Operations Research

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

Revenue Management of a Professional Services Firm with Quality Revelation

Kalyan Talluri, Angelos Tsoukalas

informs
https://pubsonline.informs.org/journal/opre

OPERATIONS RESEARCH
Articles in Advance, pp. 1–17
ISSN 0030-364X (print), ISSN 1526-5463 (online)

Crosscutting Areas

Revenue Management of a Professional Services Firm with Quality Revelation

Kalyan Talluri,[a,*] Angelos Tsoukalas[b]

[a] Imperial College Business School, Imperial College London, London SW7 2AZ, United Kingdom; [b] Rotterdam School of Management, Erasmus University, 3062 PA Rotterdam, Netherlands
*Corresponding author

Contact: kalyan.talluri@imperial.ac.uk, https://orcid.org/0000-0003-1608-6403 (KT); tsoukalas@rsm.nl, https://orcid.org/0000-0003-2679-3953 (AT)

Received: June 16, 2020
Revised: April 12, 2022
Accepted: June 21, 2022
Published Online in Articles in Advance: September 27, 2022

Area of Review: Market Analytics and Revenue Management

https://doi.org/10.1287/opre.2022.2351

Abstract. Professional service firms (PSFs) such as management consulting, law, accounting, investment banking, architecture, advertising, and home-repair companies provide services for complicated turnkey projects. A firm bids for a project and, if successful in the bid, assigns employees to work on the project. We formulate this as a revenue management problem under two assumptions: a quality-revelation setup, where the employees that would be assigned to the project are committed ex ante, as part of the bid, and a quality-reputation setup, where the bid's win probability depends on past performance, say, an average of the quality of past jobs. We first model a stylized Markov chain model of the problem amenable to analysis and show that up-front revelation of the assigned employees has subtle advantages. Subsequent to this analysis, we develop an operational stochastic dynamic programming framework under the revelation model to aid the firm in this bidding and assignment process. We show that the problem is computationally challenging and provide a series of bounds and solution methods to approximate the stochastic dynamic program. Based on our model and computational methods, we are able to address a number of interesting business questions for a PSF, such as the optimal utilization levels and the value of each employee type. Our methodology provides management with a tool kit for bidding on projects as well as to perform workforce analytics and to make staffing decisions.

Supplemental Material: The online appendix is available at https://doi.org/10.1287/opre.2022.2351.

Keywords: professional services • staffing • workforce analytics

1. Introduction

A professional service firm (PSF), such as a management consulting firm, has employees of varying skills, backgrounds, qualifications, and experience (Nachum 1996). Clients request a quote for their project, and the probability of winning a bid depends both on the price quoted (for the entire duration of the project) as well as the quality and suitability of the employees assigned to the job. If the firm wins the project, it is committed to executing it at that price and dedicating its resources to it.

This operational problem of PSFs, abstracting away specific industry details, can be summarized as a bidding-cum-matching problem where the probability of winning depends on the quality of the employees promised for the job as well as the price quoted in the bid. When the set of employees is specified as part of the bid, we call it the *quality-revelation* model. This typically happens in high-impact high-profile projects, or when the customer is a new one, or when the PSF is competing with larger rivals and needs to reassure the client on quality. This is the primary model we consider in this paper. We also consider an alternate model, the *quality-reputation* model, where the bid does not name specific employees and the winning probability depends on a quality reputation that is accumulated over past projects. This latter model is more appropriate for lower-end PSFs and for simpler, shorter-term projects with established relationships with the client.[1]

Although many papers in the literature have addressed various concomitant research issues for service firms in general (as we discuss in Section 1.1), relatively little work has been done specific to this operational and computational problem faced by PSFs. In this paper, we formulate it as a revenue management problem where the area partner (throughout, we assume the PSF is run by partners, as is typical in law or consulting) assesses each project, client, future work, and employees, and evaluates win probabilities and decides on a bid to maximize expected revenue. Although employee salaries are sunk in the short term, their opportunity cost—as the employee will be tied up with the project for a duration—has to be considered in the bid. We model the success of the bid as depending on the bid as well as the quality of the employees assigned to the project, which is a new dimension for revenue management. As we mentioned, quality comes into the equation either ex ante, revealed along with the bid, or as an accumulated reputation effect.

We first resort to a stylized Markov chain model to study the revelation versus reputation effects for a two-employee, one-project-type case. The analysis yields insights in favor of the transparency of the revelation model, on which we subsequently focus our attention.

Because the problem has a temporal dimension with uncertainty about new requests for quotes as well as bid wins, the appropriate framework is stochastic dynamic programming. Unfortunately, the dynamic program is impossible to solve exactly in practice, and we show that it is inapproximable to any constant factor (unless $P = NP$) when projects require greater than two employees, motivating a search for good tractable approximations. Because of the nonlinearity inherent in price optimization, we show that a deterministic math programming formulation does not approximate the dynamic program as well, so we look for more sophisticated value-function approximations. We propose a simple and fast greedy heuristic to calculate bids and use the approximations to bound its revenue performance numerically.

Our methodology is useful not just for the bid optimization problem, but also for performing workforce analytics and Monte Carlo simulations to determine the number of employees of each skill set to hire and when and the marginal value of each employee type, to guide the firm's overall resource management.

We summarize the main contributions of our work as follows:

1. On the modeling side, we formulate a stylized one-project, two-employee Markov chain model to gain managerial insights into the PSF's revenue management problem. In particular, we find that a counterintuitive strategy of bidding higher for lower-quality employees is optimal under the reputation model. We also show that in a hybrid reputation–revelation model, where a percentage of customers are perceptive and can infer the quality from the bid, the optimal revenue is lower than in the two pure models that generate identical revenue, thus illustrating the value of transparency. This motivates us to focus on the operational problem of the more transparent revelation model.

2. For the revelation model, we formulate the operational version of the problem as a dynamic program that is flexible enough to allow many modeling variations but, unfortunately, is intractable computationally. The main difficulty is the combinatorial effect of bundling employees to a project and the nonlinearity of the win-probability and revenue functions. We outline below our technical contributions in this regard.

a. We show that the problem for projects requiring more than one employee is inapproximable to any constant factor (assuming $P \neq NP$). On the positive side, somewhat surprisingly given the nonlinearity inherent in our problem, the methodology developed in Rusmevichientong et al. (2020) in the context of assortment optimization for reusable resources with fixed prices can be partly extended to our setup: we show that for one-employee projects, the revenue from a greedy algorithm has a guarantee of at least 50% of the maximum expected revenue.

b. Then, we consider the online single-stage problem and the many technical challenges that even this raises. We show how one can solve it via a two-step procedure: the first step involves solving a two-constraint knapsack problem, and the second step maximizing a univariate log-concave function.

c. Next, we obtain a relatively tight bound on the value function of the dynamic program by approximately solving an affine approximation linear program (LP) via constraint generation, in the spirit of Adelman (2007). Our innovation is in the solution of the separation problem which, unlike in Adelman (2007), is a difficult nonconvex problem. We solve this problem locally to obtain fast cuts and also develop an algorithm to obtain an upper bound to the global optimum. To that end, we use dual ideas as well as a trick to isolate the nonconvexity of the auxiliary problem to one variable.

d. Finally, we show how our revenue management formulation and its solution methods can be used to obtain valuable insights on staffing and utilization levels. There are very few analytical tools available for project-bidding and staffing of PSFs, so our framework provides a valuable tool kit for workforce management.

The remainder of this paper proceeds as follows. We give a survey of the literature in Section 1.1. In Section 2, we analyze a stylized Markov chain model. In Section 3, we formulate the operational problem, and in Section 4, we give analytical bounds and solution methods. Finally, in Section 5, we give numerical results on the performance of the algorithms as well as their application in workforce analytics. All proofs are relegated to the online appendix. In addition, in Section A of the online appendix, we describe our problem in the larger context of bidding and staffing for a PSF.

1.1. Literature Review

The literature on operations of PSFs is relatively sparse and spread out over many disciplines, including profession-specific areas such as accounting or law practice. Gilson and Mnookin (1985), for instance, provide an early survey specific to the law profession, which draws connections to economic principal–agent theories and remains highly relevant. Two further industry publications, by Cotterman (2016) and Mudrick (1990), describe the operations of law and accounting firms. In contrast, relatively little has been documented for management consulting firms, except for a few older articles (Maister 1982) and management books such as that by Maister (2012). Of a more academic and insightful nature are papers by Teece (2003) and a survey by Roth and Menor (2003) who both mention the paucity of research into the operations of a PSF.

Quality of an employee and reputation of a PSF are of course rather intangible concepts. Nachum (1996) surveys these aspects for large advertising PSFs. Two proxies they use for quality of employees are average salary and productivity of the employee.

On the subject of reputation there are many articles in the management science literature. Most empirical studies in this area are based on online customer feedback, either numerical ratings or textual reviews (for instance, Moreno and Terwiesch 2014). Unfortunately there is little of that sort of information for PSFs like large advertising firms or management consulting or law firms. However, both their clients and those in the industry are acutely aware of the reputations of the players, and past performance invariably contributes to such reputation.

Queueing models with *k*-identical servers and pricing control have a similar flavor as the problem faced by PSFs. This is usually stated as admission control using state-dependent pricing and no buffers. The well-known square-root staffing rule and the literature surrounding it provide great insight into how staffing grows with load, but the literature generalizing the rule usually does not consider all three elements in our model: matching, quality, and pricing. An exception is a recent article by Zhan and Ward (2019), which studies the problem of staffing where quality is a concern; the pricing part in that paper, however, is distinct from ours, as it is about the payments *to* the employees. A good recent survey of the area is provided by van Leeuwaarden et al. (2019).

As PSFs match employees to jobs dynamically, online bipartite matching is another relevant area. This has been extensively studied in the context of online advertisements with quality of the match being an important modeling element. In this application, web page requests are matched with an inventory of ads to show on the page. The resources are perishable, that is, not released back, so there is no concept of duration of the service. As in our model, decisions are online and irrevocable, and there is also a pricing element in the sense that the advertisers bid to place their ads on the page (real-time bidding model) and the highest bidder wins. Note that the logistic model that we use is a probabilistic version of the max-net-utility or envy-free pricing models used in this stream of literature. This is related to our problem as follows: Each ad corresponds to an employee, with the number of such employees corresponding to the number of ads of that type to be shown. Projects are users visiting the site, and each project has to be matched to at most one ad. The duration of a job is ∞ (so it exhausts the ad inventory). In our PSF model, each job requires a bundle of resources and can have arbitrary durations; hence, we are dealing with a significantly more difficult computational problem.

The problem is also related to network revenue management (Talluri and van Ryzin 2004) that is used to model the sale control process for hotels, railways, and airlines. The PSF revenue management problem is a generalization of the standard network revenue management problem—with a pricing and matching component and an underlying bipartite graph structure with qualities as weights on the edges. The difference from PSFs is that there is no matching based on qualities, and the prices are not personalized. Rusmevichientong et al. (2020) study the assortment optimization problem of such reusable resources. Because employees can be considered reusable resources, the computational techniques of Rusmevichientong et al. (2020) are relevant to us, and we discuss and extend them further in this paper. By adapting revenue management solution techniques, our paper brings new tools for use in workforce analytics for PSFs.

In the operations management area, there are a few important articles that study reputation for quality for service firms. Specific to PSFs, the papers by Boone et al. (2008) and Roels et al. (2010) are two such papers, but they concentrate on organizational and contracting aspects, respectively, in contrast to our revenue management focus. Adelman and Mersereau (2013) model a situation where customers remember past fill rates, which leads to goodwill that is updated by an exponentially smoothed average, and the firm faces the problem of allocating limited capacity—roughly corresponding to our quality-reputation regime, but at a personalized level. The customer demand is a function of the accumulated goodwill and an exogenous shock. They analyze the performance of a greedy policy and give an approximate dynamic programming approach. Using numerical studies, they evaluate the dynamics for a two-customer case to gain insights. In contrast, we primarily concentrate on the quality-revelation model.

The empirical paper by Bolton et al. (2006) is very relevant to the issues that we study in this paper. They study firms' contract renewal decisions as a function of the suppliers' service operation metrics over time. Based on a data set of support service contracts for high-tech systems, they find that a firm with very favorable experiences is more likely to renew that contract after controlling for average service levels. A similar study was performed by Sriram et al. (2015) based on a video-on-demand data set, focusing on the role of service variability in influencing retention.

