Competition Alleviates Present Bias in Task Completion

Aditya Saraf⋆, Anna R. Karlin⋆⋆, and Jamie Morgenstern

University of Washington, Seattle, WA, USA
{sarafa,karlin,jamiemmt}@cs.washington.edu

Abstract. We build upon recent work by Kleinberg, Oren, and Raghavan [10–12] that considers *present biased* agents, who place more weight on costs they must incur now than costs they will incur in the future. They consider a graph theoretic model where agents must complete a task and show that present biased agents can take exponentially more expensive paths than optimal. We propose a theoretical model that adds *competition* into the mix – two agents compete to finish a task first. We show that, in a wide range of settings, a small amount of competition can alleviate the harms of present bias. This can help explain why biased agents may not perform so poorly in naturally competitive settings, and can guide task designers on how to protect present biased agents from harm. Our work thus paints a more positive picture than much of the existing literature on present bias.

Keywords: present bias · behavioral economics · incentive design

1 Introduction

One of the most influential lines of recent economic research has been *behavioral* game theory [3, 9]. The majority of economics research makes several idealized assumptions about the behavior of rational agents to prove mathematical results. Behavioral game theory questions these assumptions and proposes models of agent behavior that more closely align with human behavior. Through experimental research [5, 6], behavioral economists have observed and codified several common types of cognitive biases, from loss aversion [9] (the tendency to prefer avoiding loss to acquiring equivalent gains) to the sunk cost fallacy [4] (the tendency to factor in previous costs when determining the best future course of action) to present bias [7] (the current topic). One primary goal of theorems in game theory is to offer predictive power. This perspective is especially important in the many computer science applications of these results, from modern ad auctions to cryptocurrency protocols. If these theorems are to predict human behavior, the mathematical models ought to include observed human biases. Thus,

⋆ Corresponding author
A full version is available from https://arxiv.org/abs/2009.13741
Research supported in part by NSF grant CCF-1813135.
⋆⋆ Research supported by NSF grant CCF-1813135.

rather than viewing behavioral game theory as conflicting with the standard mathematical approach, the experimental results of behavioral game theory can inform more sophisticated mathematical models. This paper takes a step towards this goal, building on seminal work of Kleinberg, Oren, and Raghavan [10–12] who formulated a mathematical model for planning problems where agents are present biased.

Present bias refers to overweighting immediate costs relative to future costs. This is a ubiquitous bias in human behavior that explains diverse phenomena. The most natural example is procrastination, the familiar desire to delay difficult work, even when this predictably leads to negative consequences later. Present bias can also model the tendency of firms to prefer immediate gains to long-term gains and the tendency of politicians to prefer immediate results to long-term plans. One simple model of present bias [10–12] is to multiply costs in the current time period by present bias parameter b when making plans. This model is a special case of hyperbolic discounting, where costs are discounted in proportion to how much later one would experience them. But even this special case suffices to induces *time-inconsistency*, resulting in a rich set of strategies consistent with human behavior.

Examples of time inconsistent behavior extend beyond procrastination. For example, one might undertake a project, and abandon it partway through, despite the underlying cost structure remaining unchanged. One might fail to complete a course with no deadlines, but pass the same course with weekly deadlines. Many people pay for a gym membership but never use it. Kleinberg and Oren [10] presented the key insight that this diverse range of phenomena can all be expressed in a single graph-theoretic framework, which we describe below.

Fix a directed, acyclic graph G, with designated source s and sink t. Refer to G as a task graph, where s is the start of the task and t the end. A path through this graph corresponds to a plan to complete the task; each edge represents one step of the plan. Each edge has a weight corresponding to the cost of that step.

The goal of an agent is to complete the task while incurring the least cost (i.e., to take the cheapest path from s to t). An optimal agent will simply follow such a cheapest path. A *naive* present biased agent with bias parameter b behaves as follows. At s, they compute their perceived cost for taking each path to t by summing the weights along this path with the first edge scaled up by $b > 1$. They choose the path with the lowest perceived cost and take *one* step along this path, say to v, and recompute their perceived cost along each the path from v to t. Notice that such an agent may choose a path at s, take one edge along that path, and then deviate away from it. This occurs because the agent believes that, after the current choice of edge, they will pick the path from v to t with lowest true cost. But once they arrive at v, their perceived cost of a path differs from the true cost, and they pick a path with lowest perceived cost. This is why the agents are considered naive: they incorrectly assume their future self will behave optimally, and thus exhibit time-inconsistent behavior. See Figure 1 for an example.

