rho)\,\pi_i \\ &= V(1 - \prod_{k=1}^{m}(1 - G_{s_k}(D - \tau_k))) - \sum_{i\in\rho_s} \pi_i \,(\prod_{j:o_j(\rho)<o_i(\rho)} (1 - G_j(\tau_{o_i(\rho)} - \tau_{o_j(\rho)}))). \end{aligned} \tag{9}$$

4.4 Expected Social Welfare of an Outsourcing Plan

We denote the expected social welfare when outsourcing plan ρ employed by

$$SW(\rho) = V(\rho, D) - C_E(\rho), \tag{10}$$

where $V(\rho, D)$ is the expected value of task completion and $C_E(\rho)$ is the expected execution cost. The expected value $V(\rho, D)$ is previously derived in Section 4.3.1. Therefore, what we need to do in order to derive the expected social welfare is to compute the expected execution cost $C_E(\rho)$ and substitute it in (10).

4.4.1 Expected Execution Cost

The expected execution cost $C_E(\rho)$ is the expected cost the service providers incur to execute the task. Each provider i incurs a cost c_i to execute the task if it gets invoked, which happens with probability $P_i(\rho)$. Therefore, we have

$$C_E(\rho) = \sum_{i=1}^{n} P_i(\rho)\, c_i. \tag{11}$$

4.4.2 Expected Social Welfare

Substituting (4) and (11) into (10) gives us the following closed form expression for the social welfare:

$$SW(\rho) = V(1 - \prod_{k=1}^{m}(1 - G_{s_k}(D - \tau_k))) - \sum_{i \in \rho_s} c_i \,(\prod_{j: o_j(\rho) < o_i(\rho)} (1 - G_j(\tau_{o_i(\rho)} - \tau_{o_j(\rho)}))). \tag{12}$$

We can see that the expected social welfare (12) and the consumer's expected utility (9) are similar. The difference is that each payment π_i in (9) is replaced by the actual service cost c_i in (12). This difference is fundamental, however, because unlike the payments that are decision variables and can be determined by the consumer inline with its objective, the actual service costs are fixed and also unknown to the consumer.

Our goal in this paper is to propose a unified contingent planning-based incentive mechanism design approach that can be used to maximize the two aforementioned objective functions. We start working towards this goal by focusing on maximizing the consumer's expected utility in Sections 5-7. We then discuss in Section 8 how the same approach with different parameters can be used to design a social welfare maximizing incentive mechanism.

5. Contingent Planning-based Incentive Mechanism Design

The purpose of this section and the next is to design a contingent planning-based incentive mechanism to maximize the consumer's expected utility. We achieve this goal by proceeding as follows. In Section 5.1, we first introduce the two key components of an incentive mechanism (i.e., allocation function and payment function). We then formulate the consumer's problem as a mechanism design problem and describe our design goals. In Section 5.2, we derive the payment function of an optimal contingent planning-based incentive mechanism by constructing a mapping from contingent outsourcing plans into randomized simultaneous outsourcing strategies. This payment function achieves our desired features at the minimum possible cost. Using this payment scheme, we design the allocation function of an optimal contingent planning-based incentive mechanism in Section 6.

5.1 Mechanism Design Problem

As discussed in Section 3.2, the service costs are privately held by the self-interested service providers and may be falsely reported if it increases the providers' utilities. Each provider may even decide to not contribute its service to the consumer if it finds it more beneficial. In order to avoid such situations, which would lead to a non-optimal outsourcing plan, the consumer should provide sufficient incentives to the providers, through an incentive mechanism, to motivate providers' voluntary participation and truth-telling. Ensuring these two properties are known as individual rationality and incentive compatibility, respectively (Börgers, 2015).

Incentive mechanisms can have many different forms. However, it has proven to be without loss of generality to restrict attention to direct procurement auctions, where the service providers directly bid their costs (Börgers, 2015). In a direct procurement auction, each provider i's bid b_i must belong to the set of possible costs $\mathcal{C}$ and can be interpreted as provider i's reported cost. Notice that each provider i's reported cost is not necessarily equal to its actual cost c_i. We denote the vector of all providers' submitted bids by $\boldsymbol{b} = (b_1, \ldots, b_n) \in \mathcal{C}^n$.

A procurement auction $M = (A(.), \boldsymbol{\pi}(.))$ is defined by an allocation function $A(.)$ and a vector of payment functions $\boldsymbol{\pi}(.) = (\pi_1(.), \ldots, \pi_n(.))$. The allocation function $A : \mathcal{C}^n \to \mathcal{H}$ determines the outsourcing plan $A(\boldsymbol{b}) \in \mathcal{H}$ that is selected by the consumer based on the received bids $\boldsymbol{b}$, and each payment function $\pi_i : \mathcal{C}^n \to \mathbb{R}_+$ determines the amount of money $\pi_i(\boldsymbol{b})$ that will be paid to provider i, upon recruitment. As stated before, in Sections 5-7, we focus on settings where the consumer aims to maximize its own expected utility. In such settings, the consumer's objective is equivalent to designing a procurement auction that (1) satisfies incentive compatibility, (2) satisfies individual rationality, and (3) maximizes the consumer's expected utility. In the following subsections, we derive conditions on the auction's allocation function $A(.)$ and payment function $\boldsymbol{\pi}(.)$ that guarantee each of these properties. We then conclude this section by formulating the consumer's auction design problem as an optimization problem.

5.1.1 Incentive Compatibility (IC)

Any procurement auction $M = (A(.), \boldsymbol{\pi}(.))$ induces a non-cooperative game with incomplete information among service providers, where each provider i's strategy is a mapping from its true cost c_i to the bid b_i it submits. We say that a procurement auction is incentive compatible (IC) if it motivates providers to bid their costs truthfully, in each game induced by the auction. However, different forms of IC can be defined based on the conditions under which this requirement is satisfied. In this paper, we investigate two notions of IC that are applicable to our setting: Bayesian IC (BIC) and Dominant Strategy IC (DSIC).

Procurement auction M is said to be Bayesian incentive compatible (BIC) if truth-telling (i.e., $b_i = c_i, \forall i \in \mathcal{N}$) is a best strategy for each provider, given its belief, whenever the other providers also reveal their true costs (Börgers, 2015), i.e.,

$$U_i(A, \pi_i, c_i, c_i) \geq U_i(A, \pi_i, c_i, b_i), \quad \forall c_i, b_i \in \mathcal{C}, i \in \mathcal{N}, \tag{13}$$

where

$$U_i(A, \pi_i, c_i, b_i) = \int P_i(A(b_i, \boldsymbol{c}_{-i}))[-c_i + \pi_i(b_i, \boldsymbol{c}_{-i})] f_{-i}(\boldsymbol{c}_{-i}) d\boldsymbol{c}_{-i}, \tag{14}$$

is provider i's expected utility in auction $M = (A(.), \boldsymbol{\pi}(.))$ when it submits bid b_i, which could be different from its true cost c_i, and all other providers report their costs truthfully. This definition of IC is equivalent to requiring that truth-telling be a Bayesian Nash equilibrium (BNE) in each game induced by the auction (Börgers, 2015). However, it does not violate the possibility of existence of other Bayesian Nash equilibria. Each game induced by an auction may have multiple equilibria, however, the BIC condition only requires that the truth-telling strategy should be one of them.

As a stronger notion of incentive compatibility, auction M is said to satisfy DSIC if truth-telling is a best strategy for each provider irrespective of what other providers do, i.e.,

$$P_i(A(c_i, \boldsymbol{c}_{-i}))[-c_i + \pi_i(c_i, \boldsymbol{c}_{-i})] \geq P_i(A(b_i, \boldsymbol{c}_{-i}))[-c_i + \pi_i(b_i, \boldsymbol{c}_{-i})], \tag{15}$$

for all $c_i, b_i \in \mathcal{C}, \boldsymbol{c}_{-i} \in \mathcal{C}^{n-1}, i \in \mathcal{N}$. DSIC is a more desirable property than BIC as: (i) Unlike BIC, each provider's best strategy is independent of its belief about the distribution of other providers' costs. Thus, in DSIC implementation, providers can have inconsistent and incorrect beliefs about the common prior without influencing the outcome (Jaekel, 2019). (ii) Unlike BIC, it ensures that no other equilibrium can be more beneficial than truth-telling to the providers. Therefore, providers have a strong reason to always stick to the truth-telling strategy.

Remark 3 *For the definitions of BIC and DSIC, we do not need to worry about mixed strategies. The reason is that each provider's expected utility from a mixed strategy is the weighted average of its utilities from pure actions, where the weight of each utility is the probability of selecting the corresponding action. It is clear that the weighted average of a set of numbers cannot be larger than the largest value in the set. Therefore, if there is no pure action that outperforms truth-telling, there would also be no mixed strategy that does so.*

5.1.2 Individual Rationality (IR)

An auction is individually rational (IR) for a provider, if the utility it gains from the auction is non-negative. There are different notions of IR in the literature; the two we focus on here are called *interim IR* and *ex-post IR*. The difference between these two notions is at the time the providers are allowed to drop out of the auction. In interim IR, providers should make their decisions before the auction starts. However, in ex-post IR, the providers are allowed to opt out of the auction even at the very end when the consumer decides to recruit them.

The interim IR is guaranteed if the expected utility each provider i with cost c_i gains from participating in the auction is non-negative, i.e.,

$$U_i(A, \pi_i, c_i, c_i) \geq 0, \quad \forall c_i \in \mathcal{C}, i \in \mathcal{N}. \tag{Interim IR}$$

However, the ex-post IR requires that no provider has regrets regarding participation even if any bid vector $\boldsymbol{c}_{-i}$ is submitted by other providers, i.e.,

$$\pi_i(c_i, \boldsymbol{c}_{-i}) - c_i \geq 0, \quad \forall c_i \in \mathcal{C}, \boldsymbol{c}_{-i} \in \mathcal{C}^{n-1}, i \in \mathcal{N}. \tag{Ex-post IR}$$

Ex-post IR is a stronger requirement than interim IR and hence any mechanism that satisfies ex-post IR is also interim individually rational.

5.1.3 Utility Maximization

In Section 4, we derived the consumer's expected utility $U_{con}(\rho, \boldsymbol{\pi})$ when it employs the outsourcing plan ρ and makes payments $\boldsymbol{\pi} = (\pi_i)_{i\in\mathcal{N}}$ to providers upon their recruitment. In the truthful equilibrium of the game induced by auction $M = (A(.), \boldsymbol{\pi}(.))$, each bid vector $\boldsymbol{c}$ is submitted with probability $f(\boldsymbol{c})$. When bid vector $\boldsymbol{c}$ is received, the consumer employs the outsourcing plan $A(\boldsymbol{c})$ and the payment strategy $\boldsymbol{\pi}(\boldsymbol{c})$. Therefore, the consumer's expected utility in an incentive compatible auction $M = (A(.), \boldsymbol{\pi}(.))$ is

$$\begin{aligned} U(A, \boldsymbol{\pi}) &= \int U_{con}(A(\boldsymbol{c}), \boldsymbol{\pi}(\boldsymbol{c})) f(\boldsymbol{c}) d\boldsymbol{c} \\ &= \int [V P_{succ}(A(\boldsymbol{c}), D) - \sum_{i\in\mathcal{N}} P_i(A(\boldsymbol{c}))\pi_i(\boldsymbol{c})] f(\boldsymbol{c}) d\boldsymbol{c}. \end{aligned} \tag{16}$$

In (16), the expectation is over both the providers' service durations and the providers' service costs, as the service costs are also unknown to the consumer.

The consumer would like to choose the allocation function A and the payment function $\boldsymbol{\pi}$ so as to maximize the expected utility $U(A, \boldsymbol{\pi})$ under the strong notions of IC and IR constraints (i.e., DSIC and ex-post IR). However, these notions are very demanding and difficult to deal with. Therefore, in the design process, we relax these constraints to the BIC and interim IR constraints. Then, we will theoretically show that the auction is sufficiently well-designed to satisfy the DSIC and ex-post IR conditions, as well. Note that in addition to simplifying the design process, this approach has another clear advantage as well. The auction we will design with this approach will be an auction whose performance is optimal not only among auctions that satisfy strong notions of DSIC and ex-post IR but also among a wider group of auctions that only satisfy weaker requirements of BIC and interim IR.

With this explanation, we formulate the relaxed version of the problem faced by a utility maximizing consumer as follows:

$$\max_{\{A, \boldsymbol{\pi}\}} \quad U(A, \boldsymbol{\pi}) \tag{17a}$$

$$\textbf{s.t.} \quad U_i(A, \pi_i, c_i, c_i) \geq 0, \quad \forall c_i \in \mathcal{C}, i \in \mathcal{N}, \tag{17b}$$

$$U_i(A, \pi_i, c_i, c_i) \geq U_i(A, \pi_i, c_i, b_i), \quad \forall c_i, b_i \in \mathcal{C}, i \in \mathcal{N}. \tag{17c}$$

The rest of this section as well as Section 6 are devoted to solving optimization problem (17).

5.2 Characterization of BIC and Interim IR Contingent planning-based Auctions

The variables in optimization problem (17) are the functions $A(.)$ and $\boldsymbol{\pi}(.)$ defined on $\mathcal{C}^n$ implying the dimensionality of the problem is infinite[8]. Such problems are generally difficult to solve and require special analytical or numerical methods (Gelfand & Fomin, 2012; Saadatmandi & Dehghan, 2008; Maleki & Mashali-Firouzi, 2010). In mechanism design literature, such optimization problems are often simplified by applying Myerson's approach that characterizes the set of auctions that satisfies the BIC and interim IR constraints. This technique has been developed for simultaneous auctions, where the decisions are taken at one point in time, and not over time (Myerson, 1981). In our problem, however, the outsourcing happens gradually over time and can include decisions that are contingent on the outcome of the past recruitments.

To adapt Myerson's approach to our setting, we construct a mapping from the set of all contingent planning-based auctions to the set of all randomized simultaneous auctions. As we will prove in Lemma 1, this mapping keeps the providers' expected utilities and hence the BIC and interim IR constraints (17b)-(17c) invariant. Therefore, we can use the results of Myerson for randomized simultaneous auctions to characterize the set of BIC and interim IR contingent planning-based auctions.

Definition 2 (Equivalent simultaneous auction) *Let ψ be a mapping that maps each outsourcing plan $\rho = ((s_1, \tau_1), \ldots, (s_m, \tau_m))$ into a randomized simultaneous outsourcing strategy $\psi(\rho)$ that recruits each SP s_k at time 0 with probability $P_{s_k}(\rho)$ (see (7)). We use mapping ψ to define the simultaneous equivalent of each contingent planning-based auction $M = (A(.), \boldsymbol{\pi}(.))$ as a randomized simultaneous auction $\hat{M} = (\hat{A}(.), \boldsymbol{\pi}(.))$, where $\hat{A}(\boldsymbol{c}) = \psi(A(\boldsymbol{c}))$, for all $\boldsymbol{c} \in \mathcal{C}^n$.*

Lemma 1 *The expected utility of each SP i is the same in each contingent planning-based auction $M = (A(.), \boldsymbol{\pi}(.))$ and its simultaneous equivalent $\hat{M} = (\hat{A}(.), \boldsymbol{\pi}(.))$, i.e.,*

$$U_i(A, \pi_i, c_i, b_i) = U_i(\hat{A}, \pi_i, c_i, b_i), \quad \forall i \in \mathcal{N}, \forall c_i, b_i \in \mathcal{C}. \tag{18}$$

We present the proofs of all the lemmas and propositions in Appendices A-K.

We can now use the Myerson's results for simultaneous auctions to characterize the BIC and interim IR contingent planning-based auctions. To this end, we define

$$Q_i(A, b_i) = \int P_i(A(b_i, \boldsymbol{c}_{-i})) f_{-i}(\boldsymbol{c}_{-i}) d\boldsymbol{c}_{-i}, \tag{19}$$

as the conditional probability that service provider i will be hired in auction $M = (A(.), \boldsymbol{\pi}(.))$ given that it submits bid b_i.

Lemma 2 *A contingent planning-based auction $M = (A(.), \boldsymbol{\pi}(.))$ satisfies the interim IR and BIC constraints if and only if the following conditions hold:*

(I1) The conditional probability function $Q_i(A, b_i)$ defined in (19) is decreasing in its second argument, i.e.,

$$(b_i - c_i)(Q_i(A, b_i) - Q_i(A, c_i)) \leq 0; \tag{20}$$

8. The dimension of an optimization problem is the number of its decision variables.

(I2) *Each provider i's expected utility when it tells the truth satisfies the two following conditions:*

$$U_i(A, \pi_i, c_i, c_i) = U_i(A, \pi_i, c_{max}, c_{max}) + \int_{c_i}^{c_{max}} Q_i(A, b_i) db_i, \tag{21}$$

$$U_i(A, \pi_i, c_{max}, c_{max}) \geq 0. \tag{22}$$

Based on Lemma 2, $M = (A(.), \boldsymbol{\pi}(.))$ represents an optimal contingent planning-based auction if and only if it maximizes $U(A, \boldsymbol{\pi})$ subject to (20)-(22). Using this result, we can derive an explicit closed form expression for the optimal payment function.

Lemma 3 *Given the monotone allocation function A, the lowest payment that entices service providers to participate in the auction and bid truthfully is*

$$\pi_i^A(\boldsymbol{b}) = b_i + \int_{b_i}^{c_{max}} \frac{P_i(A(\hat{b}_i, \boldsymbol{b}_{-i}))}{P_i(A(\boldsymbol{b}))} d\hat{b}_i, \quad \forall i \in \mathcal{N}. \tag{23}$$

The payment scheme proposed in (23), which we call the *weighted threshold payment scheme*, is a two-part tariff. The first part is the bid submitted by the provider, and the second part is called an information rent. The information rent is due to the asymmetry of information and will be paid to a service provider in exchange for accurate disclosure of its private information.