In a very recent paper, DeCroix et al. (2021) studied quality variability and how it affects personalized dynamic pricing. Their concerns are concordant to ours, but in a different setting. We focus specifically on PSFs and the problem of bidding for projects as well as allocation of resources. Our quality-reputation modeling shares many similarities, however; firms' reputation for quality follows an exponential-smoothing process, and the purchase decision has a logit probability.

2. Markov Chain Model: Reputation vs. Revelation

When reputation is the driver behind the win probability, a PSF may be tempted to follow the dictum (mentioned to us by a PSF partner), "promise the best and assign what is available." This, however, has long-term consequences in terms of reputation and the possibility of winning future bids. To gain insight into the interactions of short-term revenues and long-term reputation, we develop a stylized model of the problem with one project type and two employees. The model leads to tractable near-closed-form solutions in steady state under both the revelation and reputation schemes, and allows us to compare them for insight into the main drivers and effects.

Assume we have two employees of qualities q_1 and q_2 ($q_1 > q_2$; referred to as high-quality and low-quality, respectively) and a unique project type that arrives with probability λ in each period and requires a single employee for a duration of exactly two periods; thus, an employee assigned at time t is not available in period $t+1$. Also assume zero employee cost (i.e., it is a sunk cost). We model the probability to win the project as

$$w(b,q) = \frac{1}{1+e^{\beta_0-\beta_q q+\beta_b b}}, \tag{1}$$

where b is the bid, and q is the supplied quality under the revelation scheme. The parameters β_q, β_b are assumed positive, so that the winning probability is increasing in quality and decreasing in bid amount. Equation (1) also yields the win probability under the reputation model, where q now represents the reputation of the firm. The reputation of the firm is taken to be the average over all qualities up to time t.

The system will transition between three states $(1,0)$, $(0,1)$, and $(1,1)$, where the elements of the tuple represent the first and second employee respectively, and the value one represents availability for assignment and zero otherwise. For example, $(1,0)$ represents a state where the higher-quality employee 1 is available and the lower quality employee 2 is on a project and not available. Note that there will always be an employee available (hence, (0, 0) is not a reachable state), as only one project arrives per period, each project lasts exactly two periods, and a maximum of one employee is assigned to a project.

Remark 1. Under both the reputation and revelation models, we should always greedily assign the high-quality employee, if available, over the low-quality employee. Indeed, there are two possibilities at state (1, 1). Either we assign the hiqh-quality employee or the low-quality employee (i.e., we do not have to consider the case of not assigning anyone, as we can always bid high enough to make an assignment profitable). If it is optimal to assign the high-quality employee, the state is identical to (1, 0), as (1, 1) and (1, 0) are indistinguishable in terms of both immediate revenue and transition to the next state. Similarly, if it is optimal to assign the low-quality employee, the state is identical to (0, 1).

But under both models, being in the state (1, 0) is preferable to being in state (0, 1); that is, we would prefer to have the high-quality employee available more often, and therefore the state (1, 1) is equivalent to the state (1, 0), whence assigning the high-quality employee is preferable.

In view of Remark 1, we collapse the states (1, 0) and (1, 1) into one and consider a two-state model (Figure 1) with the following collapsed states:

- *A*—A high-quality individual is available;
- *N*—A high-quality individual is not available (which implies a low-quality individual is available).

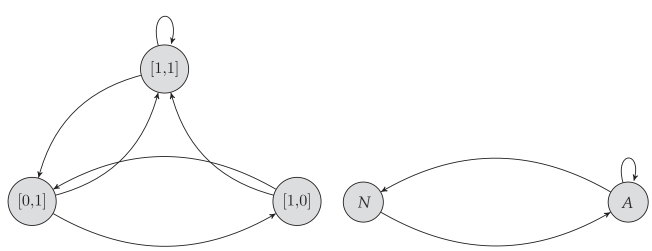
Then, a policy is defined by the bidding strategy (b_A, b_N) in the revelation model and $(b_A(\bar{q}), b_N(\bar{q}))$ in the reputation model. For a given quality, in both models, there is a one-to-one correspondence between bid amount and winning probability. Therefore, when convenient, we can also define the policies with respect to the winning probabilities (w_A, w_N) in the revelation model and $(w_A(\bar{q}), w_N(\bar{q}))$ in the reputation model.

In the reputation model, if τ_t projects have been won up to time t, then if a project is won and assigned to employee i at time t, we will have $\bar{q}_{t+1} = \bar{q}_t + \frac{1}{\tau_t+1}(q_i - \bar{q}_t)$, whereas if the project is not won, we will have $\bar{q}_{t+1} = \bar{q}_t$. Thus, in all cases there holds

$$|\bar{q}_{t+1} - \bar{q}_t| \le \frac{|q_1 - q_2|}{\tau_t + 1}.$$

Because of the long-term infinitesimal effect of a single period on reputation, we consider only stationary

Figure 1. (Left) The Markov Chain with Three States and (Right) Two States



policies, and thus at steady state, we can assume that reputation stabilizes to a time-invariant $\bar{q}$. Indeed, consider a constant winning rate w_N at state N. The state N is reached exactly as many times as a high-quality individual is assigned to a project and the project is won. For every one of these times, exactly λw_N times a project is won and a low-quality individual is assigned. Therefore, in the limit, the ratio of high-quality individuals to low-quality individuals is $\frac{1}{\lambda w_N}$, and the average reputation converges to $q_2 + \frac{1}{1+\lambda w_N}(q_1 - q_2)$.

We thus take as a given a constant $\bar{q}$ in the steady state and constant policies $(b_A(\bar{q}), b_N(\bar{q}))$ for the subsequent analysis. For any fixed such policy, the system can be analyzed as a two-state Markov chain.

For the rest of this section, unless explicitly stated, the discussion and equations refer to both the reputation and revelation models. At the steady state, let π_A and π_N be the probabilities of being in the two states, respectively (see Table 1).

The transition probabilities given the policy are

$$\begin{bmatrix} 1-\lambda w_A & \lambda w_A \\ 1 & 0 \end{bmatrix},$$

and the corresponding steady state probabilities are $\pi_A = \frac{1}{1+\lambda w_A}$ and $\pi_N = \frac{\lambda w_A}{1+\lambda w_A}$.

The percentage of periods where we win a project and assign the high-quality employee is $\lambda \pi_A w_A$, and the percentage of periods where we win a project and assign the low-quality employee is $\lambda \pi_N w_N$. Therefore, the average quality, also confirming an earlier statement, is

$$\bar{q} = \frac{\lambda \pi_A w_A q_1 + \lambda \pi_N w_N q_2}{\lambda \pi_A w_A + \lambda \pi_N w_N} = \frac{\lambda w_A q_1 + \lambda^2 w_A w_N q_2}{\lambda w_A + \lambda^2 w_A w_N}$$
$$= \frac{q_1 + \lambda w_N q_2}{1 + \lambda w_N} = q_2 + \frac{1}{1+\lambda w_N}(q_1 - q_2),$$

and the average revenue per period extracted by the policy is $\lambda \pi_A b_A w_A + \lambda \pi_N b_N w_N$.

When a project arrives, the average revenue extracted is $b_A w_A$ when at state A and $b_N w_N$ when at state N. In Proposition 1, we show that when operating optimally, the average revenue that can be extracted is the same under both models.

Table 1. Notation for the Markov-Chain Model

State	State probability	Resource	Bid	Win probability	Revenue
A	π_A	1	b_A	w_A	$b_A w_A$
N	π_N	2	b_N	w_N	$b_N w_N$

Proposition 1. *When operating optimally, the average revenue per period under the revelation model is equal to the average revenue per period under the reputation model.*

We next show that even though the same revenue can be generated under both models, this is possible in the reputation model only by adopting a counterintuitive policy that bids higher for the lower-quality employee. The revelation model behaves predictably in this respect, as we first show in Proposition 2.

Proposition 2. *For the optimal bidding policy (b_A^*, b_N^*) under the revelation model, there holds $b_A^* > b_N^*$.*

Proposition 3 is somehow counterintuitive. It says that in the reputation model, we should bid higher when supplying the low-quality employee. As we extract value only from our reputation, we can extract equal value from high- and low-quality employees in the current period. Reputation considerations imply that we benefit in the long term from assigning good employees when winning; therefore, we are inclined to ask for higher compensation for assigning a low-quality employee. As dictated by intuition and shown in Lemma 1, in spite of the higher bidding for the low-quality employee, we generate more revenue when the high-quality employee is available.

Lemma 1. *For the optimal bidding policy (b_A^*, b_N^*) under the reputation model with corresponding win probabilities (w_A^*, w_N^*), there holds $w_A^* b_A^* \geq w_N^* b_N^*$.*

Proposition 3. *For the optimal bidding policy (b_A^*, b_N^*) under the reputation model, there holds $b_N^* \geq b_A^*$.*

We have seen that the firm gets the same revenue in the reputation and revelation cases. Let us examine the situation from the customer's point of view. From symmetry, the total fees are equal in the two cases. But what is the utility for the customer with respect to the number of projects assigned for these fees and the distribution of high- and low-quality employees? It follows from the reasoning of the proof of Proposition 1 that the optimal reputation and revelation policies will have the same w_A and w_N, which also implies an identical quality distribution. What changes is the payment schedule, and we next discuss a couple of things that may make the reputation schedule unattractive.

2.1. Reputation vs. Revelation, and Signaling

One could argue that the bidding "inversion" under the reputation model reflects the absence of signaling effects in the model: if we were to implement the bidding strategy of the reputation model, customers would be likely to infer, at least to some degree, the quality from the bid. If we assume all customers are highly observant and infer the bid-to-quality relationship exactly, then we converge to the revelation model. The reputation model, as studied in this section, is the other extreme, where customers do not learn at all.

But what about an in-between zone, where some customers can deduce the quality of the employee from the bid while others cannot and act based on the firm's reputation? The core of the insight remains relevant: the reputation drop associated with winning using low-quality employees is a force pushing toward biding high when supplying such employees. In such a hybrid regime, we face a dilemma. Using "revelation bidding" leads to higher acceptance of low bids and lower acceptance of high bids by naïve customers, lowering our revenues. Using "reputation bidding" leads to exploitation by perceptive customers. In Proposition 4, we resolve this dilemma, showing that under a hybrid regime, we cannot achieve the same revenue as in either of the pure cases. Thus, there is value in the clarity of knowing all customers behave in the same way, something we can achieve by revealing quality ex ante.

Proposition 4. *Consider a hybrid regime where a fraction $0 < \alpha < 1$ of the customers can deduce the quality of the employee from the bid and $(1-\alpha)$ cannot and decide based on reputation, where the reputation $\bar{q}$ is the average quality of the wins of the naïve customers. The optimal revenue generated when operating under this hybrid regime is strictly less than the optimal revenue of the revelation regime.*

Remark 2. A secondary, somewhat unsatisfying property of reverse bidding is related to employee incentives. Although total revenue for the firm does not change under the two models, the division of billing is different. Even though in both models the hiqh-quality employee brings in more revenue, the difference is higher in the revelation model, as we demonstrate in Proposition 5. Thus, employees may be less incentivized for quality work under the reputation model.

Proposition 5. *In both the reputation and revelation models, the high-quality employee brings in more revenue than the low-quality employee. The difference, however, is higher under revelation.*

Motivated by the results of this section, we focus the rest of this paper on studying the PSF problem under the revelation model.

3. An Operational Model Under Quality Revelation

In this section, we consider the PSF revenue management problem with quality revelation under a general setup, allowing for an arbitrary number of employees, multiple project types of varying durations, and employee quality match metrics that may vary by project. We formulate the problem as a dynamic program that unfortunately is computationally difficult to solve: the state space of the dynamic program explodes even for small problems; so the goal is to obtain "good" policies via approximations (in Section 4).

Time is discrete and runs over a horizon from one to T. We assume a set P of project types and a set I of employees $i \in I$. Each project type $p \in P$ has a fixed requirement of k_p individuals for a duration of d_p. A project of type p arrives with probability λ_{pt} in period t, with at most one project arrival per period $\sum_{p\in P}\lambda_{pt} \le 1$. We assume that $\lambda_{pt} = 0$ for $t > T - d_p$. When it is clear from the context, we write *a project p* instead of *a project of type p*. We denote by $\boldsymbol{q}_p$ the vector of size $|I|$ of quality measures of assigning individuals to project p.