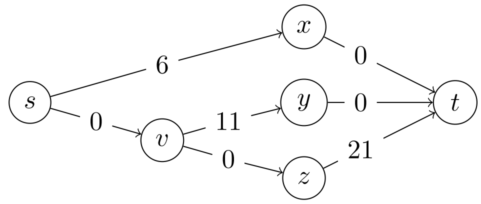

Fig. 1: The optimal path is (s, x, t) with total cost 6. However, an agent with bias $b = 2$ will take path (s, v, z, t), with cost 21. Importantly, when the agent is deciding which vertex to move to from s, they evaluate x as having total cost 12, while v has total cost 11. This is because they assume they will behave optimally at v by taking path (v, y, t). However, they apply the same bias at v and deviate to the worst possible path.

The power of this graph theoretic model is that it allows us to answer questions over a range of planning problems, and to formally investigate which tasks represent the "worst-case" cost of procrastination. This is useful both to understand how present-biased behavior differs from optimal behavior and to design tasks to accommodate present bias. We now briefly summarize the existing literature, to motivate our introduction of competition to the model.

1.1 Prior Work

The most striking result is that there are graphs where the *cost ratio* (the ratio of the optimal agent's cost to the biased agent's cost) is exponential in the size of the graph. In addition, all graphs with exponential cost ratio have a shared structure – they all have a large n-fan as a graph minor (and graphs without exponential cost ratio do not) [10, 15]. So this structure encodes the worst-case behavior for present bias in the standard model (and we later show how competition is especially effective in this graph). An n-fan is pictured in Figure 2.

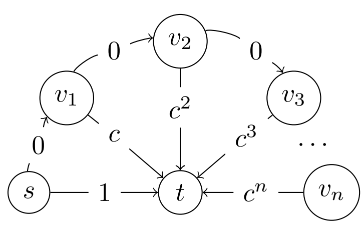

Fig. 2: A naive agent with bias $b > c > 1$ will continually choose to delay finishing the task.

The exponential cost ratio demonstrates the severe harm caused by present bias. How, then, can designers of a task limit the negative effects of present bias? Kleinberg and Oren [10] propose a model where a reward is given after finishing the task, and where the agent will abandon the task if at any point, they perceive the remaining cost to be higher than the reward. Unlike an optimal agent, a biased agent may abandon a task partway through. As a result, they give the task designer the power to arbitrarily delete vertices and edges, which can model deadlines. They then investigate the structure of *minimally motivating subgraphs* – the smallest subgraph where the agent completes the task, for some fixed reward. Follow-up work of Tang et al. [15] shows that finding *any* motivating subgraph is NP-hard. Instead of deleting edges, Albers and Kraft [2] consider the problem of spreading a fixed reward onto arbitrary vertices to motivate an agent, and find that this too is NP-hard (with a constrained budget). For other recent work involving present bias, see [1, 8, 13, 14, 16].

The above results all focus on *accommodating* present bias rather than *alleviating* it. By that, we mean that the approaches all focus on whether the agent can be convinced to complete the task – via edge deletion or reward dispersal – but not on guarding the agent from suboptimal behavior induced by their bias. [11] partially investigates the latter question in a model involving *sophisticated* agents, who plan around their present bias. They consider several types of *commitment devices* – tools by which sophisticated agents can constrain their future selves. However, these tools may require more powerful agents or designers and don't necessarily make sense for naive agents. We take a different approach – we show that adding competition can simultaneously explain why present-biased agents may not perform exponentially poorly in "natural" games and guide task designers in encouraging biased agents towards optimal behavior.

1.2 Our Model

In our model, a task is still represented by a directed, acyclic graph G, with a designated source s and sink t. There are two naive present-biased agents, A_1 and A_2, both with bias b, who compete to get to t first. The cost of a path is the sum of the weights along the path, and time is represented by the number of edges in the path, which we call the *length* of the path. In other words, each edge represents one unit of time. The first agent to get to t gets a reward of r; ties are resolved by evenly splitting the reward. Recall that naive agents believe that they will behave optimally in the future. Thus, an agent currently at u considers the cost to reach the target t to be $bc(u, v)$ plus the cost of the optimal path from v to t minus the reward of that path. More formally, let $\mathcal{P}(v \to t)$ denote the set of paths from $v \to t$ and let $P(s \to u)$ denote the path the agent has taken to u. Let $C_n(u, v)$ denote the remaining cost that the naive agent believes they will incur while at u and planning to go to v. The subscript n Then:

$$C_n(u,v) = b\cdot c(u,v) + \min_{P(v\to t)\in\mathcal{P}(v\to t)} c(P(v\to t)) - R_{A_2}(P(s\to u)\cup(u,v)\cup P(v\to t)), \quad (1)$$

where $c(P) = \sum_{e\in P} c(e)$ denotes the cost of path P and $R_{A_2}(P)$ denotes the reward of taking path P from s to t. This reward depends on the path the other agent A_2 takes. Specifically, if A_2 takes a path of length k, and $Q := P(s \to u)\cup(u,v)\cup P(v \to t)$ is a path of length ℓ, then $R(Q)$ is r if $\ell < k$, $r/2$ if $\ell = k$ and