Based on (23), the information rent of each provider i is a weighted integral over the provider's possible bid values. The weight of each bid $\hat{b}_i \geq b_i$ is the conditional probability that i got hired if it reported $\hat{b}_i$, given that it is recruited when it submits bid b_i. We denote this weight by

$$d^A_{b_i, \boldsymbol{b}_{-i}}(\hat{b}_i) = P_i(A(\hat{b}_i, \boldsymbol{b}_{-i}))/P_i(A(\boldsymbol{b})), \tag{24}$$

and call it relative desirability of bid $\hat{b}_i$ for provider i over b_i.

5.3 Formulating Problem (17) in terms of the Allocation Function

Lemma 3 reduces the complexity of designing an optimal auction by expressing the optimal payment function $\boldsymbol{\pi}(.)$ in terms of the allocation function $A(.)$. This result helps us to focus on optimizing the allocation rule $A(.)$ under the monotonicity condition (20) and then use equation (23) to derive the minimum payments required to make the auction with this allocation function BIC and interim IR.

Following this approach, in the next lemma, we state the optimal auction design problem solely in terms of the allocation function $A(.)$.

Lemma 4 *Suppose that the allocation function $A : \mathcal{C}^n \to \mathcal{H}$ solves the following optimization problem:*

$$\max_{A} \quad \int [V P_{succ}(A(\boldsymbol{c}), D) - \sum_{i \in \mathcal{N}} (c_i + \frac{F_i(c_i)}{f_i(c_i)}) P_i(A(\boldsymbol{c}))] f(\boldsymbol{c}) d\boldsymbol{c}, \tag{25a}$$

$$\textbf{\textit{s.t.}} \quad (b_i - c_i)(Q_i(A, b_i) - Q_i(A, c_i)) \leq 0, \quad \forall i \in \mathcal{N}, c_i, b_i \in \mathcal{C}. \tag{25b}$$

Then, $M = (A(.), \boldsymbol{\pi}^A(.))$, where $\boldsymbol{\pi}^A(.)$ is derived based on (23), represents the utility maximizing contingent planning-based auction.

The objective function of problem (25) is similar in form to the consumer's expected utility (16), except that each payment $\pi_i(\boldsymbol{c})$ is replaced by a virtual cost $\phi_i(c_i) = c_i + \frac{F_i(c_i)}{f_i(c_i)}$. These virtual costs are the consumer's estimation of the payments it should make to the providers. We see by definition that the providers' virtual costs are always greater than or equal to their actual costs, i.e., $\phi_i(c_i) \geq c_i$. This basically means that the cost that a utility maximizing consumer evaluates for recruiting each provider i is higher than its actual cost. The reason is simple. According to (23), the consumer should pay each recruited provider i not only its actual cost c_i, but also an information rent to induce the provider to disclose its private information. The difference between the virtual cost and the real cost of each provider is the expectation of the information rent that must be paid to the provider to reveal truthful information.

The purpose of the next section is to develop an algorithm to solve optimization problem (25). However, before we finish this section, two remarks should be made.

Remark 4 *The mapping ψ we introduced in Definition 2 keeps the providers' expected utilities invariant (see Lemma 1), but does not do the same for consumer's expected utility. As shown by Example 2 in Appendix D, a contingent planning-based auction can produce a much higher utility to the consumer compared to its equivalent simultaneous auction. This is why we do not restrict our attention to simultaneous auctions and instead allow the possibility of contingent decision making.*

Remark 5 *The optimal payment function* (23) *provides a good insight on why contingent planning can reduce the expected invocation cost $C_I(\rho, \boldsymbol{\pi}) = \sum_{i=1}^{n} P_i(\rho)\pi_i$ (see Section 4.3.2). The reason of this is two-fold: (i) reducing providers' invocation probabilities $P_i(\rho)$ and (ii) reducing the prices π_i the consumer needs to pay to providers upon recruitment. The first reason is obvious from the discussion in Section 4. When the consumer employs a gradual recruitment strategy, it can stop the process as soon as one of the hired providers completes the task. This reduces the invocation probabilities of the providers with later turns, and accordingly reduces the consumer's expected cost. The second reason is less apparent and originates from the optimal payment function* (23)*. This is illustrated by Example 3 in Appendix E.*

6. Optimal Allocation Function

The purpose of this section is to find an allocation function $A : \mathcal{C}^n \to \mathcal{H}$ that solves optimization problem (25). For each i, c_i and b_i, the monotonicity constraint (25b) depends on the outsourcing plan that allocation function A assigns not to a single cost vector, but rather to a set of cost vectors $\{(c_i, \boldsymbol{c}_{-i}), (b_i, \boldsymbol{c}_{-i}), \boldsymbol{c}_{-i} \in \mathcal{C}^{n-1}\}$. This interaction among decision variables makes optimization problem (25) non-separable across cost vectors and hence difficult to solve. However, the next lemma proves that the monotonicity constraint (25b) is satisfied "for free" at the optimal allocation.

Lemma 5 *When the providers' cost distributions $F_1, \ldots, F_n$ are regular, i.e., virtual costs are increasing in costs, the allocation function that maximizes* (25a) *is monotone and satisfies constraint* (25b)*.*

Lemma 5 allows us to relax the monotonicity constraint and be sure that this condition will be automatically satisfied at the final solution. If we drop the monotonicity constraint (25b), the problem becomes separable and can be decomposed into the following unconstrained subproblems $\eta(\boldsymbol{c})$, $\boldsymbol{c} \in \mathcal{C}^n$:

$$\eta(\boldsymbol{c}) : \max_{A(\boldsymbol{c}) \in \mathcal{H}} \quad V P_{succ}(A(\boldsymbol{c}), D) - \sum_{i \in \mathcal{N}} P_i(A(\boldsymbol{c}))\phi_i(c_i).$$

For each cost vector $\boldsymbol{c}$, problem $\eta(\boldsymbol{c})$ aims to find the optimal outsourcing plan $A^*(\boldsymbol{c}) = ((s_1, \tau_1), \ldots, (s_m, \tau_m))$ that specifies the optimal ordered set of providers $A^*_s(\boldsymbol{c}) = (s_1, \ldots, s_m)$ and their optimal invocation times $A^*_\tau(\boldsymbol{c}) = (\tau_1, \ldots, \tau_m)$. This problem is a mixture of continuous and combinatorial optimization problems. Computing the optimal procurement times $A^*_\tau(\boldsymbol{c})$ for each ordered set of providers $A_s(\boldsymbol{c})$ is a continuous optimization problem. However, determining the optimal ordered set $A^*_s(\boldsymbol{c})$ is a combinatorial optimization problem. Combinatorial optimization problems can be viewed as searching for the best element of some set of discrete items. Therefore, in principle, any sort of search algorithm can be used to solve them. However, the number of ordered subsets of n providers can, in practice, become quite large (e.g. 1,956 for $n = 6$ providers but 1,302,061,344 for $n = 12$ providers). Exploring such a huge search space is computationally prohibitive and needs a carefully constructed algorithm.

We proceed as follows to solve optimization problem $\eta(\boldsymbol{c})$ for each cost vector $\boldsymbol{c}$. In Subsection 6.1, we focus on the continuous part of $\eta(\boldsymbol{c})$ and discuss how to derive the optimal invocation times $A^*_\tau(\boldsymbol{c}) = (\tau_1, \ldots, \tau_m)$ for a fixed provider sequence $A_s(\boldsymbol{c})$. We make use of this result in Subsection 6.2 to develop a branch-and-bound algorithm for solving the combinatorial part of the problem.

6.1 Continuous Part: Optimal Invocation Times

In this part, we assume that the optimal subset of providers and their ordering is given. That is, we are given an ordered set of providers $A_s(\boldsymbol{c}) = (s_1, \ldots, s_m)$, where s_k is invoked before s_{k+1}. To compute the optimal invocation time, we must determine $A_\tau(\boldsymbol{c}) = (\tau_1, \ldots, \tau_m)$, where τ_k is the invocation time of s_k, such that the objective function of problem $\eta(\boldsymbol{c})$ gets maximized.

Using (3) and (7), we can write the objective function of $\eta(\boldsymbol{c})$ in terms of s_k and τ_k, $k = 1, \ldots, m$, as follows:

$$f(\tau) = V(1 - \prod_{k=1}^{m} (1 - G_{s_k}(D - \tau_k))) - \sum_{k=1}^{m} \phi_{s_k}(c_{s_k}) \prod_{k'=1}^{k-1} (1 - G_{s_{k'}}(\tau_k - \tau_{k'})). \tag{26}$$

To derive the optimal invocation time, we must solve an optimization problem that maximizes $f(\tau)$ under the following constraints: $\forall k : 0 \leq \tau_k \leq D$ and $\forall k, k' : k < k' \implies \tau_k \leq \tau_{k'}$. This is a continuous non-convex optimization problem. There is currently no general algorithm to find a global optimum of a non-convex optimization problem[9], however, there

9. There are some algorithms in the literature that prove the global optimality of their solutions under some specific assumptions (Dixon & Szego, 1978; Locatelli & Schoen, 2013), however, those assumptions are not valid in our problem.

are some widely used numerical methods, such as stochastic gradient descent (Robbins & Monro, 1951) and saddle-free Newton (Dauphin et al., 2014), that use interesting techniques to escape from local optima and/or saddle points[10]. Our optimization problem here can be solved by any of those numerical methods. We denote the function that returns the optimal invocation time for provider sequence $(s_1, \ldots, s_m)$ by $Times(s_1, \ldots, s_m)$.

6.2 Combinatorial Part: Optimal Provider Sequence

One of the most universally applicable approaches for reducing the search space in combinatorial optimization problems is the branch-and-bound family of algorithms. Such algorithms recursively split the search space into smaller groups; this splitting is called branching. Branching alone would amount to brute-force enumeration of candidate solutions and testing them all. To improve on the performance of brute-force search, a branch-and-bound algorithm keeps track of a lower bound and an upper bound for all solutions in given groups. The algorithm uses these bounds to prune the search space, eliminating groups that it can prove will not contain an optimal solution (i.e., its upper bound is less than the lower bound of another group). This systematic partitioning of the search space and removing the groups that cannot contain an optimal solution can potentially reduce the space of solutions that have to be searched. However, the performance of a branch-and-bound algorithm depends crucially upon the appropriate selection of branching and bounding techniques.

In this paper, we adopt the branching and bounding techniques proposed in (Stein et al., 2011). We briefly discuss each of these techniques below.

Branching technique: The main idea is to partition the search space, which is the set of all ordered subsets of providers $\mathcal{N} = \{1, \ldots, n\}$, based on their first r elements. r is a parameter that starts from 1 and increases gradually over iterations. The pseudo-code of this algorithm is presented in Algorithm 1. In the first iteration, the search space is partitioned into n groups $\{\langle 1 \rangle, \langle 2 \rangle, \ldots, \langle n \rangle\}$ (Line 3), where each $\langle i \rangle$ represents the set of orderings that have i as their first element. For example, for $n = 3$, we have $\langle 1 \rangle = \{(1), (1,2), (1,3), (1,2,3), (1,3,2)\}$. After this partitioning, the algorithm calculates a lower bound and an upper bound for the best consumer's utility within each group (Lines 5-6) and keeps track of the highest lower bound found so far (Lines 7-10). We will discuss how the Lower and Upper functions work later. The highest lower bound U_{low} serves as a touchstone for assessing the quality of the groups to be explored in the future. A group will be discarded if its upper bound is lower than or equal to U_{low} (Lines 11-13).

In the second iteration, the algorithm picks the group with the highest lower bound, as this group is likely to have a better chance to contain the optimal ordering, and partitions it into $n-1$ smaller-sized groups based on the first two elements (Lines 16-18). The function $Expand(o)$ in Line 18 sub-partitions group o based on the next unfixed element. That is, if the orderings within o match in their first r elements, the function $Expand(o)$ partitions them based on their $(r+1)$-th elements. The aim of this function is to create the next generation of groups to be explored more precisely. For example, if $\langle 1 \rangle$ is picked at Line 16 when $n = 3$, the next generation of groups will be $Expand(\langle 1 \rangle) = \{\langle 1, 2 \rangle, \langle 1, 3 \rangle\}$. These

10. The random stochastic gradient descent algorithm achieves this goal by adding a random noise to the gradient and the saddle-free newton algorithm accomplishes this objective by using the Hessian to find a descent direction.

Algorithm 1: Branch-and-Bound Algorithm

1 **Input:** Cost vector $\boldsymbol{c}$, Duration functions $\{G_i(.)\}_{i \in \mathcal{N}}$, Deadline D;
2 Let $o^*(\boldsymbol{c}) = \emptyset, U_{low} = 0$;
3 $Q = \{\langle 1 \rangle, \langle 2 \rangle, \ldots, \langle n \rangle\}$;
4 **for** *all* $o' \in Q$ **do**
5     $\underline{u} \leftarrow Lower(o')$;
6     $\bar{u} \leftarrow Upper(o')$;
7   **if** $\underline{u} > U_{low}$ **then**
8         $o^*(\boldsymbol{c}) \leftarrow o'$;
9         $U_{low} \leftarrow \underline{u}$;
10   **end**
11   **if** $\bar{u} \leq U_{low}$ **then**
12         $Q \leftarrow Q - \{o'\}$;
13   **end**
14 **end**
15 **while** $Q \neq \emptyset$ **do**
16     $o \leftarrow \arg\max_{o \in Q} Lower(o)$;
17     $Q \leftarrow Q - \{o\}$;
18     $P \leftarrow Expand(o)$;
19   **for** *all* $o' \in P$ **do**
20         $\underline{u} \leftarrow Lower(o')$;
21         $\bar{u} \leftarrow Upper(o')$;
22   **if** $\underline{u} > U_{low}$ **then**
23             $o^*(\boldsymbol{c}) \leftarrow o'$;
24             $U_{low} \leftarrow \underline{u}$;
25   **end**
26   **if** $\bar{u} \leq U_{low}$ **then**
27             $P \leftarrow P - \{o'\}$;
28   **end**
29   **end**
30     $Q \leftarrow \{x \in Q \cup P | Upper(x) > U_{low}\}$;
31 **end**
32 $A_s^*(\boldsymbol{c}) \leftarrow o^*(\boldsymbol{c})$;
33 $A_\tau^*(\boldsymbol{c}) \leftarrow Times(o^*(\boldsymbol{c}))$;
34 **Output: Optimal outsourcing plan** $A^*(\boldsymbol{c}) = (A_s^*(\boldsymbol{c}), A_\tau^*(\boldsymbol{c}))$

new groups will be evaluated based on the highest potential utility they can provide for the consumer and will be discarded if their upper bound is below U_{low} (Lines 19-29). The groups that survive from this generation along with the ones that survive from the first generation (Line 30) go for another round of branching and bounding. This process continues until there is no branchable group left.

Lower bounding technique: Deriving a lower bound for the highest expected utility of the orderings within a group is simple. This is because the best expected utility provided by any member of the group can be interpreted as a lower bound for the maximum utility the whole group can provide to the consumer.

The groups we consider in our branch-and-bound algorithm are of the form $\langle i_1, i_2, \ldots, i_j \rangle$, where $j \leq n$, and $i_s \in \mathcal{N}$, for all $1 \leq s \leq j$. Each group $o = \langle i_1, i_2, \ldots, i_j \rangle$ is the set of all orderings that start with $i_1, \ldots, i_j$. We select ordering $R(o) = (i_1, \ldots, i_j) \in o$ as a representative for group o. In ordering $R(o)$, the consumer invokes $i_1, \ldots, i_j$ in turn and does not invoke any other providers. The optimal invocation time for ordering $R(o)$ is $(t_1^*, \ldots, t_j^*) = Times(R(o))$, where $Times(.)$ is the function derived in Section 6.1. Using this result, we can construct the best outsourcing plan with the ordering $R(o)$ as $\rho^*(o) = ((i_1, t_1^*), \ldots, (i_j, t_j^*))$. We consider the consumer's expected utility when employing outsourcing plan $\rho^*(o)$ as the lower bound for the highest expected utility members of the group o can provide to the consumer. That is,

$$Lower(o) = VP_{succ}(\rho^*(o), D) - \sum_{i \in \mathcal{N}} P_i(\rho^*(o))\phi_i(c_i). \tag{27}$$

Upper bounding technique: Calculating an upper bound on the expected utility that members of a group can provide for the consumer is not immediately obvious. Obtaining a tight upper bound requires computing the utility provided by every member of the group, which is too cumbersome to be practical. However, we can obtain a looser (but still effective) upper bound by applying the following approach.