3.1. The State Space

The state $\boldsymbol{s}$ is a vector in $\mathbb{Z}_+^{|I|}$, with the ith element s_i representing the number of periods after which that individual i becomes free. So if $s_i = 0$, employee i is currently free and can be assigned immediately, whereas if $s_i > 0$, he or she is working on some other project and will become available in s_i periods.

We also use the indicator

$$\mathbb{1}_{\boldsymbol{s},i} = \begin{cases} 1 & \text{if } s_i = 0, \\ 0 & \text{otherwise} \end{cases}$$

to indicate that at state $\boldsymbol{s}$ individual i is available, or $\mathbb{1}_{\boldsymbol{s}}$ in vector notation.

3.2. The Controls

The decision variables are how much to bid and the assignment of individuals to the project. The feasible assignments $\mathcal{F}_{\boldsymbol{s}}^p$ to project p when at state $\boldsymbol{s}$ are the set of assignments that assign only among available resources, that is,

$$\mathcal{F}_{\boldsymbol{s}}^p = \{\boldsymbol{x} \in \{0,1\}^{|I|} \mid \boldsymbol{x} \le \mathbb{1}_{\boldsymbol{s}}, \quad \mathbf{1}^T\boldsymbol{x} = k_p\}. \tag{2}$$

The feasible assignments if all resources are available are $\mathcal{F}_{\mathbf{0}}^p$.

3.3. The State Dynamics

If a project is won and $\boldsymbol{x}$ is a feasible assignment of employees, then we have $\boldsymbol{s}^{t+1} = [\boldsymbol{s}^t - \mathbf{1} + d_p\boldsymbol{x}]^+$. If no project is won, then no employee is assigned, and we have $\boldsymbol{s}^{t+1} = [\boldsymbol{s}^t - \mathbf{1}]^+$.

Remark 3. Additional constraints can be imposed on $\mathcal{F}_{\boldsymbol{s}}^p$. For example, one can introduce employee types and enforce that at least k_p^l employees of type l are assigned to the project via imposing constraints $\sum_{i\in I^l} x_i \ge k_p^l$, where $I^l \subset I$ are the employees of type l. We do not consider such constraints, however, as they affect the difficulty of the combinatorial single-period subproblem that we examine in *Section* 4.1.3.

3.4. Win Probabilities

Given a feasible assignment $\boldsymbol{x} \in \mathcal{F}_{\boldsymbol{s}}^p$ and a bid b per person-day, we win the project with a probability that is a function of the bid and the average quality $\bar{\boldsymbol{q}}_p^T \boldsymbol{x}$ of the assignment where $\bar{\boldsymbol{q}}_p = \frac{q_p}{k_p}$. Specifically, we assume that the probability of winning project p is given by the logistic function

$$w_p(b, \bar{\boldsymbol{q}}_p^T \boldsymbol{x}) = \frac{1}{1 + e^{\beta_0^p + \beta_b^p b - \beta_q^p \bar{\boldsymbol{q}}_p^T \boldsymbol{x}}}, \tag{3}$$

with $\beta_q^p > 0$ and $\beta_b^p > 0$, so that the probability to win is increasing as a function of quality and decreasing as a function of the bid.

If the project is won, we incur a cost $\boldsymbol{c}_p^T \boldsymbol{x}$. If we assume the daily cost to be $\boldsymbol{c}$, that is, independent of p, then we can set $\boldsymbol{c}_p = d_p \boldsymbol{c}$. The costs $\boldsymbol{c}$ capture nonsunk costs only, if any. Employee salaries that do not depend on the workload are not part of the dynamic program; however, they play a role in optimal hiring decisions that we investigate in our numerical simulations of Section 5.

Let $V_t(\boldsymbol{s})$ be the value function representing optimal expected revenue from a project-bidding and resource-allocation policy. Noting that project p has $k_p d_p$ workdays, the dynamic program is

$$\begin{aligned} V_t(\boldsymbol{s}) = \sum_{p \in P} \lambda_{pt} \max_{\substack{b_p \in \mathbb{R}^+ \\ \boldsymbol{x}_p \in \mathcal{F}_s^p}} &[(k_p d_p b_p - \boldsymbol{c}_p^T \boldsymbol{x} \\ &+ V_{t+1}([\boldsymbol{s} - \boldsymbol{1} + d_p \boldsymbol{x}_p]^+)) w_p(b_p, \bar{\boldsymbol{q}}_p^T \boldsymbol{x}_p) \\ &+ (1 - w_p(b_p, \bar{\boldsymbol{q}}_p^T \boldsymbol{x}_p)) V_{t+1}([\boldsymbol{s} - \boldsymbol{1}]^+)] \\ &+ \left(1 - \sum_{p \in P} \lambda_{pt}\right) V_{t+1}([\boldsymbol{s} - \boldsymbol{1}]^+). \end{aligned} \tag{4}$$

The dynamic program recursion is easy to interpret. We decide on the bids and assignments (as this is quality revelation) to maximize the current period's expected revenue and the future profits given some of our employees become unavailable.

Writing $\triangle V_{dt}^p(\boldsymbol{s}, \boldsymbol{x}_p) = V_{t+1}([\boldsymbol{s} - \boldsymbol{1}]^+) - V_{t+1}([\boldsymbol{s} - \boldsymbol{1} + d_p \boldsymbol{x}_p]^+)$, we have

$$\begin{aligned} V_t(\boldsymbol{s}) = V_{t+1}([\boldsymbol{s} - \boldsymbol{1}]^+) + \sum_{p \in P} \lambda_{pt} \max_{\substack{b_p \in \mathbb{R}^+ \\ \boldsymbol{x}_p \in \mathcal{F}_s^p}} &(k_p d_p b_p \\ - \boldsymbol{c}_p^T \boldsymbol{x}_p - \triangle V_{dt}^p(\boldsymbol{s}, \boldsymbol{x}_p)) &w_p(b_p, \bar{\boldsymbol{q}}_p^T \boldsymbol{x}_p). \end{aligned} \tag{5}$$

The problem is difficult as formalized in Proposition 6.

Proposition 6. *Assuming $P \neq NP$, the bidding-and-matching problem of the PSF cannot be approximated to any constant factor when* $\max_p k_p \geq 2$.

4. Bounds and Solution Methods

The dynamic program (5) is computationally intractable to solve exactly, as the state space explodes even for a small number of employees. We thus first present a linear approximation to the value function and a greedy policy with respect to the approximation. Based on this approximation, we derive a theoretical upper bound on the optimal expected revenue. We show that when projects require exactly one employee, this approach gives a performance guarantee of being within twice the optimal value (see Proposition 6). Subsequently, we derive two increasingly tighter upper bounds on the dynamic program, the motivation being that the tighter the bounds compared with the values from the policy, the greater our confidence that policies derived from the bounds reflect the behavior of an optimal solution. The first bounding technique is based on a nonlinear program (NLP), akin to the deterministic version of the problem, that is an upper bound on the value function. The second bound is obtained via restricting the variables of the linear programming formulation of the dynamic program to linear policies while retaining the nonlinearity of the revenue function and calculating it using various optimization techniques. In Section 5, we first investigate how the bounds perform numerically and then use the greedy policy to study some questions of interest to a PSF.

4.1. Linear Approximation

A natural strategy for approximating the value function is to replace it with a linear approximation. The terms of the approximation can be interpreted as marginal values of employees, which we can approximate either via a recursive heuristic (Section 4.1.1) or via the solution of an affine approximation linear program (16) (in Section 4.3).

Let d_m be the maximum duration over all projects. Consider the approximation of the value function

$$\hat{V}_t(\boldsymbol{s}) = \hat{v}_c^t + \sum_{i \in I} \hat{v}_{i,s_i}^t,$$

parameterized by $\hat{v}_c^t, \hat{v}_{i,s_i}^t$ for $i \in I, t \in \{1, \ldots, T\}, s_i \in \{0, \ldots, d_m - 1\}$. The parameter $\hat{v}_{i,s_i}^t$ denotes the marginal value at time t of employee i if he or she were to be available in s_i periods. We use the vector notation $\hat{\boldsymbol{\nu}}_s^t$ to contain the corresponding information $\hat{v}_{i,s_i}^t$ for all individuals.

4.1.1. Recursive Heuristic.

Computing the optimal parameters $\hat{v}_{i,j}^t$ is not trivial, and in this section, we propose a fast recursive greedy heuristic as a tractable approximation, similar to the one in (Rusmevichientong et al. 2020) in the context of assortment optimization with fixed prices:

Initialization: Set $\hat{v}_c^T = 0$ and $\hat{v}_{i,j}^T = 0$ for all $i \in I, j \in \{0, \ldots, d_m - 1\}$.

Recursion: For $t = T - 1, \ldots, 1$:

1. For every project p with $\lambda_{pt} > 0$, compute the optimal allocation $\hat{\boldsymbol{x}}_p^t$ and bidding $\hat{b}_p^t$ under the assumption that all resources are available:

$$\begin{aligned} \left(\hat{\boldsymbol{x}}_p^t, \hat{b}_p^t\right) = \arg \max_{\substack{b_p \in \mathcal{R}^+ \\ \boldsymbol{x}_p \in \mathcal{F}_0^p}} &\Big(k_p d_p b_p - \boldsymbol{c}_p^T \boldsymbol{x}_p \\ &- \boldsymbol{1}^T \left(\hat{\boldsymbol{\nu}}_{\boldsymbol{0}}^{t+1} - \hat{\boldsymbol{\nu}}_{[\boldsymbol{0} - \boldsymbol{1} + d_p \boldsymbol{x}_p]^+}^{t+1}\right)\Big) w_p(b_p, \bar{\boldsymbol{q}}_p^T \boldsymbol{x}_p). \end{aligned}$$

2. Once $(\hat{x}_p^t, \hat{b}_p^t)$ is calculated for every project p, compute, for all $j \in \{t, \ldots t - d_m\}$,

$$\hat{v}_{i,j}^t = \hat{v}_{i,j-1}^{t+1} \quad \forall i \in I, j \in \{1, \ldots, d_m - 1\}, \tag{6}$$

$$\hat{\boldsymbol{\nu}}_{\mathbf{0}}^t = \hat{\boldsymbol{\nu}}_{\mathbf{0}}^{t+1} + \sum_{p \in P} \lambda_{pt} \Big(k_p d_p \hat{b}_p^t \gamma_p^t \circ \hat{\boldsymbol{x}}_p^t - \boldsymbol{c} \circ \hat{\boldsymbol{x}}_p^t - (\hat{\boldsymbol{\nu}}_{\mathbf{0}}^{t+1} - \hat{\boldsymbol{\nu}}_{[\mathbf{0}-\mathbf{1}+d_p x_p]^+}^{t+1}) \Big) w_p \Big(\hat{b}_p^t, \bar{\boldsymbol{q}}_p^T \hat{\boldsymbol{x}}_p^t \Big), \tag{7}$$

where $\circ$ denotes the Hadamard product (element-wise product), and the condition $\gamma_p^{t\,T} \hat{\boldsymbol{x}}_p^t = 1$ holds, with γ_p^t distributing the revenue among the participating individuals. We also set

$$\hat{v}_c^t = \mathbf{1}^T \hat{\boldsymbol{\nu}}_{\mathbf{0}}^t.$$

There is some flexibility in picking the parameter γ_p^t, which leaves space for different variants of the heuristic. However, we should always make a choice that satisfies the following condition to guarantee monotonicity of the marginal values (Lemma 3):

$$k_p d_p \hat{b}_p^t \gamma_p^t \circ \hat{\boldsymbol{x}}_p^t - \boldsymbol{c}_p \circ \hat{\boldsymbol{x}}_p^t - (\hat{\boldsymbol{\nu}}_{\mathbf{0}}^{t+1} - \hat{\boldsymbol{\nu}}_{[\mathbf{0}-\mathbf{1}+d_p x_p]^+}^{t+1}) \geq \mathbf{0}, \quad \forall p \in P. \tag{8}$$

Lemma 2. *There exist* $\gamma_p^t \geq \mathbf{0}$ *with* $\gamma_p^{t\,T} \hat{\boldsymbol{x}}_p^t = 1$ *that satisfies condition* (8).

From the definition of $\hat{\boldsymbol{\nu}}_{\mathbf{0}}^t$ the following is obvious.