0 if $\ell > k$. We will often rewrite the second term in (1), for ease of notation, as $\min_{P_v} c(P_v) - R_{A_2}(P_{s\to u,v\to t})$. We sometimes refer instead to the naive agent's *utility*, which is the negation of this cost. Given this cost function, the naive agent chooses the successor of node u via $S(u) = \operatorname{argmin}_{v:(u,v)\in E} C_n(u,v)$. See Figure 3 for an example.

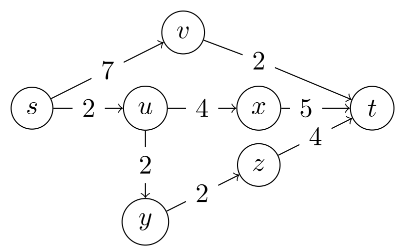

Fig. 3: Suppose $r = 5$, the bias $b = 2$, and assume A_2 takes path (s,u,x,t). Then at s, A_1 prefers to take u for perceived cost $4+4+5-2.5 = 10.5$. Notice that, due to the reward, the path A_1 believes he will take from u is (u,x,t), despite (u,y,z,t) having lower cost. However, at u, A_1 evaluates the lower path to be cheaper, despite losing the race. This shows that a reward of 5 does not ensure a Nash equilibrium on (s,u,x,t) when $b = 2$.

We now consider how this model of competition might both explain the outcomes of natural games and inform task designers on how to elicit better behavior from biased agents. For a natural game, consider the classic example of two companies competing to expand into a new market. Both companies want to launch a similar product, and are thus considering the same task graph G. The companies are also present biased, since shareholders often prefer immediate profit maximization/loss minimization over long term optimal behavior. The first company to enter the market gains an insurmountable advantage, represented by reward r. If the companies both enter the market at the same time, they split the market share, each getting reward $r/2$. This arrangement can be modeled within our framework, and the competition between the companies should lead them to play a set of equilibrium strategies.

For a designed game, consider the problem of encouraging students to submit final projects before they are due. The instructor sets a deadline near the end of finals week to give students flexibility to complete the project when it best fits their schedule. The instructor also knows that (1) students tend to procrastinate and (2) trying to complete the final project in a few days is much more challenging than spreading it out. They would like to convince students to work on and possibly submit their assignments early, *without* changing the deadline (to allow flexibility for the students whom it suits best). One possible solution would be to give a small amount of extra credit to the first submission. How might they set this reward to encourage early submissions?

In both these examples, the intuition is that competition will alleviate the harms of present bias by driving agents towards optimal behavior.

1.3 Summary of Results

We have introduced a model of competition for completing tasks along some graph. We warm up by analyzing these games absent present bias. Namely, we classify all Nash equilibria for an arbitrary task graph with unbiased agents, by first defining and eliminating all dominated paths.

We then analyze the model where agents have equal present bias. We show that a very small reward induces a Nash equilibrium on the optimal path, for any graph with a *dominant* path. This is a substantial improvement over the exponential worst case cost ratio experienced without competition. We then discuss how time-inconsistency defeats the intuition that higher rewards cause agents to prefer quicker paths. Despite this complication, we describe an algorithm that, given arbitrary graph G and path Q, determines the minimum reward needed to get a Nash equilibrium on Q, if possible.

Finally, we add an element of *bias uncertainty* to the model, by drawing agents' biases iid from distribution F and, for the n-fan, describe the relationship between F and the reward required for a Bayes-Nash equilibrium on the optimal path. For a wide range of distributions, we find small rewards suffice to ensure that agents behave optimally (with high probability) in equilibrium. For the stronger goal of ensuring a constant expected cost ratio, it suffices to offer reward linear in n when F is not heavy-tailed; competition thus helps here as well.

2 Nash Equilibria with Unbiased Competitors

To build intuition, we first describe the Nash equilibria of these games when agents have no present bias. We also pinpoint where the analysis will change with the introduction of bias. Notice that each path P in the graph is a strategy, with payoffs either $u_w = r - c(P)$, $u_t = r/2 - c(P)$ or $u_l = -c(P)$, depending on whether the agent wins, ties or loses, respectively. (These in turn depend on the path taken by the opponent.) We first rule out dominated paths. Notice that if $u_l(P) \geq u_w(P')$, path P' is dominated by path P, regardless of the path taken by the opponent. Also, if $u_w(P) \geq u_w(P')$ and $|P| \leq |P'|$ (where $|P|$ is the number of edges in P), then P' is dominated. Therefore, for any length k, a single cheapest path of length k will (weakly) dominate all other paths of length k.