For each group $o = \langle i_1, i_2, \ldots, i_j \rangle$, we define

$$\mathcal{N}'(o) = \{i \in \mathcal{N} | i \notin \{i_1, i_2, \ldots, i_j\}\}. \tag{28}$$

If $\mathcal{N}'(o) = \emptyset$, then $(i_1, i_2, \ldots, i_j)$ is the only member of group o. Therefore, the upper bound is equal to the lower bound discussed above. Otherwise, we create a virtual service provider s_o that dominates all subsets of $\mathcal{N}'(o)$. This service provider has the minimum cost among all providers $\mathcal{N}'(o)$, i.e., $c_{s_o} = \min_{i \in \mathcal{N}'(o)} c_i$, and can perform the task faster than any combination of providers $\mathcal{N}'(o)$, i.e., $G_{s_o}(x) = 1 - \prod_{i \in \mathcal{N}'(o)} (1 - G_i(x))$. Therefore, invoking s_o is strictly better than invoking any subset of providers $\mathcal{N}'(o)$. With this reasoning, we obtain a new ordering $\bar{o} = (i_1, i_2, \ldots, i_j, s_o)$ by appending s_o to o and then calculate the upper bound as the maximum utility obtained by employing outsourcing plan $\rho^*(\bar{o}) = (\bar{o}, Times(\bar{o}))$, if this utility is higher than the lower bound $Lower(o)$. Failure to meet this condition indicates that it is not possible to achieve a higher utility by invoking further providers; therefore, we can set the upper bound equal to the lower bound. That is,

$$Upper(o) = \max\left(Lower(o), VP_{succ}(\rho^*(\bar{o}), D) - \sum_{i \in \mathcal{N}} P_i(\rho^*(\bar{o}))\phi_i(c_i)\right). \tag{29}$$

Running Algorithm 1 with the lower and upper bounding techniques described above, we can derive the optimal ordering $o^*(\boldsymbol{c})$ for recruiting providers $\mathcal{N}$ with submitted cost vector $\boldsymbol{c}$. Based on the discussions in Section 6.1, the optimal time for hiring these providers can be computed as $Times(o^*(\boldsymbol{c}))$. Therefore, we can denote the optimal outsourcing plan as $A^*(\boldsymbol{c}) = (o^*(\boldsymbol{c}), Times(o^*(\boldsymbol{c})))$ (Lines 32-34).

7. The Optimal Contingent Planning-based Auction

The allocation function $A^*(.)$, presented in Section 6, along with the weighted threshold payment function $\boldsymbol{\pi}^{A^*}(.)$, derived in Section 5.2, specifies our proposed auction $M_{OCPA} = (A^*(.), \boldsymbol{\pi}^{A^*}(.))$. We call this an optimal contingent planning-based auction (OCPA) and the general structure of OCPA is as follows (see Fig. 2):

1. **Auction's rules announcement:** The consumer publicly announces that the allocation function $A^*(.)$ and the payment function $\boldsymbol{\pi}^{A^*}(.)$ are going to be used during the auction.

2. **Bidding:** Each provider i analyzes the auction's rules and submits a bid b_i that best serves its interest;

3. **Determining the outsourcing plan and payment strategy:** Based on the submitted bids, the consumer selects the outsourcing plan $A^*(\boldsymbol{b})$ and the payment strategy $\boldsymbol{\pi}^{A^*}(\boldsymbol{b})$;

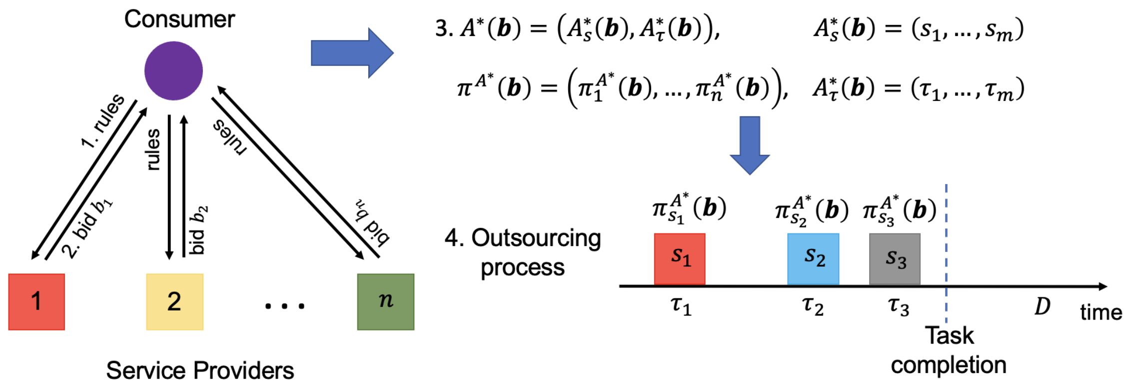


Figure 2: The general structure of the OCPA auction. This auction is composed of the following four steps: Step 1 (Auction's rules announcement), Step 2 (Bidding), Step 3 (Determining the outsourcing plan and payment strategy), and Step 4 (Outsourcing process). These steps are explained thoroughly in Section 7.

4. **Outsourcing process:** The consumer recruits providers according to the ordering $A_s^*(\boldsymbol{b})$ at times $A_\tau^*(\boldsymbol{b})$, until either the task is completed or the deadline is reached. Each provider i receives a payment $\boldsymbol{\pi}_i^{A^*}(\boldsymbol{b})$ once it is recruited.

As we discussed in Section 5.2, the weighted threshold payment function $\boldsymbol{\pi}^{A^*}(.)$ is designed to guarantee BIC and interim IR. Moreover, the allocation function $A^*(.)$ is designed to provide the highest possible utility to the consumer. We elaborate these arguments in the next subsection to present and prove the fundamental properties of OCPA. Our discussion in Section 7.1 will show that OCPA satisfies not only BIC and interim IR, but also stronger notions of DSIC and ex-post IR.

7.1 Fundamental Properties of OCPA

In this section, we prove theoretically that our proposed OCPA auction satisfies all desirable properties we were looking for. That is, the auction is dominant strategy incentive compatible, ex-post individually rational, and utility maximizing. We also prove that OCPA can always guarantee a non-negative expected utility for the consumer.

Proposition 1 (Dominant Strategy Incentive Compatibility) *The truth-telling bidding strategy is optimal for each provider irrespective of what the other providers bid. That is, providers cannot benefit by misreporting their private costs.*

Proposition 2 (Ex-Post IR) *OCPA is ex-post individually rational. That is, the recruited providers perform the task upon recruitment and do not regret their decision.*

Proposition 3 (Utility Maximization) *OCPA provides the consumer with the maximum expected utility that any BIC and interim IR auction could provide.*

Propositions 1-3 show that the OCPA auction is a DSIC and ex-post IR auction that has no competitors in terms of the consumer's expected utility even among the auctions that satisfy weaker requirements of BIC and interim IR. This property is due to the constraint relaxation approach we followed in this paper, which has been discussed in Section 5.1.3.

Proposition 4 (Non-negative Expected Utility) *The consumer's expected utility is always non-negative in OCPA. Therefore, it is rational for the consumer to participate in OCPA.*

7.2 Time Complexity

Propositions 1-3 prove that our proposed OCPA auction achieves the main properties we were looking for. However, this is at the cost of high time complexity to find the optimal allocation function using Algorithm 1. In Algorithm 1, we tried to reduce the time complexity by using a branch-and-bound algorithm instead of brute-force search. However, this algorithm still searches for the optimal solution and in the worst case, it may need to examine the entire search space. This may be the case, for example, when there are large numbers of highly similar providers and when the value of the task is very large in relation to the service costs.

To reduce the worst-case complexity, in the next subsection, we present a low-complexity heuristic algorithm (i.e., Algorithm 2) that can be substituted for Algorithm 1. As we will show in Section 9, this algorithm reduces the runtime by over 99% with less than 1% performance loss.

7.3 Heuristic Algorithm for Deriving a Near-Optimal Allocation Function

Algorithm 2 is based on a greedy search and aims to find a near optimal ordering for recruiting providers. It starts with an empty ordering and then greedily adds, removes or switches providers until a local optimum is reached. Intuitively, this algorithm benefits from selecting providers that offer a good value/cost trade-off. By also allowing providers to be removed or switched, it has some backtracking capabilities — thus an expensive but reliable provider can eventually be replaced by many cheap and unreliable providers that individually do not yield a high expected utility, but in combination result in a better strategy.

In more detail, the algorithm stores the best ordering found so far and the expected utility it can provide to the consumer by $o_{apx}(\boldsymbol{c})$ and U_{best}, respectively (Line 2). The algorithm also sets a flag that takes value 1 once a local optimum is found (Line 3). In each round, the algorithm checks to see if ordering $o_{apx}(\boldsymbol{c})$ has a neighbor that provides better utility for the consumer (Lines 5-7). If such neighbors exist, the algorithm replaces $o_{apx}(\boldsymbol{c})$ by its best neighbor and goes to the next round (Lines 7-9). This process continues until the best ordering found so far has no better neighbor and hence is a local optimum.

For the algorithm description to be complete, we need to define the neighborhood of an ordering. The larger the neighborhood is, the better the quality of the returned solution is, but at the same time the longer it takes to search the neighborhood at each iteration (Line 6). As an extreme case, if all orderings are defined to be neighbors, the algorithm will find the exact optimum ordering in just one round, but at the cost of doing an exhaustive search over all orderings.

Algorithm 2: Heuristic Algorithm (Alternative to Algorithm 1)

Input: Cost vector $\boldsymbol{c}$, Duration functions $\{G_i(.)\}_{i \in \mathcal{N}}$, Deadline D;
Let $o_{apx}(\boldsymbol{c}) = \emptyset, U_{best} = 0$;
$flag = 0$;
while $flag = 0$ **do**
  $P \leftarrow Neighbors(o_{apx}(\boldsymbol{c}))$;
  $o' \leftarrow \arg\max_{o \in P} Lower(o)$;
  if $Lower(o') > U_{best}$ **then**
    $o_{apx}(\boldsymbol{c}) \leftarrow o'$;
    $U_{best} \leftarrow Lower(o')$;
  else
    $flag \leftarrow 1$;
  end
end
Output: Near-optimal ordering $o_{apx}(\boldsymbol{c})$

In this paper, we define the neighborhood of an ordering o as the set of all orderings that can be obtained by any of the three following actions: 1) selecting a provider i which is currently not in o and adding it to o at any possible position, 2) selecting a provider $i \in o$ and removing it, 3) selecting two providers $i, j \in o$ and swapping their turns. We will show in Section 9 that this definition of neighborhood is wide enough to ensure the final solution is close to the exact optimum. At the same time, it is narrow enough to enable the algorithm to scale to realistic settings with hundreds of providers.

We close this section by Proposition 5. This proposition proves that replacing Algorithm 1 with Algorithm 2 does not ruin the incentive compatibility or individual rationality of the auction. The only difference is a significant complexity reduction at the expense of less than one percent decrease in the consumer's expected utility, as will be shown in Section 9.3.

Proposition 5 (Auction with Heuristic Allocation Function) *Our proposed sub-optimal contingent planning-based auction (SOCPA)* $M_{SOCPA} = (A_{apx}(.), \boldsymbol{\pi}^{A_{apx}}(.))$, *where* $A_{apx}(\boldsymbol{c}) = (o_{apx}(\boldsymbol{c}), Times(o_{apx}(\boldsymbol{c})))$ *is derived by running Algorithm 2, satisfies ex-post and interim individual rationality as well as DSIC.*[11]

8. Social-Welfare Maximizing (Efficient) Auction

In Sections 5-7, we developed two contingent planning-based auctions, one optimal and one low-complexity sub-optimal, with the goal of maximizing the consumer's expected utility. The aim of this section is to describe how the same approach with different parameters can be used to design DSIC and ex-post IR social welfare maximizing auctions.

11. Note that as discussed in the proof of Proposition 1, monotonicity of the allocation function is a necessary condition for DSIC. Therefore, the main step in the proof of Proposition 5, which is provided in Appendix K, is to prove the monotonicity of the heuristic allocation function $A_{apx}(.)$.

In Section 4.4, we derived the social welfare $SW(\rho)$ for an arbitrary outsourcing plan ρ. In the truthful equilibrium of the game induced by an auction $M = (A(.), \boldsymbol{\pi}(.))$, each bid vector $\boldsymbol{c}$ is submitted with probability $f(\boldsymbol{c})$. When bid vector $\boldsymbol{c}$ is received, the consumer employs the outsourcing plan $A(\boldsymbol{c})$. Thus, the expected social welfare in an incentive compatible auction $M = (A(.), \boldsymbol{\pi}(.))$ is

$$\begin{aligned} SW(A) &= \int \mathbb{E}\left[SW(A(\boldsymbol{c}))\right] f(\boldsymbol{c}) d\boldsymbol{c} \\ &= \int [V P_{succ}(A(\boldsymbol{c}), D) - \sum_{i \in \mathcal{N}} P_i(A(\boldsymbol{c})) c_i] f(\boldsymbol{c}) d\boldsymbol{c}. \end{aligned} \tag{30}$$

The problem of designing a social welfare maximizing auction is equivalent to maximizing (30) under the DSIC and ex-post IR constraints. Similar to what we have done for designing utility maximizing auctions, we first relax the constraints to BIC and interim IR, which are easier to cope with. We then show that the social welfare maximizing auction designed by taking this approach satisfies the stronger requirements of DSIC and ex-post IR.

In Section 5.2, we characterized the set of all BIC and IR incentive mechanisms. In particular, we first proved that the BIC and IR constraints (17b)-(17c) can be replaced by constraints (20)-(22). We then proposed the weighted threshold payment scheme (23) as the cheapest payment scheme that can guarantee (21)-(22) for a monotone allocation function. In the social welfare maximizing auction design problem, the objective function (30) does not depend on the payments. Therefore, although we do not need to minimize the payments and can employ any payment scheme that guarantees (21)-(22), there is nothing to prevent us from using the cheapest payment scheme (23) that satisfies the constraints.

Using (23) as the payment function, the problem is now reduced to that of determining an allocation function $A(.)$ that maximizes the social welfare (30) under the monotonicity constraint (20). This problem is exactly similar to problem (25) except that in the objective function, the virtual cost $\phi_i(c_i) = c_i + \frac{F_i(c_i)}{f_i(c_i)}$ of each provider $i \in \mathcal{N}$ is replaced by its actual service cost c_i. So, the problem of this section can be solved by using the branch-and-bound algorithm proposed in Section 6 where in both the Lower and Upper functions, each virtual cost $\phi_i(c_i)$ is replaced by its corresponding actual service cost c_i.

In more detail, let $\tilde{A}$ denote the social welfare maximizing (i.e., efficient) allocation function derived by taking this approach. Then, we can denote our proposed efficient contingent planning-based auction (ECPA) as $M_{ECPA} = (\tilde{A}(.), \boldsymbol{\pi}^{\tilde{A}}(.))$. By taking a similar approach to that used in the proof of Propositions 1-3, we can show that ECPA is interim and ex-post individually rational, dominant strategy incentive compatible, and social welfare maximizing. However, similar to OCPA, it is of high computational cost. Therefore, we use a similar approach to that of Section 7.3, to design a sub-efficient contingent planning-based auction (SECPA) that significantly reduces the runtime of ECPA with only marginal performance loss.

9. Numerical Results

In this paper, we proposed two utility maximizing (i.e., OCPA, SOCPA) and two social welfare maximizing (i.e., ECPA, and SECPA) contingent planning-based auctions for the outsourcing problem defined in Section 3. We theoretically proved that these auctions are able to incentivize providers to voluntarily participate in the mechanism and reveal their private information truthfully. We also proved that OCPA and ECPA maximize the consumer's expected utility and the expected social welfare, respectively.

In this section, we want to augment our theoretical results by demonstrating the behavior of our proposed auctions in a variety of simulated environments and evaluating their performance in terms of efficiency and robustness, compared with the existing methods. In more detail, the purpose of our evaluation is four-fold. First, we investigate how the contingent planning method enables our proposed auctions to adapt their outsourcing strategies to different situations (Section 9.1). Second, we demonstrate the performance improvements of our proposed auctions compared to the available benchmarks (Section 9.2). Third, we compare the performance of our proposed sub-optimal auctions SOCPA and SECPA with their corresponding optimal versions and show that these auctions significantly reduce the running time without noticeably reducing the performance (Section 9.3). Finally, we check the robustness of our results in Section 9.4.

Note that as the social welfare maximizing auctions ECPA and SECPA are developed by using a similar approach to that for the utility maximizing auctions OCPA and SOCPA, to avoid duplication and save space, unless specified we only report the evaluation results of the utility maximizing auctions. However, the results of the social welfare maximizing auctions are almost the same.

9.1 Behavior of Optimal Outsourcing Plans

The optimum point of tradeoff between success and invocation cost varies from situation to situation and is greatly influenced by parameters such as the value V of task completion and the deadline D. For example, the higher the value of the task, the more money the consumer is willing to spend to reduce the risk of failure. On the other hand, when the task is more urgent, the consumer may place more weight on the providers' speeds rather than their costs, when evaluating providers. Therefore, the optimal outsourcing plan should be flexible and adapts itself to different situations. In this section, we study how the contingent outsourcing planning method we used in OCPA helps the consumer to achieve this goal. We are also interested to find out how optimal outsourcing plans behave in different environments (e.g., what direct experiences will be purposeful and how much time gaps there should be between different experiences).

To do this, we run a simulation with 100 service providers where each provider i's delivery time has an exponential distribution with cumulative distribution function (cdf) $G_i(t) = 1 - e^{\lambda_i t}$.[12] We call parameter λ_i the service rate of provider i. In this experiment, we assume, for ease of demonstration, that the service rates are spaced equally over the interval (0,1], i.e., $\lambda_i = 0.01i$. We also assume that the service cost is an identity function of the service rate, i.e., $c_i = \lambda_i$, for all $i = 1, \dots, 100$. However, the nature of the results does not depend on these assumptions.

To study a range of environments, we consider four different simulation settings where the task has either a low ($V_{low} = 4$) or a high value ($V_{high} = 10$) and the deadline is either

12. As it has been shown in (Stein et al., 2011), the exponential distribution enables us to derive the optimal invocation times analytically. So, there will be no risk of getting stuck at a local optimum.

normal ($D_{normal} = 3$) or urgent ($D_{urgent} = 1$). In each of these settings, we run Algorithm 1 to derive the utility maximizing outsourcing plan. The results of this experiment are shown in Fig. 3. We can see that in Setting 1 where the task is high-valued and has a normal deadline (i.e., Fig. 3a), it is optimal for the consumer to start the outsourcing process by hiring a relatively fast provider ($\lambda = 0.74$) and wait for 0.83 time-units to see if it can perform the task alone. If it cannot, the consumer should recruit some auxiliary providers to increase success likelihood. This phase of recruitment starts with hiring slow but cheap providers and then the closer the consumer is to the deadline, the faster and more expensive providers are hired. Notice that since the completion of the task has a high value for the consumer, it hires a very fast and expensive provider ($\lambda = 0.9$) when the deadline is approaching to greatly increase the chance of success.