Lemma 3. *If the choice of* γ_p^t *satisfies condition* (8), *then* $\hat{\boldsymbol{\nu}}_{\mathbf{0}}^t \geq \hat{\boldsymbol{\nu}}_{\mathbf{0}}^{t+1}$.

Proposition 7 yields a first upper bound on the optimal value of the dynamic problem.

Proposition 7. *If the choice of* γ_p^t *satisfies condition* (8), *then* $V_1(\mathbf{0}) \leq 2 \cdot \mathbf{1}^T \hat{\boldsymbol{\nu}}_{\mathbf{0}}^1$.

We call the right-hand side of the inequality in Proposition 7 the theoretical upper bound. In our numerical simulations, it turns out not to be particularly tight, but it gives us a reference value to compare tighter bounds with.

4.1.2. Greedy Policy.

In this section, we explore a simple greedy policy to calculate the bids and assignments under the quality-revelation model once we fix the marginal values according to the recursive heuristic of the previous section. This gives a lower bound on the value function. In Section 4.3, we will discuss an alternative way to compute the marginal values.

For fixed $\hat{\boldsymbol{\nu}}_s^t$, the greedy policy with respect to the approximation is given by

$$(\overline{x_p^t}(s), \overline{b_p^t}(s)) \in \arg\max_{\substack{x_p \in \mathcal{F}_s^p \\ x_p \in \mathcal{F}_s^p}} \Big(k_p d_p b_p - \boldsymbol{c}_p^T \boldsymbol{x}_p - \mathbb{1}_s^{t\,T} \Big(\hat{\boldsymbol{\nu}}_{\mathbf{0}}^{t+1} - \hat{\boldsymbol{\nu}}_{[\mathbf{0}-\mathbf{1}+d_p \boldsymbol{x}_p]^+}^{t+1} \Big) \Big) w_p \Big(b_p, \bar{\boldsymbol{q}}_p^T \boldsymbol{x}_p \Big), \tag{9}$$

where the dependence on the state is through $\mathcal{F}_s^p$, $\mathbb{1}_s^t$.

Proposition 6 shows the problem cannot be approximated to any constant factor even when projects are restricted to a maximum of two employees. We now show that if projects need at most one employee, the greedy policy with respect to the value function approximation $\hat{V}$ is guaranteed to obtain at least 50% of the optimal total profit.

Theorem 1. *If* $k_p = 1$ *for all* $p \in P$, *the total expected profit of the greedy policy with respect to the value function approximation* $\hat{V}$ *is at least* 50% *of the optimal.*

The proofs of Proposition 7 and Theorem 1 follow similar lines of argument as lemma 3.3 and theorem 3.2 of Rusmevichientong et al. (2020). Next we examine the single-stage problem that appears in the computation of $\hat{\boldsymbol{\nu}}_s^t$ and the greedy policy with respect to the approximation.

4.1.3. Implementing the Single-Stage Problem.

We want to solve

$$\max_{\substack{b \in \mathbb{R}^+ \\ \boldsymbol{x} \in \mathcal{F}_s^p}} (k_p d_p b - \tilde{\boldsymbol{c}}^T \boldsymbol{x}) w_p (b, \bar{\boldsymbol{q}}_p^T \boldsymbol{x}), \tag{10}$$

where b is the bid and $\tilde{\boldsymbol{c}} = \boldsymbol{c}_p + (\hat{\boldsymbol{\nu}}_{\mathbf{0}}^{t+1} - \hat{\boldsymbol{\nu}}_{[\mathbf{0}-\mathbf{1}+d_p \mathbf{1}]^+}^{t+1})$ is the base cost plus the opportunity cost of the resources.

It is convenient to rewrite (10) as

$$\max_{\substack{b \in \mathbb{R}^+ \\ \boldsymbol{x} \in \mathcal{F}_s^p}} \left\{ (k_p d_p b - C) w_p \left(b, \frac{Q}{k_p} \right) \middle| \boldsymbol{q}_p^T \boldsymbol{x} = Q, \ \tilde{\boldsymbol{c}}^T \boldsymbol{x} = C \right\}. \tag{11}$$

Note that it is natural to assume integrality for $\boldsymbol{q}_p$, but not for $\tilde{\boldsymbol{c}}$ because of the inclusion of the opportunity costs. The function $r(b, C, Q) = (k_p d_p b - C) \left(1 / \left(1 + e^{\beta_0^p + \beta_b^p b - \frac{\beta_q^p}{k_p} Q} \right) \right)$ is log-concave, as can be easily verified. With the feasible assignment set $\mathcal{F}_s^p$ as in (2), we can solve (11) via a two-step procedure:

1. Solve the two-constraint knapsack problem

$$\max_{\substack{\boldsymbol{x} \in \{0,1\}^{|I|} \\ \boldsymbol{x} \leq \mathbb{1}_s}} \{ \boldsymbol{q}_p^T \boldsymbol{x} : \tilde{\boldsymbol{c}}^T \boldsymbol{x} \leq C, \quad \mathbf{1}^T \boldsymbol{x} = k_p \} \tag{12}$$

for every $0 \leq C \leq \mathbf{1}^T \tilde{\boldsymbol{c}}$ producing a Pareto front of (C, Q) pairs.

2. For each Pareto-optimal (C, Q), maximize the univariate log-concave function $r(\cdot, C, Q)$ obtaining a corresponding bidding price. Pick the Pareto-optimal (C, Q) and the corresponding bid price that yields the overall maximum.

In the online appendix, we illustrate how to produce all Pareto points of Step 1 in one go by utilizing a variation of the dynamic programming algorithm for knapsack problems (Kellerer et al. 2004) after an appropriate transformation. The single-stage problem turns out to be computationally fast in practice.

Remark 4. For more complex feasible assignment sets, as mentioned in Remark 3, Step 1 of the above procedure would involve the solution of a series of binary programs to obtain the Pareto front.

4.2. Deterministic Upper Bound

In this section, we develop a compact deterministic upper bound akin to the deterministic linear programming bound of network revenue management (Talluri and van Ryzin 2004), but with a nonlinear objective function. We effectively treat the arrivals and wins of projects to be deterministic and equal to their expectations. Then, Jensen's inequality shows that we can obtain an upper bound via the solution of a convex nonlinear program. The advantage of this bound is that it can be solved using standard nonlinear programming packages. However, as we show in Section 5, the bound is not as tight as the more complicated bound of Section 4.3.

Recall that project p requires k_p individuals and lasts d_p intervals. Let us define a 0–1 incidence matrix A with $|I| \times |P| \times T$ columns and $|I| \times T$ rows, representing resource usage. The column (i, p, t), corresponding to an assignment of individual i to project p at time t, will have ones in rows (i, t) to $(i, t + d_p - 1)$, corresponding to employee i being occupied from time t up to and including time $t + d_p - 1$.

Consider the nonlinear convex deterministic program

$$\begin{aligned} \max_{\substack{w \in \mathbb{R}^{|P| \cdot T} \\ z \in \mathbb{R}^{|I| \cdot |P| \cdot T}}} \quad & \sum_{t=1}^{T} \sum_{p \in P} k_p d_p \Bigg[\frac{\beta_q^p}{\beta_b^p} \bar{\boldsymbol{q}}_p^T z_p^t - \frac{\beta_0^p}{\beta_b^p} \lambda_p^t w_p^t \\ & \qquad + \frac{1}{\beta_b^p} \lambda_p^t w_p^t \log\left(\frac{1 - w_p^t}{w_p^t} \right) \Bigg] \\ \text{s.t.} \quad & Az \le \mathbf{1} \\ & \mathbf{1}^T z_p^t = k_p w_p^t, \quad z_p^t \le \lambda_{pt} \mathbf{1}, \quad z_p^t \le w_p^t \mathbf{1} \quad \forall t, p \\ & \mathbf{0} \le z \le \mathbf{1} \end{aligned} \qquad (DW^u)$$

where the decisions are expressed in terms of the assignment z_p^t to project p at time t, and the corresponding win probability w_p^t, rather than the bid b_p^t—a one-to-one relationship. We show that the optimal value of (DW^u) yields an upper bound to the optimal value of the dynamic program.

Proposition 8. *The optimal value of* (DW^u) *is greater than or equal to* $V_1(\mathbf{0})$.

In practice, we can tighten the above bound by simulating project arrivals and averaging the optimal solutions of the corresponding NLPs. Another advantage of simulation is that the NLPs have smaller sizes, as we only have to consider the project that actually arrived in every period, counterbalancing the effort of solving multiple NLPs.

To that end, assume that at a simulation iteration a project $p(t) \in P$ arrived at time t. Now our incidence matrix A has $|I| \times T$ columns and $|I| \times T$ rows. The column (i, t), corresponding to an assignment of individual i at time t, will have ones in rows (i, t) to $(i, t + d_{p(t)} - 1)$, corresponding to employee i being occupied from time t up to and including time $t + d_{p(t)} - 1$. The corresponding NLP is

$$\begin{aligned} \max_{\substack{w \in \mathbb{R}^{T} \\ z \in \mathbb{R}^{|I| \cdot T}}} \quad & \sum_{t=1}^{T} k_{p(t)} d_{p(t)} \left[\frac{\beta_q^{p(t)}}{\beta_b^{p(t)}} \bar{\boldsymbol{q}}_{p(t)}^T z^t - w^t \frac{\beta_0^p}{\beta_b^p} + \frac{1}{\beta_b^p} w_t \log\left(\frac{1 - w}{w} \right) \right] \\ \text{s.t.} \quad & Az \le \mathbf{1}, \\ & \mathbf{1}^T z^t = k_{p(t)} w^t, \quad z_t \le \mathbf{1}, \quad z_t \le w^t \mathbf{1} \quad \forall t, \\ & \mathbf{0} \le z \le \mathbf{1}, \end{aligned} \qquad (DW_{p(t)}^u)$$

where $p(t)$ is the project that arrived at time t. In the above, to avoid notational complications, we assumed a project arrives in every period. Time periods where no project arrives can be left out of the summation in the objective, and the corresponding variables can be dropped.

The optimal value of ($DW_{p(t)}^u$) is a random quantity, as it depends on the random arrivals.

Proposition 9. *Let* $O(DW_{p(t)}^u)$ *be the optimal value of* $DW_{p(t)}^u$ *and* $O(DW^u)$ *be the optimal value of* (DW^u). *Then*

$$O(DW^u) \ge \mathbb{E}[O(DW_{p(t)}^u)] \ge V_1(\mathbf{0}).$$

We omit the proof as it is similar to the last step of the proof of Proposition 8.

4.3. Affine Bound via Linear Programming

In this section, we investigate solving for the marginal values in the approximation using linear programming as pioneered in revenue management by Adelman (2007). This bound is in general tighter than the bound of Section 4.2 and gives us confidence that our heuristic algorithms for the bidding and assignment are reasonably close to the optimal solution.

Consider the linear program

$$\begin{aligned} \min \quad & \tilde{V}_1(\mathbf{0}) \\ \text{s.t.} \quad & \tilde{V}_t(\mathbf{s}) \ge \tilde{V}_{t+1}([\mathbf{s} - \mathbf{1}]^+) + \\ & + \sum_p \lambda_{pt} \Big(k_p d_p b_p^t - \mathbf{c}_p^T \mathbf{x}_p^t - \triangle \tilde{V}_{dt}^p(\mathbf{s}, \mathbf{x}_p^t) \Big) w_p(b_p^t, \bar{\mathbf{q}}_p^T \mathbf{x}_p^t) \\ & \qquad \forall t, \mathbf{s}, b_p^t, \mathbf{x}_p^t \in \mathcal{F}_{\mathbf{s}}^p, \end{aligned} \qquad (13)$$

where the decision variables of the LP are the $\tilde{V}_t(\mathbf{s})$'s.