For a given graph G, let $P_1, \dots, P_n$ be a minimal set of non-dominated paths, where $|P_i| < |P_{i+1}|$ for each $1 \leq i < n$. Thus, P_1 is the *quickest* path, the remaining path of minimum length. Summarizing what we know about these paths:

1. *Winning is better than losing*: for any pair of paths (P_i, P_j), we know that $u_w(P_i) > u_l(P_j)$. Thus, in particular, $c(P_1) - c(P_n) \leq r$.

2. *Longer paths are more rewarding*: That is, $c(P_i) > c(P_{i+1})$ for each i. Otherwise, P_{i+1} would be dominated since its length is greater. Therefore, in particular, P_n is the *cheapest* path, i.e., the lowest cost/weight path from s to t.

We're interested in characterizing, across all possible task graphs, the pure Nash equilibria, restricting attention to paths in $P_1, \ldots, P_n$.

Proposition 1. *Let G be an arbitrary task graph. As above, let $P_1, \ldots, P_n$ be a minimal set of (non-dominated) paths ordered so P_1 is the quickest and P_n the cheapest. Suppose $n \geq 3$. Then, path P_i, where $i > 1$, is a symmetric Nash equilibrium if and only if $c(P_{i-1}) - c(P_i) \geq r/2$. P_1 is a symmetric Nash if and only if $c(P_1) - c(P_n) \leq r/2$. There are no other pure Nash equilibria. Therefore, there can be either 0, 1 or 2 pure Nash equilibria.*

If $n = 2$, there is an additional asymmetric pure Nash equilibrium where one player plays the quickest path P_1 and the other plays the cheapest path P_2 if $c(P_1) - c(P_2) = r/2$.

The proof can be found in the full version. We next turn our attention to the biased version of this problem. In the unbiased case, we could take a "global" view of the graph, and think about paths purely in terms of their overall length and cost. But when agents are biased, the actual structure of the path is very important; time-inconsistency means that agents look at paths *locally*, not globally. It is thus very difficult to cleanly rule out dominated paths – even paths with exponentially high cost may be taken, as we see next.

3 Nash Equilibria to Elicit Optimal Behavior from Biased Agents

We assume that the agents are both naive, present biased agents, with shared bias parameter b.[1] Our high-level goal is to show that competition convinces biased agents to take cheap paths, as unbiased agents do without competition. To this end, we show that a small amount of reward creates a Nash equilibrium on the cheapest path, for all graphs which have a *dominant* path – a cheapest path that is also the *uniquely* quickest path.

3.1 Graphs with an Unbiased Dominant Strategy

To focus exclusively on the irrationality of present bias rather than the optimization problem of choosing between cheap, long paths and short, expensive paths, we focus on graphs with a *dominant path* – a cheapest path[2] that is also the

[1] The homogeneity of the agents is not particularly important to our results in this section. If the agents have different bias parameters, our results go through by setting b equal to the larger of the two biases.

[2] There may be other cheapest paths which are longer.

uniquely quickest path. An example of such a graph is the n-fan. In this setting, the problem is trivial for unbiased agents; simply take this dominant path. But for biased agents, the problem is still interesting; as the n-fan shows, they may take paths that are exponentially more costly than the dominant path. However, we prove that a small amount of competition and reward suffices to ensure the existence of a Nash equilibrium where both agents take the dominant path.

Theorem 1. *Suppose G is a task graph that has a* dominant *path, O. Then, a reward of $r \geq 2b \cdot \max_{e \in O} c(e)$ guarantees a Nash equilibrium on O, for two agents with bias b.*

Proof. Assume that A_2 takes O. Recall that a biased agent perceives the remaining traversal cost of going to v from t as $C_n(u, v) = b \cdot c(u, v) + \min_{P_v} c(P_v) - R_{A_2}(P_{s \to u, v \to t})$. We know that for any vertex v^* on the dominant path, the path that minimizes the second term is just the fragment of the dominant path from $v^* \to t$ (it is both the quickest and cheapest way to get from v^* to t). Further, any deviation from the dominant path results in no reward. So, for any v not on the dominant path, the path that minimizes the second term is again the cheapest path from $v \to t$. Thus, the cost equation simplifies to $C_n(u, v) = b \cdot c(u, v) + d(v) - r/2 \cdot \mathbf{1}\{D\}$, where $d(v)$, the *distance* from $v \to t$, denotes the cost of the cheapest path from v to t (ignoring rewards) and $\mathbf{1}\{D\}$ is simply an indicator variable that's 1 if the agent has not deviated from the dominant path.