We can see from Fig. 3b that in Setting 2 where the task is low-valued and has a normal deadline, the optimal outsourcing plan has two main differences with that of Setting 1: (i) the first recruited provider is slightly slower ($\lambda = 0.6$) and has been given a longer period of time to try the task alone (1.08 time-units); (ii) if the first provider is not successful, the consumer recruits only two cheap auxiliary providers to increase the chance of success. In this case, the task completion is not valuable enough for the consumer to recruit any more expensive providers.

Comparing Figs. 3c and 3d with Figs. 3a and 3b, respectively, shows how the deadline affects the optimal outsourcing plan. In fact, when the task is urgent, it is better for the consumer to advance the outsourcing process by spending more money and recruit faster providers sooner. This basically means that the gradual recruitment technique is of less interest when the task has a tight deadline.

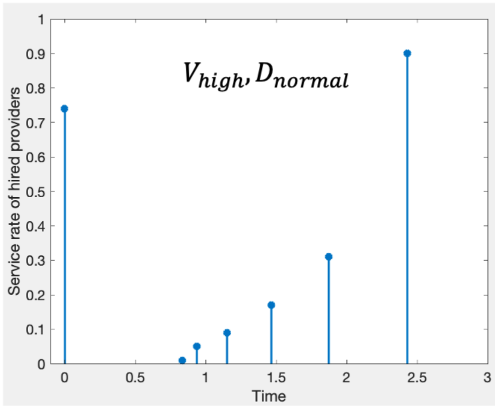
This result motivates us to study the efficiency and necessity of using redundant allocation and gradual recruitment techniques in not just a specific example (i.e., a continuum of providers with $\lambda_i = c_i$), but in more general environments. To this end, for each fixed V and D, we run 1000 simulations where in each simulation the number of providers n is chosen uniformly from $[2, 20]$ and the costs and service rates are drawn independently and uniformly from $[0, 1]$ [13]. To simulate practical settings where faster services are often more costly to procure, in all future simulations (unless otherwise stated), we construct each provider i by assigning both the i^{th} highest service cost and the i^{th} highest service rate to it. We will discuss in Section 9.4, how the results change if this correlation between the service costs and service rates does not exist.

In each simulation, we use the following metrics to measure the efficiency of the redundant allocation and gradual recruitment techniques:

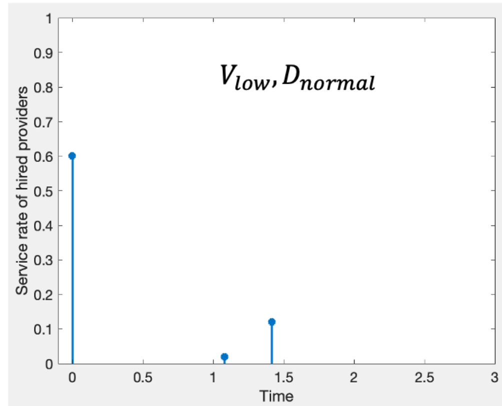
- Number of total candidates (m): For each outsourcing plan $\rho = ((s_1, \tau_1), \ldots, (s_m, \tau_m))$, m is the number of providers that have the chance to be recruited over time.

- Number of hired providers (m_h): The outsourcing process continues until at least one of the providers completes the task and hence not all of the candidate providers $(s_1, \ldots, s_m)$ are recruited in each simulation. We denote by m_h the number of providers that are actually hired in a single simulation.

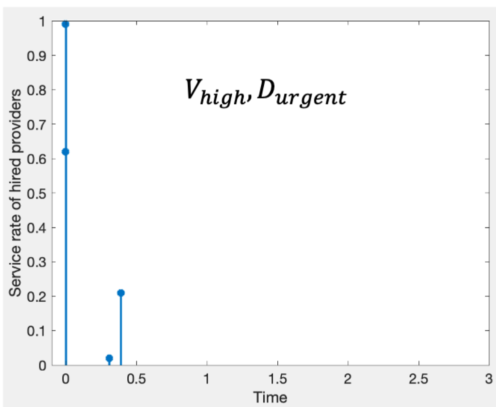
13. This technique has been previously used in (Stein et al., 2011) for generating random environments.



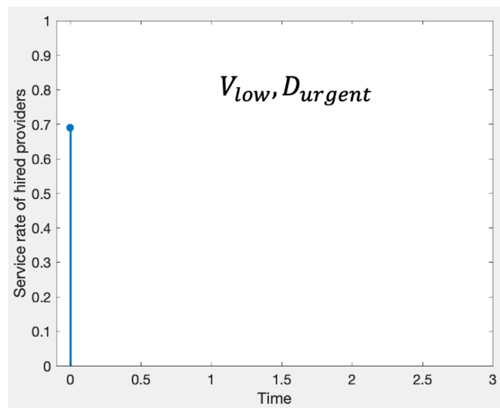
(a) Setting 1: $V = 10$, $D = 3$



(b) Setting 2: $V = 4$, $D = 3$



(c) Setting 3: $V = 10$, $D = 1$

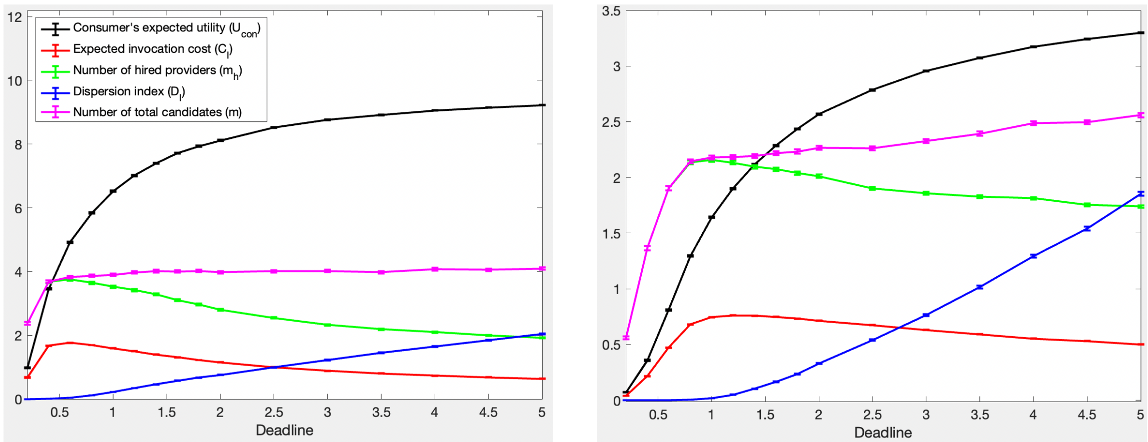


(d) Setting 4: $V = 4$, $D = 1$

Figure 3: Utility maximizing outsourcing plans when 100 providers with equally-spaced service rates $\lambda \in (0, 1]$ are available.

- Dispersion index (D_I): For each outsourcing plan $\rho = ((s_1, \tau_1), \ldots, (s_m, \tau_m))$, we define the dispersion index as $D_I = \sigma^2/\mu$, where σ^2 and μ are the variance and mean of the recruitment times. This metric, which is also called variance-to-mean ratio (VMR), is a standard measure to quantify whether a set of observed occurrences are clustered or dispersed compared to a standard statistical model (Akbarov & Wu, 2012; Gentillon et al., 2016). Low values of D_I shows that the recruitment is more centered around time 0, while high values of D_I shows that the recruitment is distributed over a longer period of time.

In Fig. 4, we plot these metrics as well as the expected utility and the expected invocation cost for the optimal outsourcing plan versus the deadline. We can see that whether the task is high-valued or low-valued, the dispersion index increases when the deadline is relaxed. This shows that the benefit of using the gradual recruitment technique is highest when the task is less urgent and hence the consumer has more time for recruitment. We can also see that the number of total candidates increases, with a steep change at a certain threshold. This threshold is 0.4 when $V = 10$ and equals 0.8 when $V = 4$. The number



(a) High-valued task ($V = 10$) (b) Low-valued task ($V = 4$)

Figure 4: The efficiency metrics for redundant allocation and gradual recruitment techniques

of hired providers, however, increases up to the same threshold and decreases afterwards. Before the threshold, the number of hired candidates is almost the same as the number of candidate providers. This shows that before the threshold, redundant allocation is the dominant technique; in this region, the consumer prefers to recruit multiple providers but also prefers to recruit all of them almost simultaneously at time 0. This fact is also confirmed by small values of D_I before the threshold. However, after the threshold, the importance of gradual recruitment becomes more apparent. In this region, the consumer finds it beneficial to reduce its costs by postponing the hiring of some of the providers.

9.2 Performance Against the Benchmarks

In this section, we compare the expected utilities obtained by OCPA and SOCPA to those obtained by the benchmarks (i.e., Bm1-Bm4). We also compare our results to the maximum theoretical utility that could be achieved if full information was available to the consumer (Bm5). This theoretical upper bound does not have practical usage in our setting as the cost information is not available to the consumer and must be elicited from self-interested providers.

(Bm1) **Best single auction:** This auction is optimal (i.e., utility maximizing) among non-redundancy-based auctions that assign the task to just one single provider. Such auctions are similar to single-object auctions discussed in Section 2.2 and hence the optimal among them can be derived by using Myerson's idea (Myerson, 1981).

(Bm2) **Best simultaneous auction:** This auction is optimal among redundancy-based auctions that do not employ gradual recruitment technique. Such auctions recruit a set of providers simultaneously at time 0 to attempt the task in parallel. Auction Bm2 is the solution of a standard homogeneous multi-object auction design problem which can be solved by techniques proposed in (Malakhov & Vohra, 2009).

(Bm3) **Random gradual auction** This auction uses both redundancy and gradual recruitment, but in a random manner. In this auction, the consumer recruits a random subset of providers at random times. The payment to each recruited provider is the maximum bid that it could have submitted, i.e., c_{max}. The random gradual auction is incentive compatible as both the allocation and payment functions are independent of the submitted bids and hence the outcome is non manipulable by the providers.

(Bm4) **Pairing mechanism (Stein et al., 2011):** This contingent planning-based auction is designed to obtain a high social welfare. The pairing mechanism is currently the state-of-the-art in the class of approximate social welfare maximizing incentive mechanisms.

(Bm5) **Best full information mechanism:** This mechanism provides the maximum utility in full-information environments, where the providers' cost function is fully available to the consumer. Therefore, it serves as a theoretical upper bound for the performance of any mechanism in the incomplete-information setting.

Comparing OCPA to these benchmarks allows us to quantify the benefit of using contingent planning in an optimal and principled manner to deal with execution uncertainty. We conduct the comparison by running 1000 simulations with randomly generated providers in each of the four simulation settings described in Section 9.1 and use Tukey's test (Keselman & Rogan, 1977) to ensure statistical significance[14]. The expected utilities the consumer obtains in these simulations for Settings 1-4 are shown in Figs. 5a-5d, respectively. In each figure, the error bars show the standard deviations of repeated measurements. Below, we will analyze the results obtained.

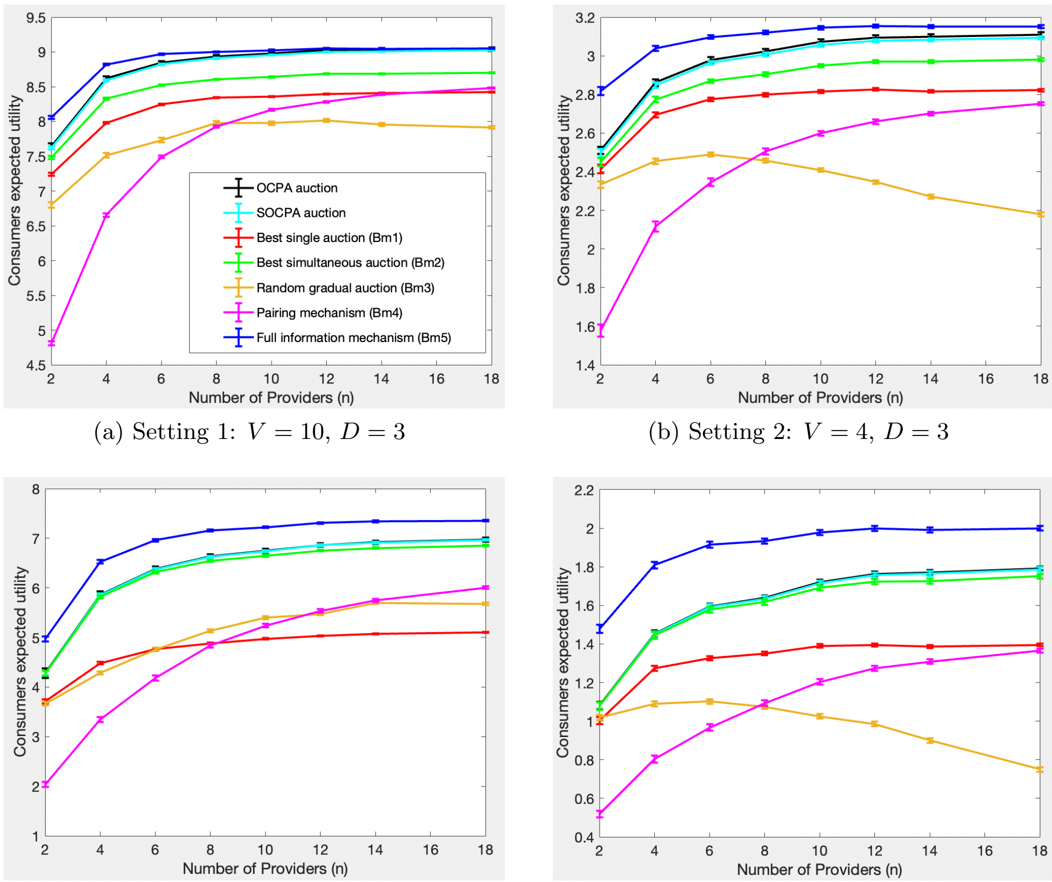
OCPA vs. SOCPA: Fig. 5 shows the close-to-optimal performance of the low-complexity sub-optimal auction SOCPA. These simulations show that the consumer's expected utility is degraded by less than 1% when it uses the less computationally demanding auction SOCPA instead of the optimal auction OCPA. We further investigate the relation between these two auctions in Section 9.3 to make sure that replacing OCPA with SOCPA does not significantly compromise optimality.

OCPA vs. Best single auction (Bm1): Comparing the expected utility of OCPA with that of Bm1, where only a single provider is recruited, show that the redundant allocation has a huge impact on the consumer's expected utility over a wide range of environments. We can see that OCPA results in 6.5%, 8.1%, 36.5%, and 26.9% higher utilities compared to those obtained by Bm1 in Settings 1-4, respectively. The percentage of improvement can go up to 377% when the value of the task and the deadline approach 40 and 0.1, respectively. Experimental evidence for this claim is presented in Appendix L.1.

OCPA vs. Best simultaneous auction (Bm2): We can see from Fig. 5 that the advantage of gradual recruitment shows itself more apparently when the task is not urgent (i.e., Figs. 5a-5b).[15] When the task is urgent (see Figs. 5c-5d), the OCPA auction is at most 2% better than Bm2. However, when the task has a normal deadline, this improvement increases to 5%.

OCPA vs. Random gradual auction (Bm3): This shows the importance of "intelligent" outsourcing planning. Fig. 5 shows that a non-intelligent use of the contingent plans

14. Tukey's test is essentially a t-test, except that it corrects for family-wise error rate.
15. This result is consistent with our discussion in Section 9.1.



(a) Setting 1: $V = 10$, $D = 3$

(b) Setting 2: $V = 4$, $D = 3$

(c) Setting 3: $V = 10$, $D = 1$

(d) Setting 4: $V = 4$, $D = 1$

Figure 5: Comparison of the performance of the OCPA and SOCPA auctions with benchmarks introduced in Section 9.2 in terms of expected utility

can even lead to a performance worse than mechanisms without redundancy and contingent decision making (e.g., Bm1).

OCPA vs. Pairing mechanism (Bm4): We can see from Fig. 5 that the OCPA auction significantly outperforms the pairing mechanism proposed in (Stein et al., 2011), in terms of the consumer's expected utility. In fact, the expected utilities obtained by OCPA are up to 59.5%, 61.8%, 122.4%, and 111% higher than those obtained by Bm4, in Settings 1-4, respectively. This improvement comes from three aspects:

1. Ignoring half of the providers in the pairing mechanism: As discussed previously, the pairing mechanism pairs the service providers randomly and disposes of the providers with the maximum cost at each pair. Some of the providers that are ignored by the mechanism could be the ones that are optimal to be recruited. The effect of this non-optimal decision making is more noticeable when the number of providers is low (e.g. $n \leq 10$), as in this case, the variety of services is low and hence there might not be a good alternative to the ignored providers.