Solving (13) yields an optimal solution to the dynamic program, but is intractable because of the exponential number of variables *and* constraints. We can obtain an upper bound by solving instead the LP corresponding to the approximate dynamic program that reduces the number of variables to something we can handle numerically:

$$\min \quad \tilde{\nu}_c^1 + \mathbf{1}^T \tilde{\boldsymbol{\nu}}_{\mathbf{0}}^1 \qquad (14)$$

$$\begin{aligned} \text{s.t.} \quad & \tilde{\nu}_c^{t+1} - \tilde{\nu}_c^t + \mathbb{1}_{\mathbf{s}}^{t\,T} (\tilde{\boldsymbol{\nu}}_{\mathbf{0}}^{t+1} - \tilde{\boldsymbol{\nu}}_{\mathbf{0}}^t) + (\mathbf{1} - \mathbb{1}_{\mathbf{s}}^t)^T (\tilde{\boldsymbol{\nu}}_{\mathbf{s-1}}^{t+1} - \tilde{\boldsymbol{\nu}}_{\mathbf{s}}^t) + \\ & \sum_{p \in P} \lambda_{pt} \Big(k_p d_p b_p^t - \boldsymbol{c}_p^T \boldsymbol{x}_p^t - \mathbf{1}^T \Big(\tilde{\boldsymbol{\nu}}_{\mathbf{0}}^{t+1} - \tilde{\boldsymbol{\nu}}_{[\mathbf{0} - 1 + d_p \boldsymbol{x}_p]^+}^{t+1} \Big) \Big) \\ & \times w_p(b_p^t, \bar{\boldsymbol{q}}_p^T \boldsymbol{x}_p^t) \le 0, \qquad \forall t, \mathbf{s}, b_p^t, \boldsymbol{x}_p^t \in \mathcal{F}_{\mathbf{s}}^p. \end{aligned} \qquad (15)$$

To move from (13) to (14)–(15) we constrain the fully general decision variables $\tilde{V}_t(\boldsymbol{s})$ to be affine in the state. Thus, the problem (14)–(15) is a restriction of (13) and indeed yields an upper bound. We can further restrict the problem, retaining the bounding property, by adding the constraints $\tilde{\boldsymbol{\nu}}_j^t = \tilde{\boldsymbol{\nu}}_{j-1}^{t+1}$ for all $t, j > 0$, which can be interpreted as saying that the marginal value of an employee that is to become available after $j > 0$ periods at time t is equal to the marginal value of the same employee that is to become available after $j-1$ periods at time $t+1$. In view of these additional constraints, we can drop the term $(\mathbf{1} - \mathbb{1}_s^t)^T(\tilde{\boldsymbol{\nu}}_{s-1}^{t+1} - \tilde{\boldsymbol{\nu}}_s^t)$ from (15). The constraints now depend on the state $\boldsymbol{s}$ only via the available resources in the definitions of $\mathcal{F}_s^p$, $\mathbb{1}_s^t$, a fact that will be critical computationally:

$$\min\ \tilde{\nu}_c^1 + \mathbf{1}^T\tilde{\boldsymbol{\nu}}_0^1 \tag{16}$$

$$\text{s.t. } \tilde{\nu}_c^{t+1} - \tilde{\nu}_c^t + \mathbb{1}_s^{t\,T}(\tilde{\boldsymbol{\nu}}_0^{t+1} - \tilde{\boldsymbol{\nu}}_0^t) + \sum_{p\in P}\lambda_{pt}\Big(k_p d_p b_p^t - \boldsymbol{c}_p^T\boldsymbol{x}_p^t - \mathbf{1}^T\Big(\tilde{\boldsymbol{\nu}}_0^{t+1} - \tilde{\boldsymbol{\nu}}^{t+1}_{[\mathbf{0}-\mathbf{1}+d_p\boldsymbol{x}_p]^+}\Big)\Big) \times w_p(b_p^t, \bar{\boldsymbol{q}}_p^T\boldsymbol{x}_p^t) \le 0, \qquad \forall t, \boldsymbol{s}, b_p^t, \boldsymbol{x}_p^t \in \mathcal{F}_s^p, \tag{17}$$

$$\tilde{\boldsymbol{\nu}}_j^t = \tilde{\boldsymbol{\nu}}_{j-1}^{t+1}, \qquad \forall t, j > 0. \tag{18}$$

The LP (16)–(18) has $T\cdot d_m\cdot|I| + T$ variables but an exponential number of constraints. Because now the dependence of (17) on the state is via the available resources, to consider the constraints corresponding to all states, we just need to consider all possible combinations of resource availability. Using $\boldsymbol{y}$ to denote the vector of available resources and $\boldsymbol{X} = \boldsymbol{x}_1, \ldots, \boldsymbol{x}_{|P|}$, we can rewrite (17) as

$$\tilde{\nu}_c^{t+1} - \tilde{\nu}_c^t + g_{\tilde{\boldsymbol{\nu}},t}(\boldsymbol{y}, \boldsymbol{X}, \boldsymbol{b}) \le 0, \quad \forall \boldsymbol{b} \in \mathbb{R}_+^{|P|},\ \forall(\boldsymbol{y}, \boldsymbol{X}) \in \mathcal{Z}, \tag{19}$$

where

$$g_{\tilde{\boldsymbol{\nu}},t}(\boldsymbol{y}, \boldsymbol{X}, \boldsymbol{b}) = (\tilde{\boldsymbol{\nu}}_0^{t+1} - \tilde{\boldsymbol{\nu}}_0^t)^T\boldsymbol{y} + \sum_p \lambda_{pt}\Big(k_p d_p b_p - \boldsymbol{c}_p^T\boldsymbol{x}_p - \mathbf{1}^T\Big(\tilde{\boldsymbol{\nu}}_0^{t+1} - \tilde{\boldsymbol{\nu}}^{t+1}_{[\mathbf{0}-\mathbf{1}+d_p\boldsymbol{x}_p]^+}\Big)\Big) w_p(b_p, \bar{\boldsymbol{q}}_p^T\boldsymbol{x}_p),$$

$$\mathcal{Z} = \{(\boldsymbol{y}, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_{|P|}) \mid \boldsymbol{y} \in \{0,1\}^{|I|}, \boldsymbol{x}_p \in \mathcal{X}^p, \boldsymbol{x}_p \le \boldsymbol{y}\},$$

$$\mathcal{X}^p = \mathcal{F}_{\mathbf{0}}^p = \{\boldsymbol{x} \in \{0,1\}^{|I|} \mid \mathbf{1}^T\boldsymbol{x}_p = k_p\}.$$

If we relax $\mathcal{X}^p$ to its linear relaxation $\mathcal{X}_R^p$, we are effectively increasing the number of constraints. It follows that the optimal value of the corresponding restricted version of (16)–(18) remains a valid upper bound on the dynamic program. We aim to solve this restricted version of (16)–(18) by constraint generation, generating cuts iteratively by solving for every t the separation problem

$$\max\{g_{\tilde{\boldsymbol{\nu}},t}(\boldsymbol{y}, \boldsymbol{X}, \boldsymbol{b}) | \boldsymbol{b} \in \mathbb{R}_+^{|P|}, \boldsymbol{y} \in \{0,1\}^{|I|}, \boldsymbol{x_p} \in \mathcal{X}_R^p, \boldsymbol{x}_p \le \boldsymbol{y}\}, \tag{20}$$

where $\tilde{\boldsymbol{\nu}}_s^t$ are obtained by solving the (master problem) bounding LP with a subset of the constraints.

Problem (20), however, is nonconvex and hard to solve. We use two procedures, a fast one that provides valid cuts but does not theoretically guarantee an upper bound, and a slower one that needs to be called only once, when the values of the relaxed LP converge, to provide an upper bound certificate.

4.3.1. Weak Fast Cuts. The cuts are generated separately for every period t. Given $\tilde{\boldsymbol{\nu}}_s^t$ values, we find a local maximum of (20) to obtain a local solution $(\boldsymbol{y}^*, \boldsymbol{X}^*, \boldsymbol{b}^*)$. The cut

$$\tilde{\nu}_c^t \ge \tilde{\nu}_c^{t+1} + \Big(\boldsymbol{y}^* - \sum_{p\in P}\lambda_{pt}w_p(b_p^{t*}, \bar{\boldsymbol{q}}_p^T\boldsymbol{x}_p^{t*})\boldsymbol{x}_p^{t*}\Big)^T\tilde{\boldsymbol{\nu}}_0^{t+1} - \boldsymbol{y}^{*T}\tilde{\boldsymbol{\nu}}_0^t + \sum_{p\in P}\lambda_{pt}\Big(k_p d_p b_p^{t*} + \Big(\tilde{\boldsymbol{\nu}}^{t+1}_{[\mathbf{0}-\mathbf{1}+d_p\boldsymbol{x}_p^{t*}]^+} - \boldsymbol{c}_p\Big)^T\boldsymbol{x}_p^{t*}\Big)w_p(b_p^{t*}, \bar{\boldsymbol{q}}_p^T\hat{\boldsymbol{x}}_p^{t*}),$$

where, to simplify the expression, we used

$$\mathbf{1}^T\Big(\tilde{\boldsymbol{\nu}}_0^{t+1} - \tilde{\boldsymbol{\nu}}^{t+1}_{[\mathbf{0}-\mathbf{1}+d_p\boldsymbol{x}_p^{t*}]}\Big) = \boldsymbol{x}_p^{t*T}\Big(\tilde{\boldsymbol{\nu}}_0^{t+1} - \tilde{\boldsymbol{\nu}}^{t+1}_{[\mathbf{0}-\mathbf{1}+d_p\boldsymbol{x}_p^{t*}]}\Big),$$

is valid. Indeed, for any feasible point $\boldsymbol{\nu}_c, \boldsymbol{\nu}_s^t$ of (16)–(18), we have

$$\nu_c^{t+1} - \nu_c^t + g_{\boldsymbol{\nu},t}(\boldsymbol{y}^*, \boldsymbol{x}^*, \boldsymbol{b}^*) \le \nu_c^{t+1} - \nu_c^t + \max_{\boldsymbol{y},\boldsymbol{x},\boldsymbol{b}} g_{\boldsymbol{\nu},t}(\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{b}) \le 0.$$

4.3.2. Upper Bound Certificate. As should be clear, solving the bounding LP (13) with only a subset of constraints will not lead to an upper bound on the optimal revenue. Here we provide an upper bound certificate using dual ideas. Proposition 10 provides an upper bound certificate for Problem (16)–(18) and thus also for the dynamic program.

Proposition 10. *Let* $\tilde{\boldsymbol{\nu}}_j^t$ *with* $\tilde{\boldsymbol{\nu}}_j^t = \tilde{\boldsymbol{\nu}}_{j-1}^{t+1}$ *for* $j > 0$. *If* $\hat{g}_t$ *is an upper bound on the optimal value of the separation problem* (20), *then an upper bound on the optimal value of* (16)–(18) *is given by* $\mathbf{1}^T\tilde{\boldsymbol{\nu}}_0^1 + \sum_{t=1}^{T-1}\hat{g}_t$.

Remark 5. Proposition 10 yields an upper bound for any choice of $\tilde{\boldsymbol{\nu}}_s^t$ satisfying the last constraints of the bounding LP. An unfortunate choice, however, of such $\tilde{\boldsymbol{\nu}}_s^t$ may lead to a weak bound. The generation of fast cuts and the solutions of the relaxed LP are thus used to select $\tilde{\boldsymbol{\nu}}_s^t$'s that will lead to better bounds.

Remark 6. The marginal values of $\tilde{\boldsymbol{\nu}}_j^t$ obtained by the upper bounding process can also be used to drive an approximate policy, in place of the $\hat{\boldsymbol{\nu}}_j^t$'s of *Section* 4.1.1.