Now, let $O = (s = v_0^*, v_1^*, v_2^*, \ldots, t = v_l^*)$ be the dominant path and suppose A_2 takes this path. In order for A_1 to choose O, we require, for all i:

$$\begin{aligned} & S(v_i^*) = v_{i+1}^* \\ \iff & v_{i+1}^* = \underset{v:(v_i^*, v) \in E}{\operatorname{argmin}} \; b \cdot c(v_i^*, v) + d(v) - r/2 \cdot \mathbf{1}\{D\} \\ \iff & \forall v : (v_i^*, v) \in E, bc(v_i^*, v) + d(v) \geq bc(v_i^*, v_{i+1}^*) + d(v_{i+1}^*) - r/2 \end{aligned}$$

Now, let i be arbitrary, let $v \neq v_i^*$ be an arbitrary neighbor of v_i^*, and for ease of notation, let $c = c(v_i^*, v), c^* = c(v_i^*, v_{i+1}^*), d = d(v)$, and $d^* = d(v_{i+1}^*)$. Then, we get the following bound on the reward: $r/2 \geq b(c^* - c) + d^* - d$. To get a rough sufficient bound, notice that $c + d \geq c^* + d^*$, since O is the cheapest path. This implies that $bc^* > b(c^* - c) + d^* - d$. Thus, it suffices to set $r \geq 2bc^*$ in order to ensure $S(v_i^*) = v_{i+1}^*$. Repeating this argument for all i, we see that a sufficient reward is $r = 2b \cdot \max_{e \in O} c(e)$. □

One might object to our claim that $r = 2b \max_{e \in O} c(e)$ is "small". To calibrate our expectations, notice that we can view this problem as an agent trying to pick between several options (i.e. paths), each with their own reward and cost structure. We want to convince the agent to pick one particular option – namely, the cheapest path. But it would be unreasonable to expect that a reward significantly smaller than the cost of *any* option would sway the agent's decision. Our theorem above shows that a reward that is at most proportional to the *cheapest* option suffices; and in many cases the reward is only a fraction of the cost of

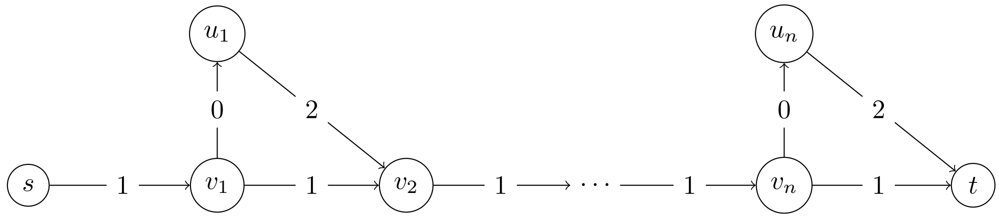

Fig. 4: A graph with many suboptimal deviations. For an agent with bias $b > 2$, a designer with access to only edge rewards must spend $O(n)$ total reward for optimal behavior ($b - 2$ on each (v_i, v_{i+1}) edge). In our competitive setting, only $2(b - 2)$ total reward is required.

the optimal path (e.g. when the optimal path has balanced cost among many edges).

For a point of comparison, even internal edge rewards (which required the same reward budget as competitive rewards for the n-fan) can require $O(n)$ times as much reward in some instances. To see the intuition, notice that internal edge rewards must be applied at every step where the agent might want to deviate. The agent also immediately "consumes" this reward; it doesn't impact his future decision making. However, the competitive reward is "at stake" whenever the agent considers deviating; this reward can sway the agent's behavior without being immediately consumed. For a concrete example, see Figure 4.

3.2 Increasing the Number of Competitors

A very natural extension to this model would involve more than 2 agents competing. The winner takes all, and ties are split evenly among those who tied. However, this modification doesn't change much *when trying to get a Nash equilibrium on the dominant path*. The only change is that if m agents are competing, the reward needed is $O(m)$, as a single agent will get a $1/m$ fraction of the reward in a symmetric equilibrium. This is true because there is no way for any agent to beat the dominant path, and claim the entire $O(m)$ reward for themselves. So if the reward is scaled appropriately (i.e. in Theorem 1 set $r \geq mb \cdot \max_{e \in O} c(e)$), we will still guarantee a Nash equilibrium. Put another way, the *per-agent* reward needed for a Nash equilibrium on the dominant path does not change as the number of competitors varies.

4 General Nash Equilibria

In this section, we describe a polynomial time algorithm that, given an arbitrary graph G, path Q and bias b, determines if Q can be made a Nash equilibrium, and if so, the minimum required reward to do so. Finding and using this minimum required reward will generally cost much less than the bound given by Theorem 1. Moreover, this algorithm does not assume the existence of a dominant path.

