Time-Inconsistent Planning: A Computational Problem in Behavioral Economics

By Jon Kleinberg and Sigal Oren

Abstract

In many settings, people exhibit behavior that is inconsistent across time—we allocate a block of time to get work done and then procrastinate, or put effort into a project and then later fail to complete it. An active line of research in behavioral economics and related fields has developed and analyzed models for this type of time-inconsistent behavior.

Here we propose a graph-theoretic model of tasks and goals, in which dependencies among actions are represented by a directed graph, and a time-inconsistent agent constructs a path through this graph. We first show how instances of this path-finding problem on different input graphs can reconstruct a wide range of qualitative phenomena observed in the literature on time-inconsistency, including procrastination, abandonment of long-range tasks, and the benefits of reduced sets of choices. We then explore a set of analyses that quantify over the set of all graphs; among other results, we find that in any graph, there can be only polynomially many distinct forms of time-inconsistent behavior; and any graph in which a time-inconsistent agent incurs significantly more cost than an optimal agent must contain a large “procrastination” structure as a minor. Finally, we use this graph-theoretic model to explore ways in which tasks can be designed to motivate agents to reach designated goals.

1. INTRODUCTION

A fundamental issue in behavioral economics—and in the modeling of individual decision-making more generally—is to understand the effects of decisions that are inconsistent over time. Examples of such inconsistency are widespread in everyday life: we make plans for completing a task but then procrastinate; we put work into getting a project partially done but then abandon it; we pay for gym memberships but then fail to make use of them. In addition to analyzing and modeling these effects, there has been increasing interest in incorporating them into the design of policies and incentives in domains that range from health to personal finance.

These types of situations have a recurring structure: a person makes a plan at a given point in time for something they will do in the future (finishing homework, exercising, paying off a loan), but at a later point in time they fail to follow through on the plan. Sometimes this failure is the result of unforeseen circumstances that render the plan invalid—a person might join a gym but then break their leg and be unable to exercise—but in many cases the plan is abandoned even though the circumstances are essentially the same as

they were at the moment the plan was made. This presents a challenge to any model of planning based on optimizing a utility function that is consistent over time: in an optimization framework, the plan must have been an optimal choice at the outset, but later it was optimal to abandon it. A line of work in the economics literature has thus investigated the properties of planning with objective functions that vary over time in certain natural and structured ways.

A basic example and model

To introduce these models, it is useful to briefly describe an example due to George Akerlof,¹ with the technical details adapted slightly for the discussion here. (The story will be familiar to readers who know Akerlof’s paper; we cover it in some detail because it will motivate a useful and recurring construction later in the work.) Imagine a decision-making agent—Akerlof himself, in his story—who needs to ship a package sometime during one of the next n days, labeled $t = 1, 2, \dots, n$, and must decide on which day t to do so. Each day that the package has not been sent results in a fixed cost of 1 (per day), due to the lack of use of the package’s contents; this means a cost of t if it is shipped on day t . (For simplicity, we will disregard the time the package spends in transit, which is a constant additional cost regardless of when it is shipped.) Also, shipping the package is an elaborate operation that will result in one-time cost of $c > 1$, due to the amount of time involved in getting it sent out. The package must be shipped during one of the n specified days.

What is the optimal plan for shipping the package? Clearly the cost c will be incurred exactly once regardless of the day on which it is shipped, and there will also be a cost of t if it is shipped on day t . Thus we are minimizing $c + t$ subject to $1 \leq t \leq n$; the cost is clearly minimized by setting $t = 1$. In other words, the agent should ship the package right away.

But in Akerlof’s story, he did something that should be familiar from one’s own everyday experience: he procrastinated. Although there were no unexpected changes to the trade-offs involved in shipping the package, when each new day arrived there seemed to be other things that were more crucial than sending it out that day, and so each day he resolved that he would instead send it tomorrow. The result was that the package was not sent out until the end of the

The original version of this paper was published in the *Proceedings of the 15th ACM Conference on Economics and Computation* (Palo Alto, CA, June 8–12, 2014), 547–564.

time period. (In fact, he sent it a few months into the time period once something unexpected happened to change the cost structure—a friend offered to send it as part of a larger shipment—though this wrinkle is not crucial for the story.)