To obtain the upper bounds $\hat{g}_t$ we will solve the dual of (20) with respect to the constraints $\boldsymbol{x}_p \le \boldsymbol{y}$,

$$\min_{\boldsymbol{\pi}\in\mathcal{R}_+^{I\times P}}\ \max_{\substack{\boldsymbol{x}_p\in\mathcal{X}_p^R\\ \boldsymbol{y}\in\{0,1\}^{|I|}\\ b\in\mathbb{R}_+^P}} g_{\tilde{\boldsymbol{\nu}},t}(\boldsymbol{y}, \boldsymbol{X}, \boldsymbol{b}) + \sum_{p\in P}(\boldsymbol{y} - \boldsymbol{x}_p)\boldsymbol{\pi}_p. \tag{21}$$

Using the definition of $g_{\tilde{\boldsymbol{\nu}},t}$, the inner problem decomposes and yields subproblems

$$\max_{y_i\in\{0,1\}} \left(\sum_{p\in P}\pi_{ip} + (\tilde{\nu}^{t+1}_{\mathbf{0}_i} - \tilde{\nu}^{t}_{\mathbf{0}_i})\right) y_i, \tag{22}$$

$$\max_{\substack{x_p^t\in[0,1]^{|I|}\\ b_p^t\in\mathbb{R}_+}} \lambda_{pt}\left(k_p d_p b_p^t - \boldsymbol{c}_p^T\boldsymbol{x}_p^t - \mathbf{1}^T\left(\hat{\boldsymbol{\nu}}^{t+1}_{\mathbf{0}} - \hat{\boldsymbol{\nu}}^{t+1}_{[\mathbf{0}-\mathbf{1}+d_p\boldsymbol{x}_p]^+}\right)\right) \times w_p(b_p^t, \bar{\boldsymbol{q}}_p^T\boldsymbol{x}_p^t) - \boldsymbol{\pi}_p^T\boldsymbol{x}_p \tag{23}$$

$$\text{s.t.}\quad \mathbf{1}^{\mathrm{T}}\boldsymbol{x}_p^t = k_p.$$

Problem (22) is trivially solved by setting $y_i = 1$ whenever its coefficient in (22) is positive and to $y_i = 0$ otherwise. Although Problem (23) is nonconvex, the nonconvexity can be isolated in one dimension and solved by a one dimensional branch-and-bound search. To that end, let

$$F(\rho) = \max_{\substack{x_p^t\in[0,1]^{|I|}\\ b_p^t\in\mathbb{R}_+}} \log\left[\lambda_{pt}\left(k_p d_p b_p^t - \boldsymbol{c}^T\boldsymbol{x}_p^t - \mathbf{1}^T\left(\hat{\boldsymbol{\nu}}^{t+1}_{\mathbf{0}} - \hat{\boldsymbol{\nu}}^{t+1}_{[\mathbf{0}-\mathbf{1}+d_p\boldsymbol{x}_p]^+}\right)\right) w_p(b_p^t, \bar{\boldsymbol{q}}_p^T\boldsymbol{x}_p^t)\right]$$
$$\text{s.t.}\quad \mathbf{1}^T\boldsymbol{x}_p^t = k_p,\quad \boldsymbol{\pi}_p^T\boldsymbol{x}_p^t \le \rho,$$

and note that the objective function of the defining problem is concave, because the logistic distribution w_p is log-concave. In turn, this implies concavity of $F(\cdot)$, as a perturbation function of a concave maximization problem with convex constraints.

Problem (23) can be reformulated as the univariate optimization problem

$$\max_{\rho\in[0,\mathbf{1}^T\boldsymbol{\pi}_p]} \{e^{F(\rho)} - \rho\}$$

with a nonconvex objective function. Note that evaluating $e^{F(\rho)} - \rho$ at a given ρ is an easy concave problem. To find the optimal ρ^* via branch and bound, given an interval $[\rho_l, \rho_u]$, we further need a concave overestimator of $e^{F(\rho)} - \rho$ in $[\rho_l, \rho_u]$, with the property that as the interval $[\rho_l, \rho_u]$ gets smaller, the overestimator becomes arbitrary tight (see Horst and Tuy 2013).

By replacing the exponential with its secant in the segment $[\rho_l, \rho_u]$, we obtain such a concave overestimator,

$$e^{F(\rho_l)} + \frac{e^{F(\rho_u)} - e^{F(\rho_l)}}{F(\rho_u) - F(\rho_l)}(F(\rho) - F(\rho_l)) - \rho,$$

and we observe that the overestimator is exact at the edges of the interval $[\rho_l, \rho_u]$. In turn, because of continuity, this implies that the overestimator can be made arbitrary tight by shrinking the corresponding interval. We emphasize that although branch and bound is in general an exponential algorithm, in our case, it is efficient, as we have isolated the nonconvexity of the problem in one variable.

Remark 7. The dual does not have to be solved to optimality. Any $\boldsymbol{\pi} \ge 0$ leads to a valid upper bound. In our numerical experiments, we use the subgradient algorithm with a modest number of iterations to calculate $\boldsymbol{\pi}$.

5. Numerical Study

In this section, we use our computational procedures to numerically investigate questions of great interest for a PSF, namely, what the right utilization level is and what employee types to hire. Our first experiment, in Section 5.1, however, is technical and focuses on examining the tractability and relative tightness of the different bounds as well as giving some insight on the optimality gap of the greedy policy. In the second experiment, in Section 5.2, we perform simulations based on the approximate solution of the dynamic program to explore the interplay between operational and hiring decisions, and in the third set of experiments, in Section 5.3, we look at optimal hiring. The focus is not so much on the specific insights, as they are dependent on parameters, but on illustrating the potential uses of our tool kit and that the value of our formulation and solution methods extends beyond just the bidding and assignment problem and can be useful in workforce analytics and staffing decisions.

We generate instances parameterized by the number of employees $|I|$, the number of project types $|P|$, durations d_{min} and d_{max}, and k, where k is a proxy for the number of employees per project. In each period, a project p arrives with probability λ_p, which is sampled from a uniform $[0, 1]$ distribution and subsequently normalized so that exactly one project arrives in each period, that is, $\sum_{p\in P}\lambda_p = 1$. The project duration d_p is taken to be a random integer between d_{min} and d_{max}, and $k_p = \lfloor \bar{k}_p \rfloor$, where $\bar{k}_p$ is sampled from a uniform distribution with support $[0.7k, 1.3k]$.

We assume we have three types of employees, A, B, and C, and each project type belongs to one of three classes, a, b, and c, all with equal probability. We emphasize that there are more than three project types, as a project type, apart from its class, is defined by its duration, its workforce requirements, and its sensitivity to quality and price. The suitability of an employee type to a project class is denoted by $\mathcal{S}_{ip}$ and takes values from Table 2. Furthermore, each employee has generic capability $\mathcal{C}_i$, which is an integer between one and five. The overall quality of assigning employee i to project p is $q_{ip} = \mathcal{C}_i \cdot \mathcal{S}_{ip}$; it is a combination of the inherent capability of the employee and the suitability to the project

Table 2. Employee Capabilities

	a	*b*	*c*
A	3	2	1
B	2	3	2
C	1	2	3

and takes values in [1, 15]. We assume all labor costs are sunk and set $c_p = 0$.

The price sensitivity β_b^p and the quality sensitivity β_q^p are sampled uniformly from [0.008, 0.012] and [0.3, 0.5], respectively, whereas we set $\beta_0^p = \epsilon - 1{,}000\beta_b^p + 9\beta_q^p$ with ϵ sampled uniformly from [−0.5, 0.5]. Figure 2 shows the winning probability as a function of the bid for different average qualities $\bar{Q}$. Note that $\epsilon = 0$ and the parameters β_b^p, β_q^p being at the centers of their corresponding intervals leads to a probability of 50% of winning the project if we bid 1,000 per workday and supply a bundle of employees with an average quality of nine.

5.1. Tightness of Bounds and Computational Performance

For our first experiment, we pick the type of an employee to be A with probability 50%, B with probability 30%, and C with probability 20%. The capabilities are picked randomly, with equal probabilities. The computations are to examine the tractability and relative tightness of the different bounds, as well as to give some insight on the optimality gap of the greedy policy.

We run our experiments on a Linux workstation with 10 cores clocked at 2.8 GHz and 64 gigabytes of memory. All LPs are solved via Gurobi (Gurobi Optimization 2022), and all NLPs via SNOPT (Gill et al. 2005).

We solve instances for eight project types and 50 periods, and we scale the number of individuals from 15 to 75. Because exactly one project arrives per period, as we increase the number of individuals, we also increase the durations and resource requirements of the project types, in order to maintain a reasonable balance of workload with demand. In this section, we report on one instance of each size, whereas in the online appendix, we provide an additional four instances for each problem size. We do not observe any qualitative difference in the results of the additional instances.

In our implementation of the bound of Section 4.3, we terminate the weak cut generation when the cumulative change in the value of the LP in 10 consecutive iterations is smaller than 0.33%. Subsequently, we compute an upper bounding certificate using 40 iterations of the subgradient algorithm for the outside minimization of (21). We note that these criteria for terminating the generation of cuts and the multiplier search may be

Table 3. Numerical Results for $|P| = 8$ and $T = 50$

Problem			Revenue/employee ($1,000)						Performance
$\lvert I\rvert$	k	*Duration*	Sim_h	Sim_{NLP}	Sim_{AFF}	UB_T	UB_{NLP}	UB_{AFF}	*Perf*
15	6	3–6	28.4 ± 0.21	27.4 ± 0.22	28.5 ± 0.2	48.8	39.7	33.7	84.6
15	7	4–7	25.5 ± 0.18	24.1 ± 0.19	25.4 ± 0.18	58.0	42.4	34.5	73.9
20	6	4–7	24.4 ± 0.21	23.6 ± 0.18	24.7 ± 0.19	40.5	33.4	29.3	84.3
20	8	4–7	26.3 ± 0.2	25.3 ± 0.21	26.3 ± 0.21	58.2	42.4	33.6	78.3
25	7	4–7	22.0 ± 0.17	21.1 ± 0.18	22.1 ± 0.16	38.8	31.8	27.1	81.5
25	8	5–8	26.8 ± 0.2	25.8 ± 0.19	26.9 ± 0.19	53.7	41.6	34.7	77.5
30	7	5–8	26.0 ± 0.19	24.9 ± 0.2	26.4 ± 0.21	44.6	35.9	31.1	84.9
30	9	5–8	25.1 ± 0.18	24.3 ± 0.18	25.7 ± 0.18	46.3	36.5	32.5	79.1
35	8	5–8	31.5 ± 0.21	30.5 ± 0.22	31.7 ± 0.22	52.1	42.3	37.7	84.1
35	9	6–9	28.2 ± 0.2	26.7 ± 0.19	28.3 ± 0.2	46.1	39.5	34.9	81.1
40	8	6–9	25.9 ± 0.2	23.8 ± 0.18	25.8 ± 0.22	43.5	36.0	31.5	82.2
40	10	6–9	28.2 ± 0.22	26.9 ± 0.2	28.5 ± 0.19	48.6	41.9	36.3	78.5
45	9	6–9	29.1 ± 0.18	27.8 ± 0.19	29.0 ± 0.2	48.9	40.1	36.2	80.4
45	10	7–10	24.2 ± 0.21	22.4 ± 0.16	24.6 ± 0.21	39.5	33.4	29.0	84.8
50	9	7–10	25.2 ± 0.21	24.5 ± 0.19	25.6 ± 0.21	38.5	34.3	30.5	83.9
50	11	7–10	28.5 ± 0.2	27.5 ± 0.2	28.9 ± 0.2	51.0	41.1	36.5	79.2
55	10	7–10	25.1 ± 0.22	23.9 ± 0.18	25.5 ± 0.2	39.8	35.2	31.2	81.7
55	11	8–11	24.4 ± 0.19	23.6 ± 0.17	25.1 ± 0.18	43.6	35.6	32.7	76.8
60	10	8–11	20.5 ± 0.19	20.2 ± 0.15	21.1 ± 0.18	32.7	26.7	24.9	84.7
60	12	8–11	28.1 ± 0.21	26.5 ± 0.19	28.4 ± 0.2	47.1	39.0	36.8	77.2
65	11	8–11	28.5 ± 0.2	26.4 ± 0.16	29.0 ± 0.2	46.4	38.8	36.7	79.0
65	12	9–12	25.9 ± 0.2	23.9 ± 0.18	26.0 ± 0.21	44.9	35.7	33.7	77.2
70	11	9–12	29.4 ± 0.21	27.6 ± 0.19	29.4 ± 0.24	46.0	38.8	37.3	78.8
70	13	9–12	27.8 ± 0.2	26.6 ± 0.17	28.3 ± 0.22	51.4	40.4	38.9	72.8
75	12	9–12	28.2 ± 0.21	26.0 ± 0.18	28.0 ± 0.23	45.5	38.2	36.3	77.7
75	13	10–13	26.9 ± 0.21	25.0 ± 0.16	26.8 ± 0.24	43.6	36.7	35.0	76.9

Notes. The column UB_T shows the theoretical bound of Proposition 7. Column UB_{NLP} shows the upper bound obtained via the deterministic NLP method described in Section 4.2 with the tightening based on simulated arrivals: we simulate arrivals 50 times, and we report the upper end of the 99% confidence interval of $\mathbb{E}[O(DW_{p(t)}^{u})]$ as an upper bound. Column UB_{AFF} shows the upper bound obtained via the methodology of Section 4.3.

changed for a different trade-off between bound tightness and computation time. We have kept all parameters constant for all instances.