We start by describing how time-inconsistency defeats the intuition that higher rewards cause agents to prefer quicker paths. We then present a very high level overview of how to compute the minimum reward that results in Q being a symmetric Nash equilibrium.

4.1 Higher Rewards Need Not Encourage Quicker Paths

The proof of Theorem 1 suggests the following algorithm for this problem. Start with a reward of 0, and step along each vertex $u \in Q$, increasing the reward by just enough to ensure A_1 stays on Q for one more step (assuming A_2 is taking Q). However, if A_1 wants to deviate onto a quicker/tied path at any point, return $\perp$; decreasing the reward would cause them to deviate earlier, and, intuitively, it seems that increasing the reward could not cause them to switch back to Q from the quicker/tied path. After one pass, simply pass through again to ensure that the final reward doesn't cause A_1 to deviate early on. The following lemma, which we prove in the full version, shows that this algorithm is tractable, by showing that we can compute the minimum reward required for A_1 to stay on Q at any step (and determine whether A_1 wants to deviate onto a quicker path).

Lemma 1. *Assuming that A_2 takes path Q, A_1 can efficiently compute $\min_{P_v} c(P_v) - R_{A_2}(P_{s\to u, v\to t})$ by considering the cheapest path (from $v \to t$), the cheapest path where A_1 ties A_2, and the cheapest path where A_1 beats A_2. (Some of these paths may coincide, and at least one must exist).*

Unfortunately, while tractable, the approach described above does not yield a correct algorithm. This is because it relies implicitly on the following two properties, which formalize the intuition that increasing the reward causes agents to favor quicker paths.

Property 1 If a reward r guarantees a Nash equilibrium on some path Q, any reward $r' > r$ will either (a), still result in a Nash equilibrium on Q, or (b), cause an agent to deviate to a *quicker* path Q'.

Property 2 If A_1 deviates from Q onto a quicker/tied path for some reward r, increasing the reward will not cause them to follow Q

Both properties are intuitive – if we increase the reward, that should motivate the agent to take a path that beats their opponent. And vice versa – increasing the reward shouldn't cause them to shift onto a slower path or shift between equal length paths. But surprisingly, both properties are false, as Figure 5 demonstrates.

For Property 1, consider the graph in Figure 5(a) and define paths $Q = (s, q_1, q_2, t)$, $V = (s, v_1, v_2, v_3, t)$, and $X = (s, v_1, t)$. Suppose both agents have bias 10 and that A_2 takes Q. Then a reward of 1 guarantees that A_1 takes Q, as the optimal path from $v_1 \to t$ would follow V. However, if $r = 300$, the optimal path from $v_1 \to t$ follows X. So, A_1 goes to v_1, intending to beat A_2. But at v_1, the perceived cost of (v_1, t) is actually 1000, and so the agent prefers to take

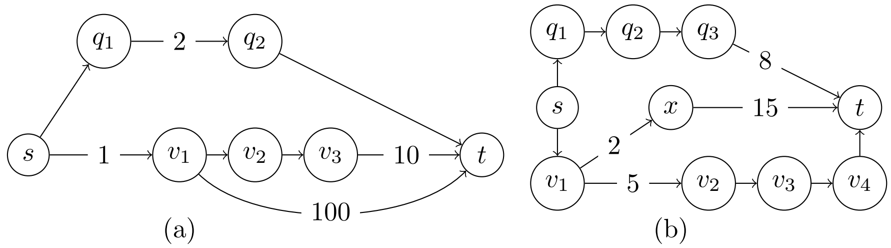

Fig. 5: Graphs which do not exhibit the two expected properties. Unlabeled edges have cost 0.

path V. Thus, increasing the reward from 1 to 300 causes the agent to deviate from Q onto a slower path!

For Property 2, consider the graph in Figure 5(b) and define paths Q, V, and X in the obvious manner. Again, suppose both agents have bias 10 and that A_2 takes Q. Then, with a reward of 10, A_1 to stick to Q as well. But if $r = 2$, the optimal path from $v_1 \to t$ follows V and thus loses, which is not as meaningful. So A_1 goes to v_1, intending to follow V. But there, with $b = 10$, deviating to X is more attractive than remaining on V, and thus the agent takes X. So, although A_1 deviates from Q to a quicker path for reward 2, they remain on Q with reward 10.

To summarize, Property 1 fails because present biased agents can take slower paths than they planned and Property 2 fails because present biased agents can take quicker paths than they planned. In other words, while higher rewards do *tempt* agents to take quicker paths, and lower rewards tempt agents towards cheaper paths, their time inconsistency may make them do the opposite.