There is a natural way to model an agent's decision to procrastinate, using the notion of *present bias*—the tendency to view costs and benefits that are incurred at the present moment to be more salient than those incurred in the future. In particular, suppose that for a constant $b > 1$, costs that one must incur in the current time period are increased by a factor of b in one's evaluation.^a Then in Akerlof's example, when the agent on day t is considering the decision to send the package, the cost of sending it on day t is $bc + t$, while the cost of sending it on day $t + 1$ is $c + t + 1$. The difference between these two costs is $(b - 1)c - 1$, and so if $(b - 1)c > 1$, the agent will decide on each day t that the optimal plan is to wait until day $t + 1$; things will continue this way until day n , when waiting is no longer an option and the package must be sent.

Quasi-hyperbolic discounting

Building on considerations such as those above, and others in earlier work in economics,^{16,19} a significant amount of work has developed around a model of time-inconsistency known as *quasi-hyperbolic discounting*.¹² In this model, parametrized by quantities $\beta, \delta \leq 1$, a cost or reward of value c that will be realized at a point $t \geq 1$ time units into the future is evaluated as having a present value of $\beta\delta^t c$. (In other words, values at time t are discounted by a factor of $\beta\delta^t$.) With $\beta = 1$ this is the standard functional form for exponential discounting, but when $\beta < 1$ the function captures present bias as well: values in the present time period are scaled up by β^{-1} relative to all other periods. (In what follows, we will consistently use b to denote β^{-1} .)

Research on this (β, δ) -model of discounting has been extensive, and has proceeded in a wide variety of directions; see Ref. Frederick et al.⁷ for a review. To keep our analysis clearly delineated in scope, we make certain decisions at the outset relative to the full range of possible research questions: we focus on a model of agents who are *naïve*, in that they do not take their own time-inconsistency into account when planning; we do not attempt to derive the (β, δ) -model from more primitive assumptions but rather take it as a self-contained description of the agent's observed behavior; and we discuss the case of $\delta = 1$ so as to focus attention on the present-bias parameter β . Note that the initial Akerlof example has all these properties; it is essentially described in terms of the (β, δ) -model with an agent who is naïve about his own time-inconsistency, with $\delta = 1$, and with $\beta = b^{-1}$ (using the parameter b from that discussion).

Our starting point in this paper is to think about some of the qualitative predictions of the (β, δ) -model, and how to analyze them in a unified framework. In particular, research in behavioral economics has shown how agents making plans in this model can exhibit the following behaviors.

1. *Procrastination*, as discussed above.
2. *Abandonment* of long-range tasks, in which a person starts on a multi-stage project but abandons it in the middle, even though the costs and benefits of the project have remained essentially unchanged.^{15,b}
3. The benefits of *choice reduction*, in which reducing the set of options available to an agent can actually help them reach a goal more efficiently.^{9, 14} A canonical example is the way in which imposing a deadline can help people complete a task that might not get finished in the absence of a deadline.⁴

These consequences of time-inconsistency, as well as a number of others, have in general each required their own separate and sometimes quite intricate modeling efforts. It is natural to ask whether there might instead be a single framework for representing tasks and goals in which all of these effects could instead emerge “mechanically,” each just as a different instance of the same generic computational problem. With such a framework, it would become possible to search for worst-case guarantees, by quantifying over all instances, and to talk about designing or modifying given task structures to induce certain desired behaviors.

The present work: A graph-theoretic model

Here we propose such a framework, using a graph-theoretic formulation. We consider an agent with present-bias parameter β who must construct a path in a directed acyclic graph G with edge costs, from a designated start node s to a designated target node t . We assume without loss of generality that all the edges of G lie on some $s - t$ path. We will call such a structure a *task graph*. Informally, the nodes of the task graph represent states of intermediate progress toward the goal t , and the edges represent transitions between them. Directed graphs have been shown to have considerable expressive power for planning problems in the artificial intelligence literature;¹⁸ this provides evidence for the robustness of a graph-based approach in representing these types of decision environments. Our concerns in this work, however, are quite distinct from the set of graph-based planning problems in artificial intelligence, since our aim is to study the consequences of time-inconsistency in these domains.

A sample instance of this problem is depicted in Figure 1, with the costs drawn on the edges. When the agent is standing at a node v , it determines the minimum cost of a path from v to t , but it does so using its present-biased evaluation of costs: the cost of the first edge on the path (starting from v) is evaluated according to the true cost, and all subsequent edges have their costs reduced by β . If the agent chooses path P , it follows just the first edge (v, w) of P , and then it re-evaluates which path to follow using this same present-biased evaluation but now from w . In this way, the agent iteratively constructs a path from s to t .