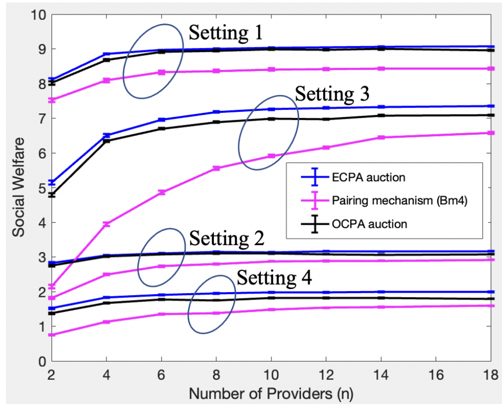


Figure 6: Comparison of the performance of the our proposed auctions with the pairing mechanism (Bm4) in terms of social welfare

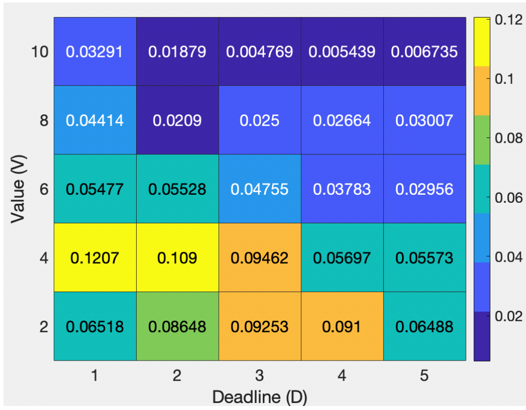
2. Different payment functions: OCPA employs the weighted threshold payment scheme which guarantees BIC and IR at the minimum cost. However, the pairing mechanism uses a heuristic payment function, which based on the results provided in Fig. 5, pays providers more than what is needed to incentivize them to tell the truth.

3. Different objectives: The pairing mechanism has the goal of maximizing social welfare, not the consumer's expected utility. Therefore, we shouldn't expect it to provide a high expected utility to the consumer. However, the pairing mechanism is the main competitor for our social welfare maximizing auction ECPA. So, our last evaluation in this section is to compare the social welfare levels under the ECPA auction and the pairing mechanism.

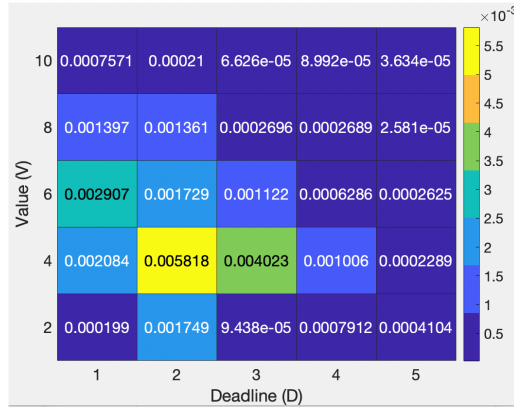
In Fig. 6, we plot the social welfares obtained by our proposed auctions (ECPA and OCPA) and the pairing mechanism in the four simulation settings described previously. We can see that in all settings, our proposed social welfare maximizing auction ECPA provides a significantly higher social welfare than the pairing mechanism (Bm4). The improvement is up to $9.5\%, 55.9\%, 137.2\%$, and 103%, in Settings 1-4, respectively. Surprisingly, we can also see that our utility maximizing auction OCPA results in a relatively high social welfare. Results in Fig. 6 show that the maximum difference of social welfare between OCPA and the social welfare maximizing auction ECPA is $2\%, 2.8\%, 7\%$, and 10.3%, in Settings 1-4, respectively. Also note that the performance of our low-complexity sub-efficient auction SECPA is not depicted in Fig. 6, as it is visually indistinguishable from that of ECPA. In fact, the maximum difference of expected social welfare between SECPA and ECPA in different settings and at different population sizes is below 1%.

9.3 Heuristics to Reduce the Time Complexity

In Section 9.2, we showed that the sub-optimal auction SOCPA which determines the outsourcing plans by running Algorithm 2 (instead of Algorithm 1 in the optimal auction OCPA) provides a close-to-optimal expected utility to consumer in Settings 1-4. The



(a) Maximum percentage of inefficiency caused by using Algorithm 2 instead of Algorithm 1



(b) Average percentage of inefficiency caused by using Algorithm 2 instead of Algorithm 1

Figure 7: Accuracy of the sub-optimal auction SOCPA introduced in Section 7.3

purpose of this section is to investigate the performance of SOCPA in a wider range of environments.

To this end, for each fixed $V \in \{2, 4, \dots, 10\}$ and $D \in \{1, 2, \dots, 5\}$, we conduct several simulations with different number of providers n and different cost vectors $\boldsymbol{c}$ and compare both the runtime and the utility provided by Algorithms 1 and 2. In terms of time complexity, experiments show that Algorithm 2 reduces the average runtime by $64\% - 99.8\%$, and thus enables settings with large numbers of providers (e.g. 100 providers runs in 4.4 seconds).

We define the "inefficiency" of Algorithm 2 for each cost vector $\boldsymbol{c}$ and population size n as the percentage decrease in the consumer's expected utility from the optimal expected utility of the outsourcing plan derived by Algorithm 1. These expected utilities include expectations with respect to the stochastic execution times. The results show that for each V and D, the inefficiency depends heavily on the cost vector $\boldsymbol{c}$, and slightly on the population size n. In Figs. 7a and 7b, we report the results of our worst-case and average-case analysis, respectively. In particular, for each V and D, we report the maximum inefficiency and the average inefficiency over all simulated cost vectors and population sizes. We can see from Fig. 7a that the maximum inefficiency could be as high as 12.07% when $V = 4$ and $D = 1$. However, the average inefficiency is not larger than 0.0058 (see Fig. 7b). This shows that the worst-case instances are very rare and do not appear in practice very frequently.

In the setting of this article, where the providers' costs are unknown to the consumer, the consumer compares different mechanisms based on their average expected utilities, where the average is taken over all possible cost vectors. Our simulations show that the average inefficiency has a low sensitivity to the population size. For example, for $V = 4$ and $D = 2$, the average inefficiency varies between 0.0047 and 0.0061 when n varies between 2 to 20. Therefore, for each population size n, the average inefficiency over all possible cost vectors is very close to the numbers reported in Fig. 7b. This shows that the SOCPA auction is a very good choice for a consumer with limited computational power that aims to maximize its expected utility.

9.4 Robustness Analysis

We made the following assumptions in all our previous simulations (except Fig. 3):

A1 The consumer has precise knowledge about the providers' duration functions;

A2 The providers' costs and service rates are correlated, i.e., providing a faster service incurs a higher cost;

A3 The providers' population is homogeneous, i.e., all providers' costs and service rates are chosen from the same distribution;

A4 The providers' duration distributions are exponential.

To check the robustness of the results in a wide range of circumstances, in this section, we study how the results change when each of these assumptions is relaxed. Since OCPA and SOCPA have very close performance, to save space, we will only report the results of robustness analysis for OCPA. The details of the simulations are in Appendix L.2 and only the main results are provided here.

Robustness to incorrect information (violation of Assumption A1): Our analysis shows that although the information inaccuracy worsens the performance of all outsourcing mechanisms (except for benchmark Bm3 where the available information is not used for decision making), its negative effect on OCPA is much lower than the other available benchmarks. The main impact of the information inaccuracy is not on the expected value, but instead on the variance of the utility obtained, and this is true for all the auctions. This is due to the nature of the problem. In a procurement auction, the decisions are often made based on the relative order and not absolute values of providers' service rates. The information inaccuracy, however, often changes the providers' absolute service rates but less frequently their relative order. This feature allows the auctions to bypass the inaccuracy and still choose a good outsourcing strategy even if the information is inaccurate.

Independent costs and service rates (violation of Assumption A2): Our analysis shows that Assumption A2 has no substantial effect on the results and hence all the trends discussed in Sections 9.2 and 9.3 still exist when this assumption is lifted. The only notable impacts of relaxing Assumption A2 are as follows: (1) For each fixed outsourcing mechanism, the expected utility that the consumer can achieve if the service rates and service costs are independent is higher than what it obtains when a positive correlation exists among them; (2) Unlike the correlated setting, where the consumer's expected utility approaches a stationary level when the population size n increases, in environments with independent service rates and service costs, the consumer's utility is strictly increasing in n. The reasons of such behaviors are discussed in Appendix L.2.2.

Heterogeneous population (violation of Assumption A3): In Appendix L.2.3, we investigate how different outsourcing mechanisms behave when providers are of different types with different duration and cost distributions. Our results suggests that OCPA offers an improvement over the benchmarks in the heterogeneous setting. Moreover, OCPA as well as Bm2 and Bm4 are capable of well adapting their outsourcing strategies to the population distribution.

Non-exponential duration distributions (violation of Assumption A4): In Appendix L.2.4, we explored different duration distributions and observed that the results are very similar to those reported in previous sections. The only meaningful difference is the percentage of improvement of OCPA over Bm2. Our studies reveal that OCPA outperforms the best simultaneous auction more significantly (up to 59%) when delivery times have multi-modal (mixture) distributions (Razali & Al-Wakeel, 2013). A multi-modal distribution is mainly used when the provider has two or more operation modes with different average delivery times. In such distributions, when the time passes the first mean and the task is not delivered yet, the consumer may believe that the provider is not in its best mode and hence update its estimation of the provider's delivery time accordingly. This phenomenon reveals some valuable information to the consumer over time and often gives it extra incentives to hire providers gradually.

10. Conclusions and Future Work

We have designed four contingent planning-based outsourcing auctions for situations where the service costs are privately and asymmetrically distributed among self-interested providers. These mechanisms allow the consumer to base its decisions on the important information it gains over time through direct experience. Our proposed auctions OCPA and ECPA achieve the highest possible consumer's utility and the highest possible social welfare, respectively. Low-complexity versions of these auctions, called SOCPA and SECPA, enable the consumer to achieve the same goals with less than 1% performance loss in a much lower running time.

All our developed auctions motivate self-interested providers to (i) voluntarily participate (i.e., interim individual rationality), (ii) report their costs truthfully irrespective of what others do (i.e., dominant strategy incentive compatibility), and (iii) perform the task upon recruitment with no possibility of experiencing regret in the future (i.e., ex-post individual rationality). Moreover, we showed empirically that the consumer's utility of OCPA significantly surpasses those of the best available outsourcing technologies by up to 59%. The results also reveal that ECPA outperforms the current state-of-the-art in terms of social welfare by up to 137%.

We plan to extend our work in several ways in the future. First, we would like to provide a theoretical upper bound for the average-case computational complexity of the heuristic-based auctions provided in this paper. Although this is a challenging task, it is worth attempting as the analysis may suggest further improvements to the auctions. The techniques provided in (Zhang, 2021) might be a good starting point towards this goal.

Second, it would be interesting to extend our binary completion model, with success values at V and failure values at 0, to settings where the partial completion is valuable. This extension is particularly important for settings where an any-time algorithm, which returns a valid solution even if it is interrupted before completion, is available for task execution. This extension is reasonably straightforward, since a similar approach to what we present in this paper can be applied to an updated utility function.

Third, we will deal with dynamic settings where some structural parameters, such as the set of providers, the service costs, and/or the task's value, can change over time. This extension is important as it enables the mechanism to adapt its outsourcing strategy to dynamically changing real-world environments. The dynamic mechanism design techniques provided in (Gallien, 2006) and (Pavan et al., 2014) may shed some light on how this goal can be achieved.

Finally, we intend to investigate settings where service durations (and/or costs) of different providers are correlated. In such domains, the execution uncertainty is not associated with the providers' workloads, but associated with the task difficulty. To address such settings, we will extend our branch-and-bound algorithm and also consider more general utility functions that capture such correlations.

11. Acknowledgment

The authors wish to thank the anonymous reviewers for their comments which have significantly improved the presentation of the results appearing in this paper.

Appendix A. Proof of Lemma 1

Equation (14) shows the expected utility of SP i in a contingent planning-based auction $M = (A(.), \boldsymbol{\pi}(.))$. In the simultaneous equivalent of this auction, i.e., auction $\hat{M} = (\hat{A}(.), \boldsymbol{\pi}(.))$, when provider i submits bid b_i and other providers submit $\boldsymbol{c}_{-i}$, provider i will be recruited at time 0 with probability $P_i(A(b_i, \boldsymbol{c}_{-i}))$. If this happens, provider i incurs cost c_i to perform the task and receives payment $\pi_i(b_i, \boldsymbol{c}_{-i})$ in compensation. Therefore, the expected utility of provider i in auction $\hat{M}$ can be derived by taking an expectation over the opponents' submitted cost vectors $\boldsymbol{c}_{-i}$ as follows:

$$U_i(\hat{A}, \pi_i, c_i, b_i) = \int P_i(A(b_i, \boldsymbol{c}_{-i}))[-c_i + \pi_i(b_i, \boldsymbol{c}_{-i})] f_{-i}(\boldsymbol{c}_{-i}) d\boldsymbol{c}_{-i}. \tag{31}$$

We can see that (31) is equal to (14), which completes the proof.

Appendix B. Proof of Lemma 2

The BIC and interim IR constraints (17b)- (17c) depend only on the providers' expected utilities. We have shown in Lemma 1 that the providers' expected utilities in each auction M and its simultaneous equivalent $\hat{M}$ are the same. Therefore, each auction M satisfies BIC and interim IR constraints if and only if its simultaneous equivalent $\hat{M}$ does so.

It has been shown in (Myerson, 1981) that conditions (I1)-(I2) of Lemma 2 are necessary and sufficient to ensure the BIC and interim IR of a randomized simultaneous auction $\hat{M}$. Therefore, the same conditions can characterize the set of BIC and interim IR contingent planning-based auctions.

Appendix C. Proof of Lemmas 3 and 4

The proofs of these two lemmas follow directly from the proof of Lemma 3 in (Myerson, 1981).

Appendix D. Proof of the Statement of Remark 4

The example below shows that mapping ψ does not keep the consumer's expected utility invariant. This example is inspired from Example 1.

Example 2 *Suppose that a high-valued task can be performed by two SPs. Provider* 1 *is cheap to hire, but its delivery time varies based on its workload. Under a low workload (prob.* α*), provider* 1 *delivers the task after* $D/2$ *units of time. However, it needs at least* $2D$ *time-units to deliver the task when working at a high workload (prob.* $1-\alpha$*). The second provider is expensive to hire, but it is totally reliable and always delivers the task by time* $D/2$*.*

Consider a contingent planning-based auction $M = (A(.), \boldsymbol{\pi}^A(.))$*, where*

$$A((c_1, c_2)) = \begin{cases} ((1,0),(2,D/2)), & \text{if} \quad c_1/c_2 < \alpha, \\ ((2,0)), & \text{if} \quad c_1/c_2 \geq \alpha, \end{cases} \tag{32}$$

and the payments $\boldsymbol{\pi}^A(.)$ *are derived based on the weighted threshold payment scheme. This auction always guarantees task completion, i.e.,* $P_{succ}(A(\boldsymbol{c}), D) = 1$*. So, the consumer's expected utility from auction* M *is*

$$U(A, \boldsymbol{\pi}^A) = V - \int \sum_{i \in \mathcal{N}} P_i(A(\boldsymbol{c}))\pi_i^A(\boldsymbol{c}) f(\boldsymbol{c}) d\boldsymbol{c}. \tag{33}$$

Now, consider $\hat{M} = (\hat{A}(.), \boldsymbol{\pi}^A(.))$ *as the simultaneous equivalent of auction* M*. This auction uses the same outsourcing plan when* $c_1/c_2 \geq \alpha$*, but recruits providers* 1 *and* 2 *with probabilities* 1 *and* $P_2(((1,0),(2,D/2))) = 1-\alpha$*, when* $c_1/c_2 < \alpha$*. Auction* $\hat{M}$ *has the same expected invocation cost as* M*, but a lower success probability. That is,*

$$P_{succ}(\hat{A}(\boldsymbol{c}), D) = \begin{cases} \alpha^2 - \alpha + 1, & \text{if} \quad c_1/c_2 < \alpha, \\ 1, & \text{if} \quad c_1/c_2 \geq \alpha. \end{cases} \tag{34}$$

Therefore, when the service costs come from a uniform distribution on $[0,1]$*, we have*

$$U(\hat{A}, \boldsymbol{\pi}^A) = V(1 + \alpha^3/2 - \alpha^2/2) - \int \sum_{i \in \mathcal{N}} P_i(A(\boldsymbol{c}))\pi_i^A(\boldsymbol{c}) f(\boldsymbol{c}) d\boldsymbol{c}, \tag{35}$$

and hence,

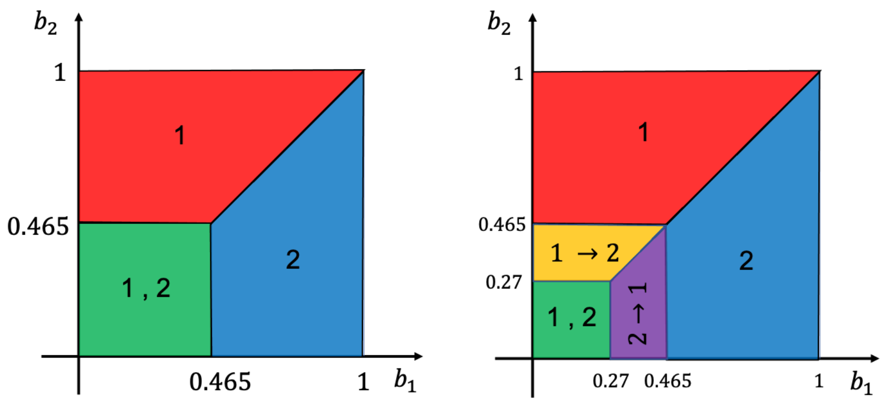
$$U(A, \boldsymbol{\pi}^A) - U(\hat{A}, \boldsymbol{\pi}^A) = V\frac{\alpha^2}{2}(1-\alpha). \tag{36}$$

This difference can be arbitrarily high when the task's value V is high.