We also report 90% confidence intervals for the expected revenue generated by the greedy policy (9) via forward simulation of 500 paths. The confidence interval obtained when using the marginal values $\hat{\boldsymbol{\nu}}$ from the heuristic of Section 4.1 is given in column Sim_h of Table 3. In column Sim_{NLP}, we give the interval we obtain if we instead substitute the opportunity cost $\mathbb{1}_s^{t\,T}(\hat{\boldsymbol{\nu}}_{\mathbf{0}}^{t+1} - \hat{\boldsymbol{\nu}}_{[\mathbf{0}-\mathbf{1}+d_p x_p]^+}^{t+1})$ in (9) with the opportunity cost of assigning x_p at time t for duration d_p, as estimated via the dual solutions of $(DW_{p(t)}^u)$, averaged over the 50 runs. Column Sim_{AFF} contains the interval corresponding to the marginal values $\tilde{\boldsymbol{\nu}}$ coming from the upper bounding LP of Section 4.3. In the last column, *Perf*, we define performance as the best of the three policies using the ratio $\frac{\max\{Sim_{LP}, Sim_h, Sim_{NLP}\}}{\min\{UB_{AFF}, UB_{NLP}, UB_T\}}$. Because in the denominator we substitute the best upper bound in place of the unknown optimal expected revenue, the reported performance metric is a conservative estimate.

It is evident that both numerical bounds consistently outperform the theoretical bound of Proposition 7, although the affine bound is consistently tighter than the NLP bound. Furthermore, the policy implied by the affine bound is in general generating more revenue, but not via a large margin. As the heuristic is much more efficient, it might become a sensible choice if no upper bound is needed and computational resources are limited. The (classical) policy implied via the NLP is clearly dominated.

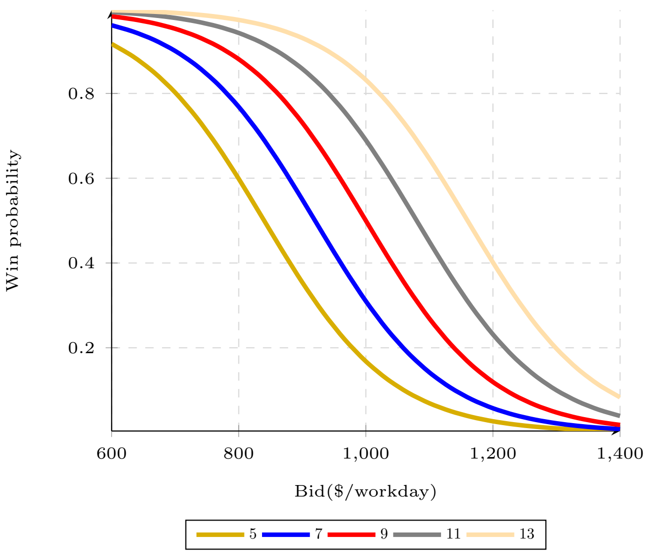
Computational times of all procedures can be found in Table 4. We implemented a parallel version of all procedures in this paper, and we report the impact on computational efficiency of using 10 parallel threads. Parallelizing the forward simulations is straightforward, as we can run the 500 paths in parallel. The same goes for the simulation-based bound of UB_{NLP}, for which we simulate arrivals 50 times. In the computation of UB_{AFF}, we run concurrently per period (a) the local optimization problems (20) in the weak cut generation process and (b) the upper bound certificate process. The only part of the UB_{AFF} that cannot be parallelized is the LP (16)–(18), which we solve repeatedly during the constraint generation process.

Table 4. Elapsed Times for $|P| = 8$ and $T = 50$

Problem			Elapsed time: 10 threads			Elapsed time: 1 thread		
$\|I\|$	k	*Duration*	*Sim*	UB_{NLP}	UB_{AFF}	*Sim*	UB_{NLP}	UB_{AFF}
15	6	3–6	1	21	56 + 6	6	126	239 + 5
15	7	4–7	1	9	58 + 4	6	58	252 + 4
20	6	4–7	1	21	62 + 12	7	97	296 + 11
20	8	4–7	1	34	85 + 15	12	160	407 + 16
25	7	4–7	2	42	78 + 38	16	379	402 + 34
25	8	5–8	3	24	83 + 47	17	129	420 + 50
30	7	5–8	3	33	111 + 89	23	222	540 + 100
30	9	5–8	3	45	91 + 161	22	240	562 + 156
35	8	5–8	4	54	150 + 237	30	341	897 + 234
35	9	6–9	3	94	118 + 220	26	490	745 + 217
40	8	6–9	4	44	133 + 239	34	411	860 + 233
40	10	6–9	7	111	146 + 429	47	630	957 + 404
45	9	6–9	6	99	177 + 652	45	531	1,203 + 645
45	10	7–10	6	49	132 + 302	47	349	910 + 292
50	9	7–10	7	200	163 + 498	49	526	1,194 + 511
50	11	7–10	9	566	175 + 1,136	66	2,579	1,284 + 1,157
55	10	7–10	11	100	202 + 1,110	76	568	1,570 + 1,110
55	11	8–11	10	393	182 + 1,349	78	2,519	1,408 + 1,375
60	10	8–11	10	783	178 + 930	66	5,214	1,428 + 932
60	12	8–11	15	127	212 + 2,028	108	805	1,688 + 2,038
65	11	8–11	13	287	246 + 2,195	93	886	1,989 + 2,219
65	12	9–12	10	197	203 + 2,077	87	907	1,620 + 2,103
70	11	9–12	17	154	283 + 2,235	124	805	2,316 + 2,256
70	13	9–12	16	366	259 + 2,781	135	1,886	2,048 + 2,812
75	12	9–12	17	363	228 + 4,642	129	2,030	1,828 + 4,679
75	13	10–13	18	386	236 + 4,117	135	1,397	1,936 + 4,168

Notes. In column *Sim*, we report the elapsed time of the forward simulation based on the heuristic of Section 4.1. Elapsed times of the forward simulation when using marginal values based on the bounding procedures are of similar scale, but the corresponding upper bounds have to be computed first. For the affine bound, the running time in column UB_{AFF} is decomposed into two parts: the (parallelizable) time for the cut generation and bound verification plus the time spent in the LP (which dominates for larger problems).

Figure 2. (Color online) Win Probability for $\beta_b^p = 0.01$ and $\beta_q^p = 0.4$ for Different Values of $\bar{Q}$



The (500 paths of the) forward simulation using the heuristic is quite efficient and terminates for all instances in a couple of minutes, and in the parallel implementation, in less than 20 seconds. The NLP bound is also quite efficient, especially in the parallel version, where for all problems, we can compute it in under 800 seconds. The LP bound is more challenging computationally. Somehow surprisingly, the bottleneck is neither the nonlinear separation problem nor the rather involved verification step, which can be parallelized and take less than 300 seconds in total for all instances. In the latter case, the isolation of the nonconvexity to one variable plays a critical role in keeping the times of the branch-and-bound algorithm low. Rather, the bottleneck is the repetitive solution of the LP itself, which cannot be parallelized, grows in size, and becomes time-consuming to solve. Still, it is evident that the methodology is applicable to moderate-size instances, as all instances can be handled within a couple of hours.

5.2. Effect of the Labor Force

As a buildup to Section 5.3 on optimal hiring, we perform experiments to determine various workforce performance measures as a function of employee hiring. We use parameters $|P| = 8$, $T = 100$, $d_{min} = 8$, $d_{max} = 14$, and $k = 6$, and vary the number of resources $|I|$ between 15 and 75. The first 15 employees are one of each type of each capability. The rest are generated randomly as described in Section 5.1, and thus the pool includes both high- and low-quality employees. This is in contrast with the next section, where we will be selecting employees (at different costs) to optimize profit. We simulate 500 paths, and we collect statistics on the performance of the greedy policy implied by the heuristic as we add employees. On average, in every period, a demand of $D = \sum_p \lambda_p k_p d_p$ workdays arrives, which, for this particular experiment, is just under 53. We report the workforce as a percentage of D. As we start with 15 employees, one of each type, the minimum workforce size we consider is $\frac{15}{53}$.

In Figure 3, we observe that as the workforce increases, both the bid and the win probability increase. With more employees, we have more flexibility to assign quality bundles, and thus we win projects more often even if we bid higher. A second observation is that although the utilization drops as expected with an increase in workforce, the drop is subproportional. This is due to the increased probability of winning projects as well as the decrease in cases where we do not have enough personnel to staff a project. The latter is a dominating factor when we are low on resources. Of course, if we look at utilization separately for each capability group, the utilization of weak employees drops very fast as stronger employees become available. The revenue per day per employed person initially increases, as a very small pool of employees results in inefficient teams and a high probability of missed opportunities, but relatively early, at a workforce of around $0.55D$, it starts to decrease as utilization approaches 70%.

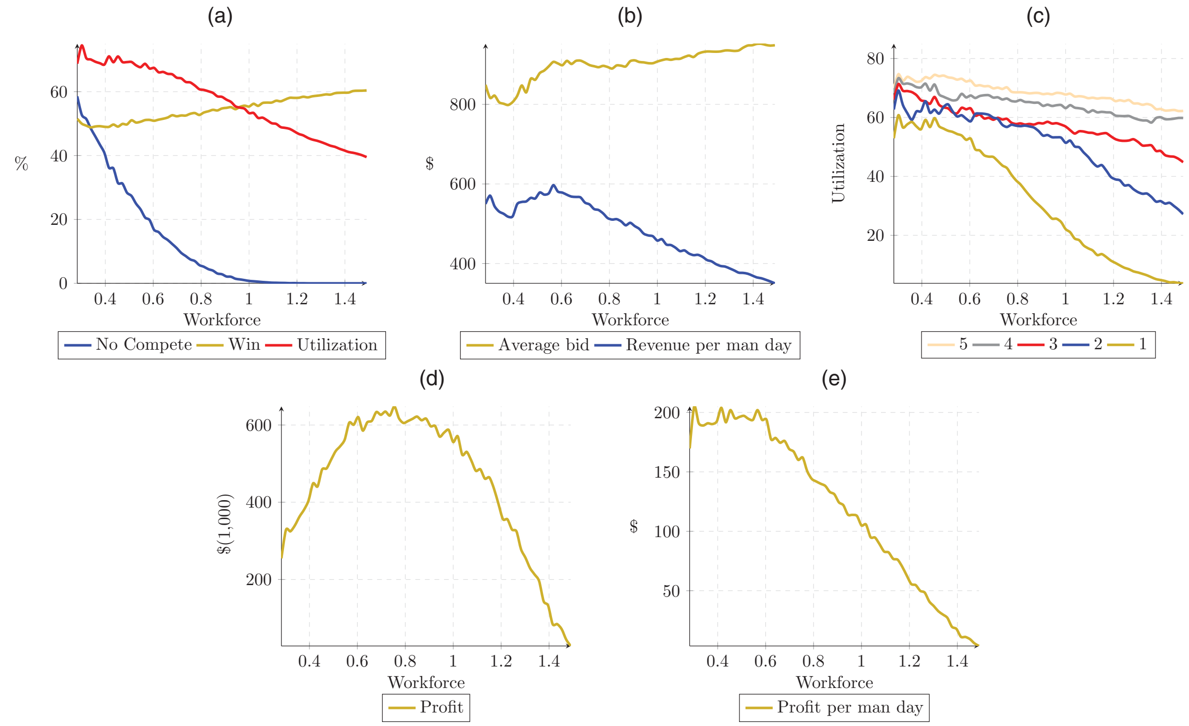
Let us now assume that the total labor cost σ_i per person-day of an employee depends on their capability only, and these costs are \$250, \$350, \$450, \$500, and \$700, respectively, for employees of capabilities one to five. With these labor costs, the profit and profit per person-day for the random selection (after the first 15) of employees is given in panels (d) and (e) of Figure 3. We observe that the optimal profit is realized for a workforce between $0.65D$ and $0.8D$. Any of these choices is close to optimal (modulo random hiring). The corresponding operational policies are similar in terms of the average bid, where the higher labor cost as we approach $0.8D$ is compensated via higher competing and project win ratios, due to the increased availability of resources, which also has a secondary effect on assignment quality. Eventually, the increase in revenue cannot counterbalance the increased labor costs, and the profits rapidly drop.

5.3. Hiring Decisions

In Section 5.2, we explored the effect of the labor force on utilization and profit, when employee capability is randomly distributed. Here we explore how we can use our framework to make informed hiring decisions.