4.2 A Description of the Algorithm

We now describe, at a very high level, how to efficiently find the set of rewards which induce a symmetric Nash equilibrium on a path Q. The algorithm narrows down the set of *feasible* rewards (rewards that ensure that Q is a Nash equilibria) by computing the set of rewards that ensure that A_1 takes (u, v) for every $(u, v) \in Q$. The key idea is that we can efficiently compute all r that ensures that A_1 prefers (u, v) over (u, v') by splitting into cases based on whether the optimal paths from $v \to t$ and $v' \to t$ involve winning, tying, or losing. From this algorithm, we get the following theorem:

Theorem 2. *There exists a polynomial time algorithm that returns the minimum r that ensures that Q is a Nash equilibrium, or $\perp$ if no such r exists.*

We prove this theorem and fully define the algorithm in the full version.

5 Extending the Model with Bias Uncertainty and Multiple Competitors

One of the shortcomings of the prior results is that agents are assumed to have publicly known, identical biases, which seems unrealistic. We therefore add the agents' uncertainty about their competitors bias to the model. The agents' biases are now represented by random variables B_1 and B_2 drawn iid from distribution F, which is publicly known to both the agents and the designer. b_1 and b_2 correspond to the realizations of these random variables. Our goal is now to construct, as cheaply as possible, Bayes-Nash equilibria (BNE) where agents behave optimally with high probability. In this setting, the cost equation becomes $C_n(u,v) = b \cdot c(u,v) + \min_{P_v} c(P_v) - \mathbb{E}_{A_2}[R_{A_2}(P_{s\to u, v\to t})]$, where the expectation is over A_2's choice of paths.

In this section, we provide a closed form BNE for the n-fan. We start with the case of two agents and then briefly consider m competing agents. Since we are searching for BNE, we assume that the agents know their competitor's strategy.

5.1 Bayes-Nash Equilibria on the n-fan

As before, let P_i represent the path that includes edge (v_i, t), and let P_0 represent the optimal path. Then, the following strategy is a Bayes-Nash equilibrium.

Theorem 3. *Let G be an n-fan with reward r and suppose B_1, B_2 are drawn from distribution B with CDF F. Let p be the solution to $F(\frac{rp}{2} + c) = p$. If $p > \frac{1}{c^{n-1}+1}$, then the following strategy is a Bayes-Nash equilibrium:*

$$P(b) = \begin{cases} \text{take } P_0, & b \leq \frac{rp}{2} + c \\ \text{take } P_n, & \text{otherwise} \end{cases}$$

In this equilibrium, for either agent, the probability that they take P_0 is p. So the expected cost ratio will be $p + (1-p)c^n$.

We prove this in the full version. Notice that while the trivial solution $p = 0$ satisfies $F(\frac{rp}{2} + c) = p$, this is not above $\frac{1}{c^{n-1}+1}$, so the trivial solution is not relevant for finding Bayes-Nash equilibria. One might wonder if there's a straightforward generalization of this BNE to other graphs with a dominant path, as in the case without bias uncertainty. In the full version, we discuss challenges that we encountered trying to do this.

We now use the theorem to understand how much reward is required for optimal behavior with high probability, or a low expected cost ratio (which is a much stronger requirement). Since the expected cost is $p + c^n(1-p)$, in order for this to be low, $1-p$ has to be close to $1/c^n$. Plugging this in to the CDF, we see that for this to happen, we must have

$$F\left(\frac{r}{2}\left(1 - \frac{1}{c^n}\right) + c\right) = 1 - \frac{1}{c^n}$$

which essentially requires that exponentially little probability mass (in n) remains after $r/2$ distance from c. For an exponential distribution, this requires r to be linear in n, and with a heavier tailed distribution like the Equal Revenue distribution, this requires r to be exponential in n. But we may be content with simply guaranteeing optimal behavior with high probability. In that case, so long as r is increasing in n, the agents will take the optimal path with high probability for at least the equal revenue, exponential, and uniform distributions. We more precisely explore the probability of optimal behavior and the cost ratios for these distributions in the full version.

5.2 Increasing Number of Competitors

We saw earlier that increasing the number of competitors doesn't change the *per-agent* reward needed for optimal behavior. But one might hope that in Bayesian settings such as bias uncertainty, increasing the number of agents significantly decreases the per-agent reward needed to encourage optimal behavior – in particular, as the number of agents increases, the probability of *some* agent having a very low bias increases. In the full version, we provide evidence against this belief. We first show that the BNE in Theorem 3 can be tweaked slightly to remain a BNE with a variable number of agents. We then consider the equal revenue distribution, which required an exponentially high reward to get a low expected cost ratio with just two competitors. But we show that even as number of competing agents goes to ∞, this relationship between the reward and the probability of optimal behavior doesn't significantly change. We conjecture that, in general, increasing the number of agents does not significantly decrease the per-agent reward required for optimal behavior.