Appendix E. Potential of Gradual Recruitment in Reducing the Invocation Cost

Example below illustrates the impact of gradual recruitment and the weighted threshold payment scheme on lowering the invocation cost.

Example 3 *Consider a task with value* $V = 4$ *and deadline* $D = 1$*. Suppose that there are two strategic service providers that can execute the task. The time needed by each of these providers to complete the task is exponentially distributed with mean* 1*, i.e.,* $G_1(t) =$



(a) Best simultaneous auction

(b) Best auction with gradual recruitment

Figure 8: Two allocation functions for the setting described in Example 3. To save space, we use short notations to show the recruitment strategy of each region. For example, 1 is a short notation for $\rho = ((1,0))$, 1, 2 is a short notation for $\rho = ((1,0),(2,0))$, and $1 \to 2$ represents any recruitment strategy $\rho = ((1,0),(2,t))$ with $t > 0$.

$G_2(t) = 1 - e^{-t}$. *Each provider i's cost c_i, which is only known to itself, is drawn from uniform distribution over* $[0,1]$.

In this setting, we denote the best simultaneous auction by $M_{no} = (A_{no}(.), \boldsymbol{\pi}_{no}(.))$. The allocation function $A_{no}(.)$ of this auction is shown in Fig. 8a. This allocation function instructs the consumer to recruit both providers simultaneously at time 0 *if submitted bids b_1 and b_2 are both below* 0.465*. Otherwise, the consumer should only recruit the provider with the minimum submitted bid. The minimum price that the consumer should pay to the providers to make auction M_{no} with allocation function $A_{no}(.)$ incentive compatible and individually rational is derived based on* (23)*, as follows:*

$$\boldsymbol{\pi}_{no}(b_1, b_2) = \begin{cases} (0.465, 0.465), & \text{if } b_1, b_2 \leq 0.465, \\ (b_2, 0), & \text{if } b_2 > 0.465, b_1 \leq b_2, \\ (0, b_1), & \text{if } b_1 > 0.465, b_2 < b_1. \end{cases} \tag{37}$$

In each pair returned by payment function $\boldsymbol{\pi}_{no}(.)$, the first and second elements are the prices paid to providers 1 and 2, respectively, upon recruitment.

Now, let's study what will happen if the consumer takes advantage of gradual recruitment. In this case, it is optimal for the consumer to run auction $M^* = (A^*(.), \boldsymbol{\pi}^{A^*}(.))$ with allocation function $A^*(.)$ shown in Fig. 8b[16]*. This allocation function divides the region with $b_1, b_2 \leq 0.465$ into three subregions below and provides a different outsourcing plan for each subregion:*

- ***Subregion 1*** *(*$b_1, b_2 \leq 0.27$*): Both providers get recruited simultaneously at time* 0.

16. The optimal allocation function can be computed by using Algorithm 1 provided in Section 6.

- ***Subregion 2*** ***(***$0.27 < b_2 \leq 0.465$***,*** $b_1 \leq b_2$***):*** *Provider 1 gets recruited at time* 0*. Provider 2 is recruited in the future (i.e., at a time* $\tau^*(b_2)$*), if the task is not completed by then.*

- ***Subregion 3*** ***(***$0.27 < b_1 \leq 0.465$***,*** $b_2 < b_1$***):*** *Provider 2 gets recruited at time* 0*. Provider 1 is recruited in the future (i.e., at a time* $\tau^*(b_1)$*), if the task is not completed by then.*

The optimal invocation time $\tau^*(.)$ *can be derived by techniques developed in Section 6.1. However, for the sake of simplicity and since in this example our goal is just to show how gradual recruitment can reduce the payments, we consider a suboptimal invocation time* $\tau(b) = 0.3$*, for all* $0.27 < b \leq 0.465$*. With this invocation time, the invocation probability of the provider who is second in the queue is* $1 - G_i(0.3) = 0.74$*.*

For this allocation function, we can derive the optimal payment function according to (23)*, as follows:*

$$\boldsymbol{\pi}^{A^*}(b_1, b_2) = \begin{cases} (0.414, 0.414), & \text{if } (b_1, b_2) \in \text{Subregion 1}, \\ (0.3441 + 0.26b_2, 0.465), & \text{if } (b_1, b_2) \in \text{Subregion 2}, \\ (0.465, 0.3441 + 0.26b_1), & \text{if } (b_1, b_2) \in \text{Subregion 3}, \\ (b_2, 0), & \text{if } b_2 > 0.465, b_1 \leq b_2, \\ (0, b_1), & \text{if } b_1 > 0.465, b_2 < b_1. \end{cases} \tag{38}$$

Comparing (38) *with* (37)*, we can see that in each Subregion 1-3, the gradual recruitment reduces the total cost of incentivizing providers to submit true bids. This fact is more attractive and apparent in Subregion 1, as in this region, the allocation function* A^* *is exactly the same as* A_{no}*. Therefore, the gradual recruitment has not changed the providers' invocation probabilities in this region. However, due to the changes it has made to the allocation functions of other regions, the payments of Subregion 1 are reduced as well.*

To see the reason, let's take a closer look at $\pi_1^{A^*}(b_1, b_2)$ *for Subregion 1. In this subregion, the invocation probability of provider 1 is* 1*, i.e.,* $P_1(A^*(b_1, b_2)) = 1$*. If provider 1 increases its bid* $\hat{b}_1$*, its invocation probability remains* 1 *up until* $\hat{b}_1 = 0.27$*, and then drops to* 0.74*, for* $0.27 < \hat{b}_1 \leq 0.465$*. Provider 1 will have no chance for being invoked if it submits a bid greater than* 0.465*. Based on these values, we can use* (23) *to compute payment* $\pi_1^{A^*}(b_1, b_2)$ *for Subregion 1 as follows:*

$$\pi_1^{A^*}(b_1, b_2) = b_1 + \int_{b_1}^{0.27} 1 \, d\hat{b}_1 + \int_{0.27}^{0.465} 0.74 \, d\hat{b}_1 = 0.414. \tag{39}$$

Comparing (39) *with* $\pi_{no,1}(b_1, b_2) = b_1 + \int_{b_1}^{0.465} 1 \, d\hat{b}_1 = 0.465$ *clarifies why payments of Subregion 1 are reduced when gradual recruitment is employed. In auction* M^**, if providers increase their bids, they will be at the risk of losing their first position in the recruitment queue and going to the second position. If this happens, the providers' invocation probabilities and hence their expected utilities decrease. Therefore, the providers have less incentive to submit higher bids and hence it is easier for the consumer to incentivize them to tell the truth.* □

Appendix F. Proof of Lemma 5

Consider optimization problem (25) when the monotonicity constraint is relaxed. This problem can be written as follows

$$\max_{A} \quad \int [VP_{succ}(A(\boldsymbol{c}), D) - \sum_{i \in \mathcal{N}} \phi_i(c_i) P_i(A(\boldsymbol{c}))] f(\boldsymbol{c}) d\boldsymbol{c}. \tag{40}$$

We can see that in this problem, each provider i's virtual cost $\phi_i(c_i)$ works to the disadvantage of its hiring. Therefore, if provider i's virtual cost $\phi_i(c_i)$ goes up when everything else is unchanged, provider i's invocation probability $P_i(A(\boldsymbol{c}))$ cannot be increased in the optimal solution. When the providers' cost distributions are regular, each provider i's virtual cost $\phi_i(c_i)$ is an increasing function of its cost c_i. In this case, each provider i's invocation probability is a decreasing function of its reported cost as well. That is,

$$b_i \geq c_i \Rightarrow P_i(A(b_i, \boldsymbol{c}_{-i})) \leq P_i(A(c_i, \boldsymbol{c}_{-i})), \quad \forall \boldsymbol{c}_{-i}. \tag{41}$$

Substituting (41) into (19) results in

$$b_i \geq c_i \Rightarrow Q_i(A, b_i) \leq Q_i(A, c_i), \tag{42}$$

which is equivalent to the monotonicity constraint (25b). This shows that the solution of problem (40) is monotone and hence, completes the proof of Lemma 5.

Appendix G. Proof of Proposition 1

The BIC property of the OCPA auction can be easily derived from Lemmas 2, 3, and 5. However, to prove the stronger notion of IC, i.e., DSIC, we need the following lemma, borrowed from (Manelli & Vincent, 2010) and adopted to our problem.

Lemma 6 *A procurement auction M with allocation function $A(.)$ and payment function $\boldsymbol{\pi}^A(.)$ derived in (23) is dominant-strategy incentive compatible if and only if for all $i \in \mathcal{N}$ and $\boldsymbol{c}_{-i}$, $P_i(A(b_i, \boldsymbol{c}_{-i}))$ is non-decreasing on b_i.*

The requirement of Lemma 6 is stronger than the monotonicity condition (25b), as it is not on each provider i's "expected" invocation probability $Q_i(A, b_i)$, where the expectation is taken over all others' cost vectors, but rather on provider i's invocation probability for each cost vector $\boldsymbol{c}_{-i}$ that might be reported by others.

However, we have already proven this stronger notion of monotonicity in the proof of Lemma 5. In Appendix F, we proved that the solution of optimization problem (40) satisfies both (41) and (42), which are equivalent to the stronger and weaker notions of monotonicity mentioned above, respectively. Using this, we can conclude that the OCPA auction whose allocation function is a solution to (40) satisfies the monotonicity requirement of Lemma 6 and hence is dominant strategy incentive compatible.

Appendix H. Proof of Proposition 2

Based on Proposition 1, all providers report their costs truthfully at the equilibrium. When a provider i reports its cost truthfully, it receives payment

$$\pi_i^{A^*}(\boldsymbol{c}) = c_i + \frac{1}{P_i(A^*(\boldsymbol{c}))} \int_{c_i}^{c_{max}} P_i(A^*(\hat{b}_i, \boldsymbol{c}_{-i})) d\hat{b}_i, \tag{43}$$

upon recruitment, which is clearly greater than its own cost c_i. This fact is irrespective of the other providers' bids and hence proves the ex-post individual rationality of OCPA.

Appendix I. Proof of Proposition 3

We designed OCPA's allocation function as the solution of optimization problem (25). The OCPA's payment function is also designed based on the weighted threshold payment scheme (23). Therefore, the utility-optimality of OCPA among all auctions that satisfy BIC and interim IR follows directly from Lemma 4.

Appendix J. Proof of Proposition 4

Based on Proposition 3, OCPA provides the consumer with the maximum expected utility that any BIC and interim IR auction could provide. Consider the auction $\tilde{M} = (\tilde{A}(.), \tilde{\boldsymbol{\pi}}(.))$ with

$$\tilde{A}(\boldsymbol{c}) = \emptyset, \quad \tilde{\pi}_i(\boldsymbol{c}) = 0, \quad \forall \boldsymbol{c} \in \mathcal{C}^n. \tag{44}$$

This auction neither recruits any provider nor makes any payment. This auction is both BIC and interim individually rational as the providers' expected utilities are 0 regardless of the cost they declare. Therefore, OCPA should outperform $\tilde{M}$ in terms of the consumer's expected utility. That is,

$$U(A^*, \boldsymbol{\pi}^{A^*}) \geq U(\tilde{A}, \tilde{M}) = 0. \tag{45}$$

Appendix K. Proof of Proposition 5

Monotonicity: To prove strong notions of monotonicity stated in Lemma 6, we show that a provider i's invocation probability $P_i(.)$ increases if it declares a lower cost c_i. This increase occurs for two reasons:

1. Increasing the chance of i being among the candidate providers o_{apx}: In Algorithm 2, each ordering o is valued according to the expected utility $Lower(o)$ it can provide to the consumer. For regular distributions, this expected utility is decreasing in terms of c_i for all $i \in o$ (see (27)). Therefore, declaring a lower cost by provider i increases the values of all orderings that include i, while keeps the value of all other orderings fixed. This provides a higher chance for orderings containing i to beat other orderings and get selected as the final ordering o_{apx}, and ultimately increases the invocation probability $P_i(.)$.

2. Advancing provider i's invocation: As discussed in Section 6.1, for each ordering $o = (1, \ldots, m)$, the providers' optimal invocation times $\tau_1^*, \ldots, \tau_m^*$ are derived by solving an optimization problem that maximizes $f(\tau)$ under some constraints. It can be shown, by some algebra, that each τ_i^* is an increasing function of $\phi_i(c_i)$. Therefore, when distributions are regular, reporting a lower cost c_i decreases $\phi_i(c_i)$ and hence τ_i^*, and ultimately increases $P_i(.)$.

Incentive compatibility and individual rationality: Given the monotonicity of allocation function A_{apx}, dominant strategy IC and ex-post and interim IR of SOCPA auction can be proved by following the same procedure as in the proofs of Propositions 1 and 2.

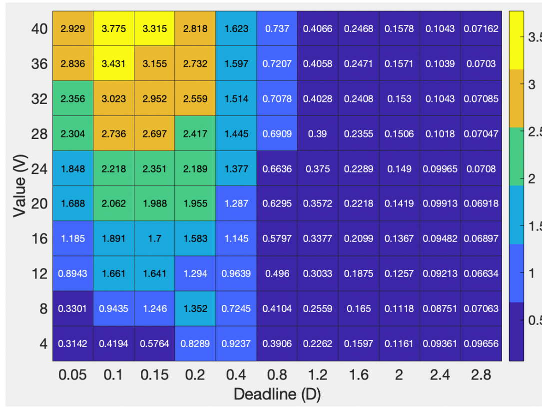


Figure 9: improvement-over-Bm1

Appendix L. Supplementary Numerical Experiments

L.1 Comparison between OCPA and Best Single Auction (Bm1) in terms of the Consumer's Expected Utility

For each pair of task's value V and deadline D, we run 1000 experiments for each number of providers $n \in \{2, 4, 6, \dots, 20\}$ with randomly generated providers. We then calculate the expected utilities provided by OCPA and Best Single Auction (Bm1) for each fixed tuple (V, D, n), and denote them by $U_{OCPA}(V, D, n)$ and $U_{Bm1}(V, D, n)$, respectively. We then derive the maximum percentage of improvement of OCPA over Bm1 for each pair of (V, D) by taking the maximum over the number of providers n. That is, the number reported in each cell (V, D) of Fig. 9 is calculated as follows:

$$I_1(V, D) = \max_n \frac{U_{OCPA}(V, D, n) - U_{Bm1}(V, D, n)}{U_{Bm1}(V, D, n)}. \tag{46}$$

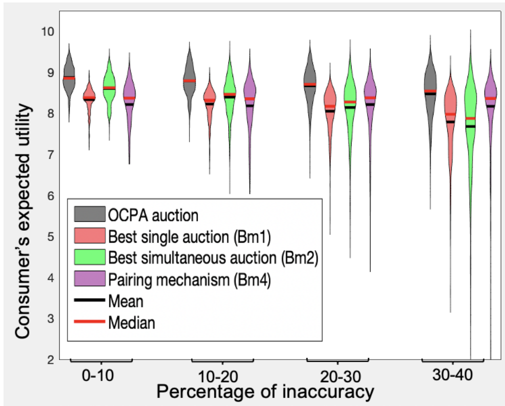
We can see from Fig. 9 that for each task's value V, the maximum percentage of improvement I_1 is first increasing and then decreasing in the deadline D. When the task's value goes up to 40, the maximum percentage of improvement goes up to 377%, and this occurs at deadline $D = 0.1$.

L.2 Robustness Analysis

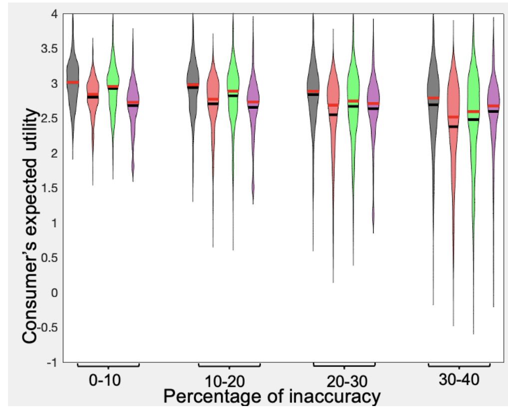
L.2.1 OCPA's Robustness to Incorrect Information

Our goal in this section is to assess the robustness of the OCPA auction regarding the consumer having incorrect information about providers' duration functions. To this end, for each percentage of inaccuracy $\delta \in (0, 40)$, we conduct several simulations where the consumer's belief about the providers' service rates $\boldsymbol{\lambda} = (\lambda_1, \dots, \lambda_n)$ has at most δ percent error (i.e., the consumer's belief is in a sphere with center $\boldsymbol{\lambda}$ and radius $0.01\delta|\boldsymbol{\lambda}|$). In each of these simulations, we compare the consumer's expected utility when it employs different available mechanisms.