^a Note that there is no time-discounting in this example, so the factor of b is only applied to the present time period, while all future time periods are treated equally. We will return to the issue of discounting shortly.

^b For purposes of our discussion, we distinguish abandonment of a task from the type of procrastination exhibited by Akerlof's example, in which the task is eventually finished, but at a much higher cost due to the effect of procrastination.

Figure 1. A present-biased agent must choose a path from s to t .

In the next section we will show how our graph-theoretic model easily captures time-inconsistency phenomena including procrastination, abandonment, and choice reduction. But to make the definitions concrete, it is useful to work through the agent's computation on the graph depicted in Figure 1. As shown in Figure 1, an agent that has a present-bias parameter of $\beta = 1/2$ needs to go from s to t . From s , the agent evaluates the path $s-a-b-t$ as having cost $16 + 2\beta + 2\beta = 18$, the path $s-c-d-t$ as having cost $8 + 8\beta + 8\beta = 16$, and the path $s-c-e-t$ as having cost $8 + 2\beta + 16\beta = 17$. Thus the agent traverses the edge (s, c) and ends up at c . From c , the agent now evaluates the path $c-d-t$ as having cost $8 + 8\beta = 12$ and the path $c-e-t$ as having cost $2 + 16\beta = 10$, and so the agent traverses the edge (c, e) and then (having no further choices) continues on the edge (e, t) .

This example illustrates a few points. First, when the agent set out on the edge (s, c) , it was intending to next follow the edge (c, d) , but when it got to c , it changed its mind and followed the edge (c, e) . A time-consistent agent (with $\beta = 1$), in contrast, would never do this; the path it decides to take starting at s is the path it will continue to follow all the way to t . Second, we are interested in whether the agent minimizes the cost of traveling from s to t according to the real costs, not according to its evaluation of the costs, and in this regard it fails to do so; the shortest path is $s-a-b-t$, with a cost of 20, while the agent incurs a cost of 26.

Overview of results

Our graph-theoretic framework makes it possible to reason about time-inconsistency effects that arise in very different settings, provided simply that the underlying decisions faced by the agent can be modeled as the search for a path through a graph-structured sequence of options. And perhaps more importantly, since it is tractable to ask questions that quantify over all possible graphs, we can cleanly compare different scenarios, and search for the best or worst possible structures relative to specific objectives. This is difficult to do without an underlying combinatorial structure. For example, suppose we were inspired by Akerlof's example to try identifying the scenario in which time-inconsistency leads to the greatest waste of effort. *A priori*, it is not clear how to formalize the search over all possible "scenarios." But as we will see, this is precisely something we can do if we simply ask for the graph in which time-inconsistency produces the greatest ratio between the cost of the path traversed and cost of the optimal path.

Moreover, with this framework in place, it becomes easier to express formal questions about design for these contexts: if as a designer of a complex task we are able to specify the underlying graph structure, which graphs will lead time-inconsistent agents to reach the goal efficiently?

Our core questions are based on quantifying the inefficiency from time-inconsistent behavior, designing task structures to reduce this inefficiency, and comparing the behavior of agents with different levels of time-inconsistency. Specifically, we ask:

1. In which graph structures does time-inconsistent planning have the potential to cause the greatest waste of effort relative to optimal planning?
2. How do agents with different levels of present bias (encoded as different values of β) follow combinatorially different paths through a graph toward the same goal?
3. Can we increase an agent's efficiency in reaching a goal by deleting nodes and/or edges from the underlying graph, thus reducing the number of options available?

In what follows, we address these questions in turn. For the first question, we consider n -node graphs and ask how large the *cost ratio* can be between the path followed by a present-biased agent and the path of minimum total cost. Since deviations from the minimum-cost plan due to present bias are sometimes viewed as a form of "irrational" behavior, this cost ratio effectively serves as a "price of irrationality" for our system. We give a characterization of the worst-case graphs in terms of *graph minors*; this enables us to show, roughly speaking, that any instance with sufficiently high cost ratio must contain a large instance of the Akerlof example embedded inside it.