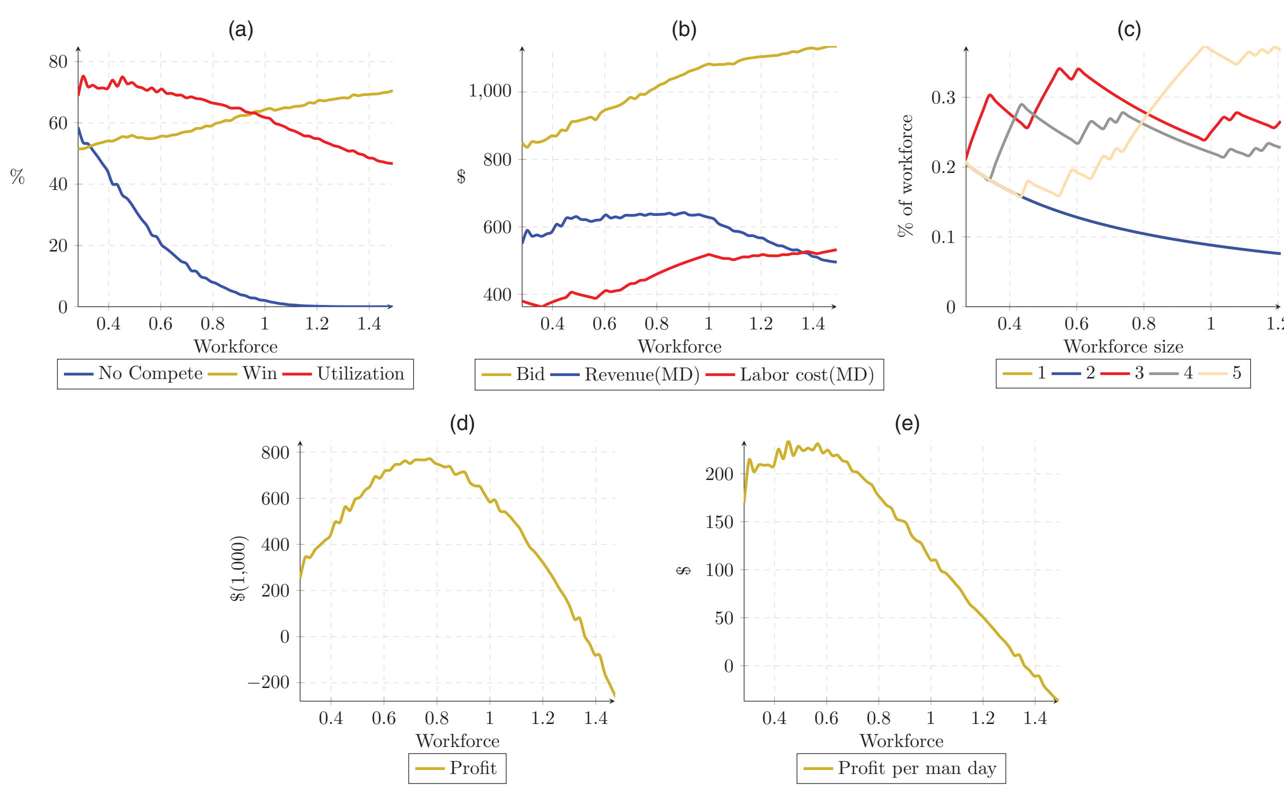
We perform a simulation to staff the firm. We keep all parameters identical to those in Section 5.2 ($|P| = 8$, $T = 100$, $d_{min} = 8$, $d_{max} = 14$, and $k = 6$). Just like in

Figure 3. (Color online) Effect of Increasing Randomly Hired Workforce



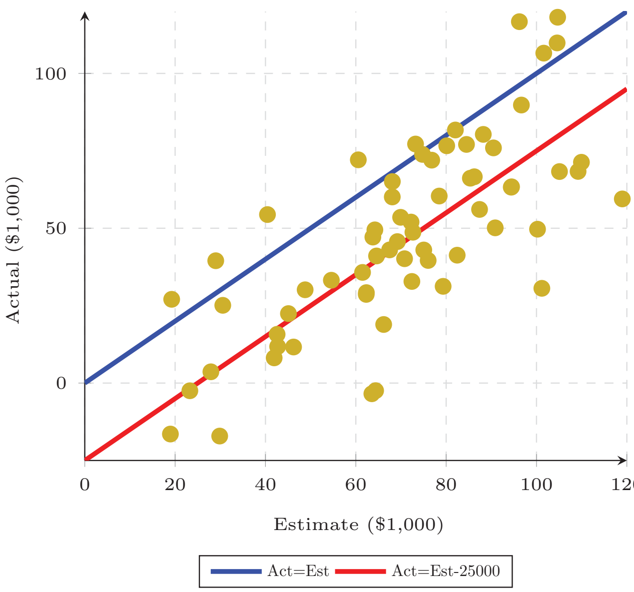
Notes. Panel (a) shows the percentage of projects the firm did not compete for because of insufficient resources, the winning percentage when competing, and utilization. Panel (b) shows the average bid and revenue per employee per day. Remaining panels show the workforce decomposed by (c) utilization by capability level, (d) profit, and (e) profit per man day.

Figure 4. (Color online) Effect of Increasing Randomly Hired Workforce According to Hiring Rule (24)



Notes. Panel (a) shows the percentage of projects that we did not compete for because of insufficient resources, the winning percentage when competing, and utilization. Panel (b) shows the average bid and revenue per employee per day. Remaining panels show the workforce decomposed by (c) capability level, (d) profit, and (e) profit per man day.

Figure 5. (Color online) Increase in Revenues of the Next Hire



Note. The estimate using marginal values is optimistic on average by $25,000.

Section 5.2, initially we start with 15 employees, one for each combination of type (*A*–*C*) and capability (1–5). Then we hire employees, one at a time, as follows:

1. We estimate the increase in revenue from hiring an extra employee of type $j \in (a, b, c)$ and capability $l \in \{1, 2, 3, 4, 5\}$ to be $\mathcal{R}^+_{l,j} = \min_{i \in I}\{\hat{v}^0_{i,0} | \mathcal{C}(i) = l, type(i) = j\}$; that is, if we have multiple employees of the same type/capability, we take the minimum marginal value of those employees to be an estimate for the increase in revenue from hiring one more.
2. We make the hire that maximizes

$$\mathcal{R}^+_{l,j} - \sigma_l. \tag{24}$$

In Figure 4, we observe that the optimal profit is achieved with a workforce of around 0.75*D*, with an approximate 20% increase in profit compared with that achieved by random hiring, whereas the corresponding utilization is slightly below 70%.

Initially, as we increase the workforce up to 0.45*D*, the utilization does not drop and stays close to 75%. As long as we cannot compete for most of the projects, we can effectively just scale up. The average labor cost, which reflects average workforce capability, stays fairly constant until we reach 0.6*D* and then starts to increase. This reflects the initial need to increase availability of resources to be able to compete for more projects, when quality is a secondary concern. Once we can compete for close to 80% of projects, quality becomes a primary concern, and we hire employees of higher quality (see Figure 4(c)).

We note that even in the early stages, although we do not focus on hiring capable employees, the average bid increases. This is because, in contrast to the initial random selection of employees, we select the type of our hires according to demand, increasing the quality of the matches.

The peak profit per employed person is achieved earlier for a workforce of 0.45*D*. The same goes for the revenue per employed person, which reaches the maximum at around 0.45*D* and stays fairly constant until 1*D*, when it starts decreasing. To maintain this revenue per person-day, however, we invest in increasingly more qualified employees, raising the average labor costs from 0.6*D* onward.

A point worth discussing is when to stop hiring. We caution that whereas the selection criterion of maximizing $\mathcal{R}_{l,j} - \sigma_l$ is sensible, a stopping criterion of stopping when this maximal value reaches zero would stop too late. The reason is that the estimate $\mathcal{R}_{l,j}$ is optimistic (see Figure 5).

6. Conclusions

In this paper, we examined some common workforce analytical and decision problems of a PSF in a revenue management framework. Using a stylized Markov chain model, we first argued that it is in the PSF's interest to be transparent on the supplied quality rather than wait for its delayed effects on reputation. By formulating the problem rigorously and developing good computational procedures, we are able to analyze a number of interesting questions and gain clear insights on the optimal number and mix of staff skills. For a managing partner of a PSF, our paper gives a tool kit for making bidding and assignment decisions under a quality-revelation model, which applies to important, large-scale projects. We hope that this paper spurs research into the operations of PSFs and brings modern analytical tools to help their management.

Acknowledgments

The authors thank the review team for many constructive comments that improved the paper.

Endnote

[1] We further elaborate on the different settings and their relevance to industry in the online appendix.

References

Adelman D (2007) Dynamic bid prices in revenue management. *Oper. Res.* 55(4):647–661.

Adelman D, Mersereau AJ (2013) Dynamic capacity allocation to customers who remember past service. *Management Sci.* 59(3): 592–612.

Bolton RN, Lemon KN, Bramlett MD (2006) The effect of service experiences over time on a supplier's retention of business customers. *Management Sci.* 52(12):1811–1823.

Boone T, Ganeshan R, Hicks RL (2008) Learning and knowledge depreciation in professional services. *Management Sci.* 54(7):1231–1236.

Cotterman JD, ed. (2016) *Compensation Plans for Law Firms*, 6th ed. (American Bar Association, Chicago).

DeCroix G, Long X, Tong J (2021) How service quality variability hurts revenue when customers learn: Implications for dynamic personalized pricing. *Oper. Res.* 69(3):683–708.

Gill PE, Murray W, Saunders MA (2005) SNOPT: An SQP algorithm for large-scale constrained optimization. *SIAM Rev.* 47(1):99–131.

Gilson R, Mnookin R (1985) Sharing among the human capitalists: An economic inquiry into the corporate law firm and how partners split profits. *Stanford Law Rev.* 37(2):313–392.

Gurobi Optimization (2022) *Gurobi Optimizer Reference Manual*. Accessed August 11, 2022, https://www.gurobi.com/documentation/9.5/refman/index.html.

Horst R, Tuy H (2013) *Global Optimization: Deterministic Approaches* (Springer, Berlin).

Kellerer H, Pferschy U, Pisinger D (2004) *Multidimensional Knapsack Problems* (Springer, Berlin), 235–283.

Maister DH (1982) Balancing the professional services firm. *Sloan Management Rev.* 24(1):15–30.

Maister DH (2012) *Managing the Professional Service Firm* (Simon and Schuster, New York).

Moreno A, Terwiesch C (2014) Doing business with strangers: Reputation in online service marketplaces. *Inform. Systems Res.* 25(4):865–886.

Mudrick H (1990) Partner compensation. *CPA J. Online*, http://archives.cpajournal.com/old/08656260.htm.

Nachum L (1996) Winners and losers in professional services: What makes the difference? *Service Indust. J.* 16(4):474–490.

Roels G, Karmarkar US, Carr S (2010) Contracting for collaborative services. *Management Sci.* 56(5):849–863.

Roth AV, Menor LJ (2003) Insights into service operations management: A research agenda. *Production Oper. Management* 12(2):145–164.

Rusmevichientong P, Sumida M, Topaloglu H (2020) Dynamic assortment optimization for reusable products with random usage durations. *Management Sci.* 66(7):2820–2844.

Sriram S, Chintagunta PK, Manchanda P (2015) Service quality variability and termination behavior. *Management Sci.* 61(11):2739–2759.

Talluri K, van Ryzin G (2004) *The Theory and Practice of Revenue Management* (Kluwer, New York).

Teece DJ (2003) Expert talent and the design of (professional services) firms. *Indust. Corporate Change* 12(4):895–916.

van Leeuwaarden J, Mathijsen B, Zwart B (2019) Economies-of-scale in many-server queueing systems: Tutorial and partial review of the QED Halfin-Whitt heavy-traffic regime. *SIAM Rev.* 61(3):403–440.

Zhan D, Ward AR (2019) Staffing, routing, and payment to trade off speed and quality in large service systems. *Oper. Res.* 67(6):1738–1751.

Kalyan Talluri is a professor of operations and analytics in the Imperial College Business School. His research interests are in pricing and revenue management, transportation, and health data and care delivery processes.

Angelos Tsoukalas is an assistant professor at the Rotterdam School of Management, Erasmus University. His research interests are in optimization theory, algorithms, and applications.