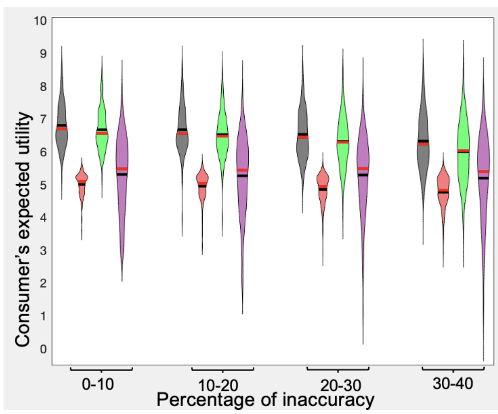
The results are shown in Figs. 10a-10d for Settings 1-4, respectively. In Figs. 10a-10d, we plot the distribution of consumer's utility for different ranges of inaccuracy. We can see



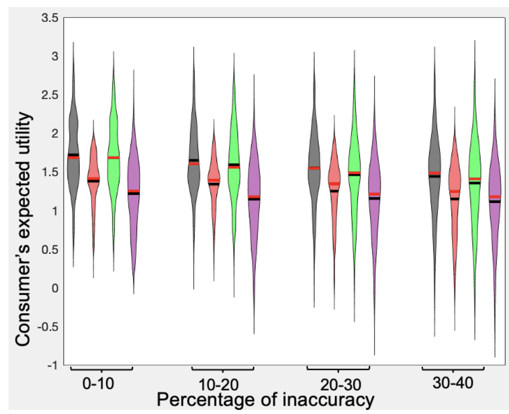
(a) Setting 1: $V = 10$, $D = 3$



(b) Setting 2: $V = 4$, $D = 3$



(c) Setting 3: $V = 10$, $D = 1$



(d) Setting 4: $V = 4$, $D = 1$

Figure 10: Robustness of OCPA as well as the other available benchmarks to information inaccuracy

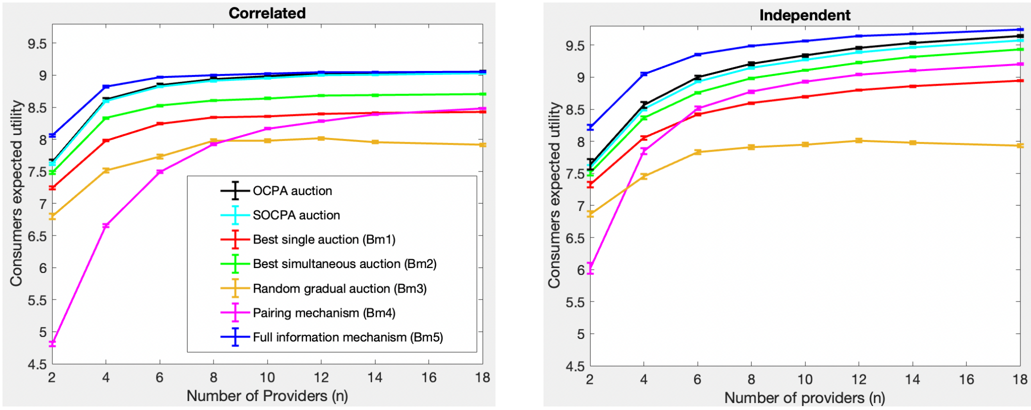
that the OCPA auction is better than the benchmarks in all the cases tested. More detailed discussions of the results are provided in Section 9.4.

Remark 6 *In this analysis, we do not include Bm3 as it does not utilize the consumer's information about the providers' duration functions and hence is not affected by information inaccuracy.*

L.2.2 Independent Costs and Service Rates

In this section, we study how the results change if Assumption A2 is lifted. To this end, we repeated all the simulations performed in Sections 9.2 and 9.3 in a new environment where providers' costs and service rates are chosen independently and uniformly from $[0, 1]$.[17] Such simulations show that Assumption A2 has no substantial effect on the results and hence all the trends discussed in Sections 9.2 and 9.3 still exist when this assumption is

17. In this environment, unlike the previous one, there may exist a provider i that provides a faster service than j at a lower cost. In such cases, provider i is said to dominate provider j.



(a) Correlated environment (with Assumption A2)

(b) Independent environment (without Assumption A2)

Figure 11: Impact of removing Assumption A2 on the relative performance of OCPA and the available benchmarks

lifted. However, to show the nature of the changes caused by removing Assumption A2, we compare the results of the same study as that in Fig. 5a for correlated and independent environments in Fig. 11.

We can see from Fig. 11 that the trends of the curves in both environments are quite similar. However, there are two differences in detail:

1. Each mechanism (except Bm3) provides a higher expected utility in the independent environment compared to the correlated one.

2. The curves (except that of Bm3) get flat after certain points in the correlated environment, however, the same cannot be observed in the independent setting.

The reason for the first difference is that in the correlated setting, the consumer has to spend more money if it wants to recruit a faster provider. However, in the independent environment, there is a chance that the consumer finds a fast provider at a low cost. This feature reduces the invocation cost and increases the utility of any intelligent mechanism in the independent environment. The second difference also comes from a similar reason. In the independent environment, the chance of there existing a high-speed provider with a low cost increases as the number of providers goes up. Therefore, all the available mechanisms (except Bm3) provide higher utilities when more service providers are available.

Note that the performance of random gradual auction (Bm3) is not improved in independent environments, as it does not make intelligent decisions and hence may invoke a slow provider at a high cost with the same probability that it may invoke a fast provider at a low cost. The effects of these two types of possible outcomes cancel each other out and keep the consumer's expected utility almost the same as what it was in a correlated environment.

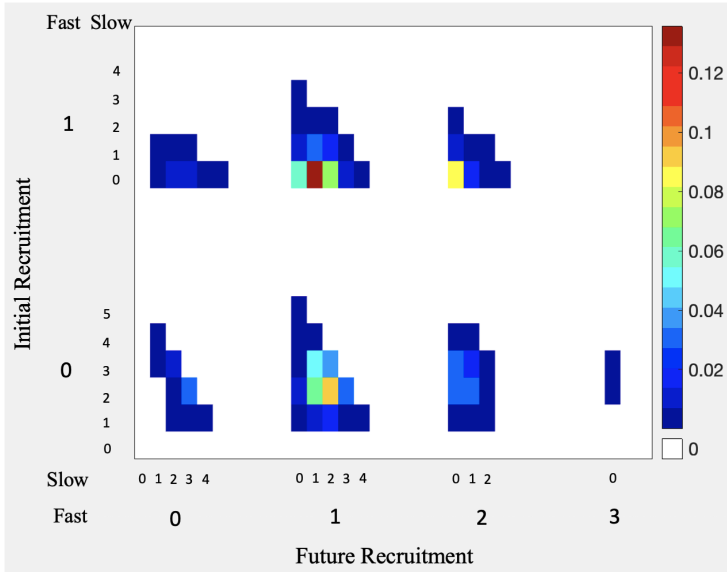


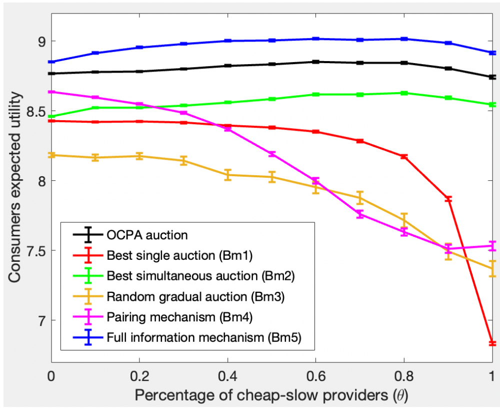
Figure 12: The distribution of different recruitment strategies in the OCPA auction for a heterogeneous environment with $n = 10$, $\theta = 0.5$, $V = 10$, and $D = 3$. The y-axis shows the numbers of fast and slow providers that are recruited at time 0. The x-axis shows how many fast and slow providers should be recruited gradually at $t > 0$.

L.2.3 Heterogeneous Population

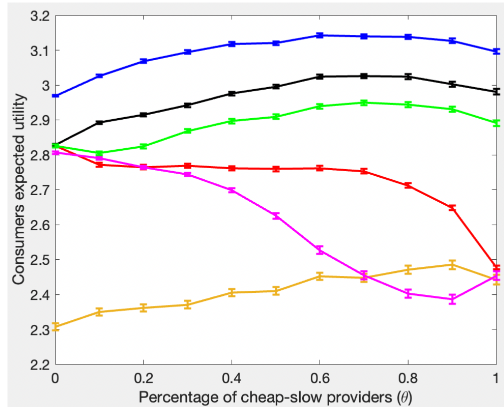
Consider an environment where the population is not homogeneous, but consists of two groups of providers with different properties. Providers of group 1 (G1) are cheap to hire but slow in delivering the task (i.e. $0 < c, \lambda < 0.5$), while providers of the second group (G2) are high-speed but expensive to hire (i.e. $0.5 < c, \lambda < 1$). We are interested to find the answers to the following three questions: 1) How does the optimal outsourcing plan behave in heterogeneous environments? 2) How much does the performance of OCPA depend on the heterogeneity of the environment? 3) How does OCPA compare to the available benchmarks in heterogeneous environments?

To answer the first question, we made several studies over environments with different percentages θ of cheap-slow providers, different task values V, and different deadlines D. We show one representative sample of the results in Fig. 12, where $n = 10$, $\theta = 0.5, V = 10$, and $D = 3$. This figure shows the distribution of different recruitment strategies in the OCPA auction. We see that in the most popular outsourcing plan, which we call fast-slow-fast (FSF), the consumer recruits one fast provider at time 0 and then recruits one slow and one fast provider gradually over time. In the second popular strategy, however, the consumer recruits two slow providers at time 0, recruits two more slow providers gradually over time, and then recruits a fast provider if none of the slow providers was successful in delivering the task. Popular recruitment strategies vary by θ, V, and D, however FSF is often among the most popular ones.

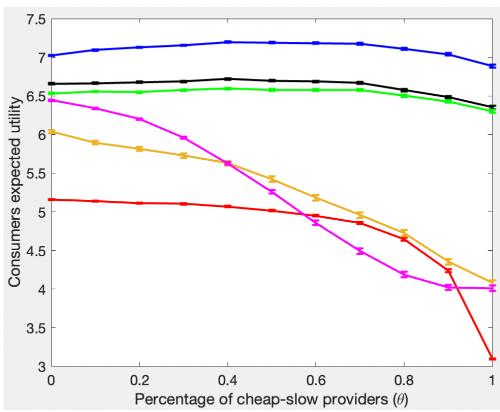
To answer the second and third questions, we compare the performance of OCPA with those of other benchmarks when the percentage θ of cheap-slow providers varies between 0



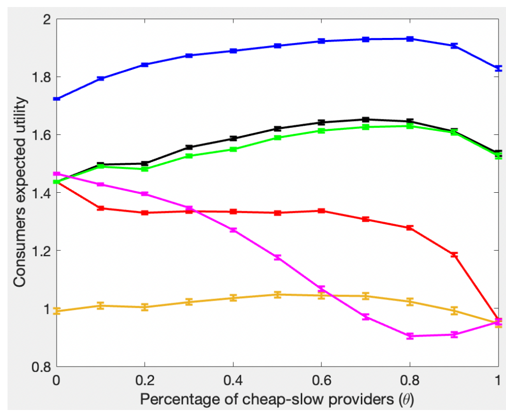
(a) Setting 1: $V = 10$, $D = 3$



(b) Setting 2: $V = 4$, $D = 3$



(c) Setting 3: $V = 10$, $D = 1$



(d) Setting 4: $V = 4$, $D = 1$

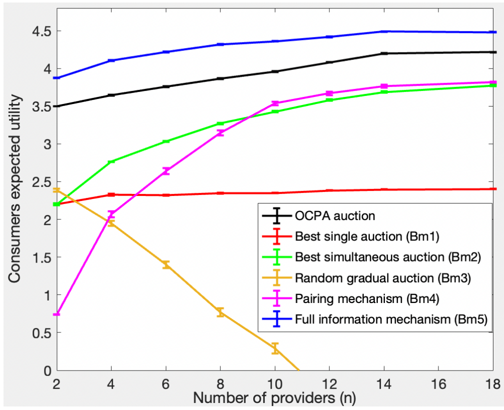
Figure 13: Comparison of the performance of the OCPA auction with benchmarks in terms of consumer's expected utility in heterogeneous setting.

to 1. The results of this comparison in Settings 1-4 are shown in Figs. 13a-13d, respectively. Fig. 13 suggests that OCPA offers an improvement over the benchmarks in the heterogeneous setting. Moreover, OCPA as well as Bm2 and Bm5 are capable of adapting such that they provide very stable utility in the whole region. In fact, the utility provided by OCPA has less than 12% variation, while the utilities provided by Bm1, Bm3, and Bm4, have over 40%, 48%, and 36%, variations, respectively.

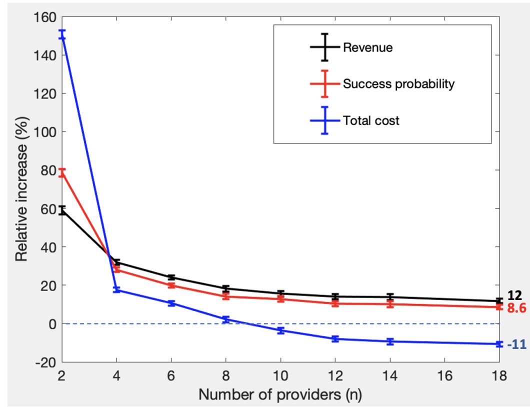
L.2.4 Non-Exponential Duration Distributions

In Fig. 14, we bring one simple example of the environments when the providers have non-exponential duration distributions.[18] In this example, we consider two groups of providers, where each provider of the first group has a deterministic delivery time $D/2$, and the delivery time of the second-group providers has a multi-modal distribution with two modes; in the first mode, the provider needs $D/2$ units of time to deliver the task, however, the delivery time of the second mode is $2D$. The second group of providers can be in any of the

18. This example is similar to Example 1 in Section 4.1.



(a) Consumer's expected utility versus the number of providers



(b) Percentage of increase in consumer's utility, success probability, and invocation cost for OCPA compared to Bm2

Figure 14: Performance comparison between OCPA and the benchmarks when the delivery times of a group of providers have multi-modal distributions and $V = 5$.

modes with probability 0.5. We show the performance of OCPA and the benchmarks in Fig. 14. In Fig. 14a, we present the consumer's expected utilities provided by different mechanisms. The analysis of variance (Anova) and Tukey's test revealed statistically significant differences in the utilities provided by OCPA compared to the available benchmarks.

Fig. 14b shows the percentage of increase in the consumer's expected utility, success probability, and invocation cost for OCPA compared to the best simultaneous auction Bm2. We can see that OCPA achieves 59% improvement in utility and 79% in success probability over Bm2 when the number of providers is low, i.e., $n = 2$. In this case, the consumer needs to spend 150% more money compared to Bm2 to obtain these advantages. When the number of providers goes up ($n \geq 15$), the OCPA auction provides 12% more utility compared to Bm2 with an 8.6% increase in success probability and 11% reduction in total costs.

References

Abolhassani, M., Ehsani, S., Esfandiari, H., HajiAghayi, M., Kleinberg, R., & Lucier, B. (2017). Beating 1-1/e for ordered prophets. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing (STOC)*, p. 61–71.

Akbarov, A., & Wu, S. (2012). Forecasting warranty claims considering dynamic over-dispersion. *International Journal of Production Economics*, *139*(2), 615–622.

Alaei, S., Fu, H., Haghpanah, N., Hartline, J. D., & Malekian, A. (2012). Bayesian optimal auctions via multi-to single-agent reduction. In *Proceedings of the 13th ACM Conference on Electronic Commerce (EC)*, p. 17.

Albore, A., Palacios, H., & Geffner, H. (2009). A translation-based approach to contingent planning. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 1623–1628.

Angeloni, L., & Martins-da Rocha, V. F. (2009). Large economies with differential information and without free disposal. *Economic Theory*, *38*(2), 263–286.

Babaioff, M., Lavi, R., & Pavlov, E. (2009). Single-value combinatorial auctions and algorithmic implementation in undominated strategies. *Journal of the ACM (JACM)*, *56*(1), 1–32.

Barbour, A. E., & Wojcik, A. S. (1989). A general constructive approach to fault-tolerant design using redundancy. *IEEE Transactions on Computers*, *38*(1), 15–29.

Bartolini, C., Preist, C., & Jennings, N. R. (2004). A software framework for automated negotiation. In *Proceedings of the 3rd International Workshop on Software Engineering for Large-Scale Multi-agent Systems (SELMAS)*, pp. 213–235.

Bergemann, D., & Valimaki, J. (2006). Efficient dynamic auctions. *Cowles Foundation Discussion Paper*.

Bertoli, P., Cimatti, A., Roveri, M., & Traverso, P. (2006). Strong planning under partial observability. *Artificial intelligence*, *170*(4-5), 337–384.

Bonet, B., & Geffner, H. (2000). Planning with incomplete information as heuristic search in belief space. In *Proceedings of the 5th International Conference on Artificial Intelligence Planning Systems (AIPS)*, pp. 52–61.

Börgers, T. (2015). *An Introduction to the Theory of Mechanism Design*. Oxford University Press.

Brubaker, E. R. (1980). On the auction mechanism and its incentive compatibility. *Journal of Political Economy*, *88*(3), 617–619.

Celis, L. E., Lewis, G., Mobius, M., & Nazerzadeh, H. (2014). Buy-it-now or take-a-chance: Price discrimination through randomized auctions. *Management Science*, *60*(12), 2927–2948.

Chli, M., & Winsper, M. (2015). Using the max-sum algorithm for supply chain emergence in dynamic multiunit environments. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, *45*(3), 422–435.

Clarke, E. H. (1971). Multipart pricing of public goods. *Public Choice*, *11*, 17–33.

Coit, D., & Smith, A. (1996). Reliability optimization of series-parallel systems using a genetic algorithm. *IEEE Transactions on Reliability*, *45*(2), 254–260.

Conitzer, V., & Sandholm, T. (2003). Applications of automated mechanism design. In *UAI-03 workshop on Bayesian Modeling Applications*.

Dauphin, Y. N., Pascanu, R., Gulcehre, C., Cho, K., Ganguli, S., & Bengio, Y. (2014). Identifying and attacking the saddle point problem in high-dimensional non-convex optimization. In *Proceedings of the 27th International Conference on Neural Information Processing Systems (NIPS)*, pp. 2933–2941.

Dixon, L. C. W., & Szego, G. (1978). *Towards global optimisation*, Vol. 2. North Holland.