6 Conclusion

We studied the impact of competition on present bias, showing that in many settings where naive agents can experience exponentially high cost ratio, a small amount of competition drives agents to optimal behavior. This paper is a first step towards painting a more optimistic picture than much of the work surrounding present bias. Our results highlight why, in naturally competitive settings, otherwise biased agents might behave optimally. Further, task/mechanism designers can use our results to directly alleviate the harms of present bias. This competitive model might be a more natural model than other motivation schemes, such as internal edge rewards, and is able to more cheaply ensure optimal behavior. Our work also leaves open many exciting questions.

First, with bias uncertainty, we only obtain concrete results on the n-fan. So one obvious direction is to determine which graphs have Bayes-Nash equilibria on the optimal path, and what these equilibria look like. Second, we explore two "dimensions" of competition – the amount of reward and the number of competitors, finding that the latter is unlikely to be significant. Another interesting goal is thus to explore new dimensions of competition.

Lastly, we could extend our work beyond cost ratios, moving to the model where agents can abandon their path at any point. For one, this move would allow us to integrate results on *sunk-cost* bias, represented as an intrinsic cost for abandoning a task that's proportional to the amount of effort expended. Previous work [12] has shown that agents who are sophisticated with regard to their present-bias, but naive with respect to their sunk cost bias can experience exponentially high cost before abandoning their traversal (this is especially interesting because sophisticated agents without sunk cost bias behave nearly optimally). Can competition alleviate this exponential worst case? There are also interesting computational questions in this model. For instance, given a fixed reward budget r, is it possible to determine in polynomial time if one can induce an equilibrium where both agents traverse the graph? Such problems are NP-hard for other reward models, but may be tractable with competition. Overall, the abandonment setting has several interesting interactions with competition that we have not explored.

Bibliography

[1] Albers, S., Kraft, D.: On the value of penalties in time-inconsistent planning. arXiv preprint arXiv:1702.01677 (2017)

[2] Albers, S., Kraft, D.: Motivating time-inconsistent agents: A computational approach. Theory of computing systems **63**(3), 466–487 (2019)

[3] Ariely, D., Jones, S.: Predictably irrational. Harper Audio New York, NY (2008)

[4] Arkes, H.R., Blumer, C.: The psychology of sunk cost. Organizational behavior and human decision processes **35**(1), 124–140 (1985)

[5] DellaVigna, S.: Psychology and economics: Evidence from the field. Journal of Economic literature **47**(2), 315–72 (2009)

[6] DellaVigna, S., Malmendier, U.: Paying not to go to the gym. american economic Review **96**(3), 694–719 (2006)

[7] Frederick, S., Loewenstein, G., O'donoghue, T.: Time discounting and time preference: A critical review. Journal of economic literature **40**(2), 351–401 (2002)

[8] Gravin, N., Immorlica, N., Lucier, B., Pountourakis, E.: Procrastination with variable present bias. arXiv preprint arXiv:1606.03062 (2016)

[9] Kahneman, D., Tversky, A.: Prospect theory: An analysis of decision under risk. Econometrica **47**(2), 263–292 (1979)

[10] Kleinberg, J., Oren, S.: Time-inconsistent planning: a computational problem in behavioral economics. In: Proceedings of the fifteenth ACM conference on Economics and computation. pp. 547–564 (2014)

[11] Kleinberg, J., Oren, S., Raghavan, M.: Planning problems for sophisticated agents with present bias. In: Proceedings of the 2016 ACM Conference on Economics and Computation. pp. 343–360 (2016)

[12] Kleinberg, J., Oren, S., Raghavan, M.: Planning with multiple biases. In: Proceedings of the 2017 ACM Conference on Economics and Computation. pp. 567–584 (2017)

[13] Ma, H., Meir, R., Parkes, D.C., Wu-Yan, E.: Penalty bidding mechanisms for allocating resources and overcoming present bias. arXiv preprint arXiv:1906.09713 (2019)
[14] Oren, S., Soker, D.: Principal-agent problems with present-biased agents. In: International Symposium on Algorithmic Game Theory. pp. 237–251. Springer (2019)
[15] Tang, P., Teng, Y., Wang, Z., Xiao, S., Xu, Y.: Computational issues in time-inconsistent planning. In: Thirty-First AAAI Conference on Artificial Intelligence (2017)
[16] Yan, W., Yong, J.: Time-inconsistent optimal control problems and related issues. In: Modeling, Stochastic Control, Optimization, and Applications, pp. 533–569. Springer (2019)