Donna, J. D., & Espin-Sanchez, J.-A. (2018). Complements and substitutes in sequential auctions: the case of water auctions. *The RAND Journal of Economics*, *49*(1), 87–127.

Dughmi, S., Hartline, J., Kleinberg, R. D., & Niazadeh, R. (2021). Bernoulli factories and black-box reductions in mechanism design. *Journal of the ACM (JACM)*, *68*(2), 1–30.

Ergin, H., & Sarver, T. (2015). Hidden actions and preferences for timing of resolution of uncertainty. *Theoretical Economics*, *10*(2), 489–541.

Erradi, A., Maheshwari, P., & Tosic, V. (2006). Recovery policies for enhancing web services reliability. In *Proceedings of the 13th IEEE International Conference on Web Services (ICWS)*, pp. 189–196.

Esfandiari, H., Hajiaghayi, M., Liaghat, V., & Monemizadeh, M. (2017). Prophet secretary. *SIAM Journal on Discrete Mathematics*, *31*(3), 1685–1701.

Farhadi, F., & Jennings, N. R. (2021). A faithful mechanism for incremental multi-agent agreement problems with self-interested and privacy-preserving agents. *SN Computer Science*, *2*(4), 1–33.

Farhadi, F., Tavafoghi, H., Teneketzis, D., & Golestani, S. J. (2019). An efficient dynamic allocation mechanism for security in networks of interdependent strategic agents. *Dynamic Games and Applications*, *9*(4), 914–941.

Farina, G., & Gatti, N. (2017). Adopting the cascade model in ad auctions: Efficiency bounds and truthful algorithmic mechanisms. *Journal of Artificial Intelligence Research (JAIR)*, *59*, 265–310.

Fatima, S. S., Wooldridge, M., & Jennings, N. R. (2004). An agenda-based framework for multi-issue negotiation. *Artificial Intelligence*, *152*(1), 1–45.

Feldman, M., Fu, H., Gravin, N., & Lucier, B. (2020). Simultaneous auctions without complements are (almost) efficient. *Games and Economic Behavior*, *123*, 327–341.

Feng, Y., Hartline, J. D., & Li, Y. (2019). Optimal auctions vs. anonymous pricing: Beyond linear utility. In *Proceedings of the 20th ACM Conference on Economics and Computation (EC)*, pp. 885–886.

Forges, F. (2013). A folk theorem for bayesian games with commitment. *Games and Economic Behavior*, *78*, 64–71.

Freeman, P. (1983). The secretary problem and its extensions: A review. *International Statistical Review/Revue Internationale de Statistique*, 189–206.

Friese, T., Müller, J., & Freisleben, B. (2005). Self-healing execution of business processes based on a peer-to-peer service architecture. In *Proceedings of the 18th International Conference on Architecture of Computing Systems (ARCS)*, pp. 108–123.

Gahletia, K. (2021). An efficient supply chain in fast fashion through iot. In *Functional Textiles and Clothing*, pp. 205–212.

Gallien, J. (2006). Dynamic mechanism design for online commerce. *Operations Research*, *54*(2), 291–310.

Gelfand, I., & Fomin, S. (2012). *Calculus of Variations*. Dover Books on Mathematics. Dover Publications.

Gentillon, H., Stefańczyk, L., Strzelecki, M., & Respondek-Liberska, M. (2016). Parameter set for computer-assisted texture analysis of fetal brain. *BMC research notes*, *9*(1), 1–18.

Gerding, E., Stein, S., Larson, K., Rogers, A., & Jennings, N. R. (2010). Scalable mechanism design for the procurement of services with uncertain durations. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pp. 649–656.

Glatard, T., Montagnat, J., & Pennec, X. (2007). Optimizing jobs timeouts on clusters and production grids. In *Proceedings of the 7th IEEE International Symposium on Cluster Computing and the Grid (CCGrid)*, pp. 100–107.

Grefen, P. (2006). Towards dynamic interorganizational business process management. In *Proceedings of the 15th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE)*, pp. 13–20.

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

Guo, M., & Conitzer, V. (2010). Computationally feasible automated mechanism design: General approach and case studies. In *Proceedings of the 24th AAAI Conference on Artificial Intelligence*, pp. 1676–1679.

Guo, M., Shen, H., Todo, T., Sakurai, Y., & Yokoo, M. (2015). Social decision with minimal efficiency loss: An automated mechanism design approach.. In *Proceedings of the 14th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pp. 347–355.

Gupta, S., Chaudhari, S., Joshi, G., & Yağan, O. (2021). Multi-armed bandits with correlated arms. *IEEE Transactions on Information Theory*, *67*(10), 6711–6732.

Ha, C., & Kuo, W. (2006). Reliability redundancy allocation: An improved realization for nonconvex nonlinear programming problems. *European Journal of Operational Research*, *171*(1), 24–38.

Hartline, J. D., Kleinberg, R., & Malekian, A. (2011). Bayesian incentive compatibility via matchings. In *Proceedings of the 22nd annual ACM-SIAM symposium on Discrete Algorithms*, pp. 734–747.

Hoffmann, J., & Brafman, R. (2005). Contingent planning via heuristic forward search with implicit belief states. In *Proceedings of the 15th International Conference on Automated Planning and Scheduling (ICAPS)*, pp. 71–80.

Holmström, B. (1977). On incentives and control in organizations..

Hosseinalipour, S., & Dai, H. (2017). Options-based sequential auctions for dynamic cloud resource allocation. In *Proceedings of the 27th IEEE International Conference on Communications (ICC)*, pp. 1–6.

Huang, Z., & Kannan, S. (2012). The exponential mechanism for social welfare: Private, truthful, and nearly optimal. In *Proceedings of the 53rd IEEE Annual Symposium on Foundations of Computer Science (FOCS)*, pp. 140–149.

Huhns, M. N., Holderfield, V. T., & Gutierrez, R. L. Z. (2003). Robust software via agent-based redundancy. In *Proceedings of the 2nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pp. 1018–1019.

Iyengar, G., & Kumar, A. (2008). Optimal procurement mechanisms for divisible goods with capacitated suppliers. *Review of Economic Design*, *12*(2), 129–154.

Jaekel, F. (2019). Fundamentals of interfirm networks and networked horizontal lsp cooperation. In *Cloud Logistics*, pp. 53–144. Springer.

Jennings, N. R., Faratin, P., Lomuscio, A. R., Parsons, S., Sierra, C., & Wooldridge, M. (2001). Automated negotiation: prospects, methods and challenges. *International Journal of Group Decision and Negotiation*, *10*(2), 199–215.

Katehakis, M. N., & Veinott Jr, A. F. (1987). The multi-armed bandit problem: decomposition and computation. *Mathematics of Operations Research*, *12*(2), 262–268.

Keselman, H., & Rogan, J. C. (1977). The tukey multiple comparison test: 1953-1976.. *Psychological Bulletin*, *84*, 1050–1056.

Koide, T., & Sandoh, H. (2009). Economic analysis of an n-unit parallel redundant system based on a stackelberg game formulation. *Computers* & *Industrial Engineering*, *56*(1), 388–398.

Kong, Y. (2021). Sequential auctions with synergy and affiliation across auctions. *Journal of Political Economy*, *129*(1), 148–181.

Kotabe, M., Mol, M., Murray, J., & Parente, R. (2011). Outsourcing and its implications for market success: Negative curvilinearity, firm resources, and competition. *Journal of the Academy of Marketing Science*, *40*, 329–346.

Kraft, D., Fadaei, S., & Bichler, M. (2014). Fast convex decomposition for truthful social welfare approximation. In *Proceedings of the 10th International Conference on Web and Internet Economics (WINE)*.

Krengel, U., & Sucheston, L. (1977). Semiamarts and finite values. *Bulletin of the American Mathematical Society*, *83*, 745–747.

Lacity, M. C., Khan, S. A., & Willcocks, L. P. (2009). A review of the it outsourcing literature: Insights for practice. *The Journal of Strategic Information Systems*, *18*(3), 130–146.

Ledyard, J. (2007). Optimal combinatoric auctions with single-minded bidders. In *Proceedings of the 8th Annual Conference on Electronic Commerce (EC)*, pp. 237–242.

Lee, H., Park, S., Kim, J., & Kim, J. (2021). Auction-based deep learning computation offloading for truthful edge computing: A Myerson auction approach. In *Proceedings of the 35th International Conference on Information Networking (ICOIN)*, pp. 457–459.

Leme, R. P., Syrgkanis, V., & Tardos, E. (2012). Sequential auctions and externalities. In *Proceedings of the 23rd Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pp. 869–886.

Locatelli, M., & Schoen, F. (2013). *Global optimization: theory, algorithms, and applications*. SIAM.

Lucier, B. (2017). An economic view of prophet inequalities. *ACM SIGecom Exchanges*, *16*(1), 24–47.

Lukose, R. M., & Huberman, B. A. (2000). A methodology for managing risk in electronic transactions over the internet. *Netnomics*, *2*(1), 25–36.

Malakhov, A., & Vohra, R. V. (2009). An optimal auction for capacity constrained bidders: A network perspective. *Economic Theory*, *39*(1), 113–128.

Maleki, M., & Mashali-Firouzi, M. (2010). A numerical solution of problems in calculus of variation using direct method and nonclassical parameterization. *Journal of computational and applied mathematics*, *234*(5), 1364–1373.

Manelli, A. M., & Vincent, D. R. (2010). Bayesian and dominant-strategy implementation in the independent private-values model. *Econometrica*, *78*(6), 1905–1938.

Mansour, Y., Slivkins, A., & Syrgkanis, V. (2020). Bayesian incentive-compatible bandit exploration. *Operations Research*, *68*(4), 1132–1161.

Maskin, E., Dasgupta, P., & Hammond, P. (1979). The implementation of social choice rules: Some general results on incentive compatibility. *Review of Economic Studies*, *46*(2), 185–216.

Maskin, E., & Riley, J. (2000). Optimal multi-unit auctions. *International library of critical writings in economics*, *113*, 5–29.

Motahari Nezhad, H. R., Stephenson, B., & Singhal, S. (2009). Outsourcing business to cloud computing services: Opportunities and challenges. Tech. rep., HLP.

Myerson, R. (1981). Optimal auction design. *Mathematics of Operations Research*, *6*(1), 58–73.

Myerson, R. B. (1979). Incentive compatibility and the bargaining problem. *Econometrica*, *47*(1), 61–73.

Myerson, R. B. (1988). Mechanism Design. Discussion papers 796, Northwestern University, Center for Mathematical Studies in Economics and Management Science.

Narayan, V. V., Prebet, E., & Vetta, A. (2022). The declining price anomaly is not universal in multi-buyer sequential auctions (but almost is). *Theory of Computing Systems*, 1–35.

Nedelec, T., Baudet, J., Perchet, V., & El Karoui, N. (2021). Adversarial learning for revenue-maximizing auctions. In *Proceedings of the 20th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*.

Oinn, T., Greenwood, R., Addis, M., Alpdemir, N., Ferris, J., Glover, K., Goble, C., Goderis, A., Hull, D., Marvin, D., Li, P., Lord, P., Pocock, M., Senger, M., Stevens, R., Wipat, A., & Wroe, C. (2006). Taverna: Lessons in creating a workflow environment for the life sciences. *Concurrency and Computation: Practice and Experience*, *18*, 1067–1100.

Pandichi, A.-I., & Leon, F. (2011). Design and implementation of a multiagent system for auctioning and voting. *Bulletin of the Polytechnic Institute of Iasi, Automatic Control and Computer Science Section*, 87–101.

Parkes, D. C., & Singh, S. (2003). An MDP-based approach to online mechanism design. In *Proceedings of the 16th International Conference on Neural Information Processing Systems (NIPS)*, pp. 791–798.

Pavan, A., Segal, I., & Toikka, J. (2014). Dynamic mechanism design: A Myersonian approach. *Econometrica*, *82*(2), 601–653.

Peot, M. A., & Smith, D. E. (1992). Conditional nonlinear planning. In *Proceedings of the 1st International Conference on Artificial Intelligence Planning Systems (AIPS)*, pp. 189–197.

Pike-Burke, C., Agrawal, S., Szepesvari, C., & Grünewälder, S. (2017). Bandits with delayed, aggregated anonymous feedback. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*.

Pryor, L., & Collins, G. (1996). Planning for contingencies: A decision-based approach. *Journal of Artificial Intelligence Research (JAIR)*, *4*, 287–339.

Pycia, M., & Woodward, K. (2021). Auctions of Homogeneous Goods: A Case for Pay-as-Bid. Cepr discussion papers 15656, C.E.P.R. Discussion Papers.

Ramchurn, S. D., Mezzetti, C., Giovannucci, A., Rodriguez-Aguilar, J. A., Dash, R. K., & Jennings, N. R. (2009). Trust-based mechanisms for robust and efficient task allocation in the presence of execution uncertainty. *Journal of Artificial Intelligence Research (JAIR)*, *35*, 119–159.

Razali, A. M., & Al-Wakeel, A. A. (2013). Mixture weibull distributions for fitting failure times data. *Applied Mathematics and Computation*, *219*(24), 11358–11364.

Reddy, Y. M., & Ramachandran, M. (2008). Outsourcing of finance and accounting operations: The feasibility of the vertical in the Indian context. *ICFAI Journal of Services Marketing*, *6*(1).

Riley, J. G., & Samuelson, W. F. (1981). Optimal auctions. *The American Economic Review*, *71*(3), 381–392.

Rintanen, J. (2004). Complexity of planning with partial observability.. In *Proceedings of the 14th International Conference on Automated Planning and Scheduling (ICAPS)*, pp. 345–354.

Robbins, H., & Monro, S. (1951). A Stochastic Approximation Method. *The Annals of Mathematical Statistics*, *22*(3), 400 – 407.

Saadatmandi, A., & Dehghan, M. (2008). The numerical solution of problems in calculus of variation using chebyshev finite difference method. *Physics Letters A*, *372*(22), 4037–4040.

Salek, M., & Kempe, D. (2008). Auctions for share-averse bidders. In *Proceedings of the 4th International Workshop on Internet and Network Economics (WINE)*, pp. 609–620.

Shen, K., & Xie, M. (1990). On the increase of system reliability by parallel redundancy. *IEEE Transactions on Reliability*, *39*(5), 607–611.

Shmaryahu, D., Shani, G., & Hoffmann, J. (2019). Comparative criteria for partially observable contingent planning. *Autonomous Agents and Multi-Agent Systems*, *33*(5), 481–517.

Stanford-Smith, B. (2000). Vesi-the solution for interorganisational dynamic supply chains. *E-business: Key Issues, Applications and Technologies*, 475.

Stein, S., Gerding, E. H., Rogers, A., Larson, K., & Jennings, N. R. (2011). Algorithms and mechanisms for procuring services with uncertain durations using redundancy. *Artificial Intelligence*, *175*(14-15), 2021–2060.

Tai, Y.-M., Ho, C.-F., & Wu, W.-H. (2010). The performance impact of implementing web-based e-procurement systems. *International Journal of Production Research*, *48*(18), 5397–5414.

Tillman, F. A., Hwang, C.-L., & Kuo, W. (1977). Optimization techniques for system reliability with redundancy:a review. *IEEE Transactions on Reliability*, *26*(3), 148–155.

Vickrey, W. (1961). Counterspeculation, auctions, and competitive sealed tenders. *The Journal of Finance*, *16*(1), 8–37.

Vries, S., & Vohra, R. (2003). Combinatorial auctions: A survey. *INFORMS Journal on Computing*, *15*, 284–309.

Wang, Y., Niu, B., Guo, P., & Song, J.-S. (2021). Direct sourcing or agent sourcing? contract negotiation in procurement outsourcing. *Manufacturing & Service Operations Management*, *23*(2), 294–310.

Weitzman, M. L. (1979). Optimal search for the best alternative. *Econometrica*, *47*(3), 641–654.

Winsper, M., & Chli, M. (2013). Decentralized supply chain formation using max-sum loopy belief propagation. *Computational Intelligence*, *29*(2), 281–309.

Xu, Y., Zhu, K., & Li, S. (2020). Hierarchical combinatorial auction in computing resource allocation for mobile blockchain. *Wireless Communications and Mobile Computing*, *2020*, 1–14.

Yan, Q. (2011). Mechanism design via correlation gap. In *Proceedings of the 22nd Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pp. 710–719.

Yao, T., Jiang, B., Young, S. T., & Talluri, S. (2010). Outsourcing timing, contract selection, and negotiation. *International Journal of Production Research*, *48*(2), 305–326.

You, P.-S., & Chen, T.-C. (2005). An efficient heuristic for series–parallel redundant reliability problems. *Computers & operations research*, *32*(8), 2117–2127.

Zhan, R. L. (2008). Optimality and efficiency in auctions design: A survey. In *Pareto Optimality, Game Theory and Equilibria*, pp. 437–454.

Zhang, J. (2021). Average-case approximation ratio of scheduling without payments. *Algorithmica*, *83*(6), 1638–1652.

Zhang, L. (2018). *Simultaneous Auctions with Item Bidding*. Ph.D. thesis, McGill University.

Zhang, Y., Manisterski, E., Kraus, S., Subrahmanian, V., & Peleg, D. (2009). Computing the fault tolerance of multi-agent deployment. *Artificial Intelligence*, *173*(3-4), 437–465.

Zhang, Y., & Verwer, S. (2012). Mechanism for robust procurements. In *Proceedings of the 15th International Conference on Principles and Practice of Multi-Agent Systems (PRIMA)*, pp. 77–91.

Zheng, R., Dai, T., Sycara, K., & Chakraborty, N. (2016). Automated multilateral negotiation on multiple issues with private information. *INFORMS Journal on Computing*, *28*(4), 612–628.