For the second question, we consider the possible paths followed by agents with different present-bias parameters β . As we sweep β over the interval $[0, 1]$, we have a type of *parametric* path problem, where the choice of path is governed by a continuous parameter (β in this case). We show that in any instance, the number of distinct paths is bounded by a polynomial function of n , which forms an interesting contrast with canonical formulations of the parametric shortest-path problem, in which the number of distinct paths can be super polynomial in n .^{5,13}

Lastly, for the third question, we show how it is possible for agents to be more efficient when nodes and/or edges are deleted from the underlying graph; on the other hand, if we want to motivate an agent to follow a particular path P through the graph, it can be crucial to present the agent with a subgraph that includes not just P but also certain additional nodes and edges that do not belong to P . We give a graph-theoretic characterization of the possible subgraphs supporting efficient traversal.

Before turning to these questions, we first discuss the basic graph-theoretic problem in more detail, showing how instances of this problem capture the time-inconsistency phenomena discussed earlier in this section.

2. THE GRAPH-THEORETIC MODEL

In order to argue that our graph-theoretic model captures a variety of phenomena that have been studied in connection with time-inconsistency, we present a sequence of examples to illustrate some of the different behaviors that the model exhibits. We note that the example as shown in Figure 1 already illustrates two simple points: that the path chosen by the agent can be suboptimal; and that even if the agent traverses an edge e with the intention of following a path P that begins with e , it may end up following a different path P' that also begins with e .

For an edge e in G , let $c(e)$ denote the cost of e ; and for a path P in G , let $e_i(P)$ denote the i^{th} edge on P . In terms of this notation, the agent's decision is easy to specify: when standing at a node v , it chooses the path P that minimizes $c(e_1(P)) + \beta \sum_{i=2}^{\infty} c(e_i(P))$ over all P that run from v to t . It follows the first edge of P to a new node w , and then performs this computation again.

We begin by observing that Figure 2(a) represents a version of the Akerlof example from the introduction. (Recall that here we use b to denote β^{-1} .) Node t represents the state in which the agent has sent the package, and node v_i represents the state in which the agent has reached day i without sending the package. The agent has the option of going directly from node s to node t , and this is the shortest s - t path. But if $(b-1)c > b$, then the agent will instead go from s to v_1 , intending to complete the path s - v_1 - t in the next time step. At v_1 , however, the agent decides to go to v_2 , intending to complete the path v_1 - v_2 - t in the next time step. This process continues: the agent, following exactly the reasoning in the example from the introduction, is procrastinating and not going to t , and in the end its path goes all the way to the last node v_n ($n = 5$ in the figure) before finally taking an edge to t . (One minor change from the set-up in the introduction is that the present-bias effect is also applied to the per-day cost of 1 as well; this has no real effect on the underlying story.)

Figure 2. Path problems that exhibit procrastination, abandonment, and choice reduction.

Extending the model to include rewards

Thus far we can not talk about an agent who abandons its pursuit of the goal midway through, since our model requires the agent to construct a path that goes all the way to t . A simple extension of the model enables to consider such situations.

Suppose we place a reward of r at the target node t , which will be claimed if the agent reaches t . Standing at a node v , the agent now has an expanded set of options: it can follow an edge out of v as before, or it can quit taking steps, incurring no further cost but also not claiming the reward. The agent will choose the latter option precisely when either there is no v - t path, or when the minimum cost of a v - t path exceeds the value of the reward, evaluated in light of present bias: $c(e_1(P)) + \beta \sum_{i=2}^{\infty} c(e_i(P)) > \beta r$ for all v - t paths P . It is important to note a key feature of this evaluation: the reward is always discounted by β relative to the cost that is being incurred in the current period, even if the reward will be received right after this cost is incurred. (e.g., if the path P has a single edge, then the agent is comparing $c(e_1(P))$ to βr .)

In what follows, we will consider both these models: the former *fixed-goal model*, in which the agent must reach t and seeks to minimize its cost; and the latter *reward model* in which the agent trades off cost incurred against reward at t , and has the option of stopping partway to t . Aside from this distinction, both models share the remaining ingredients, based on traversing an s - t path in G .

It is easy to see that the reward model displays the phenomenon of *abandonment*, in which the agent spends some cost to try reaching t , but then subsequently gives up without receiving the reward. Consider for example a three-node path on nodes s , v_1 , and t , with an edge $c(s, v_1) = 1$ and $c(v_1, t) = 4$. If $\beta = 1/2$ and there is a reward of 7 at t , then the agent will traverse the edge (s, v_1) because it evaluates the total cost of the path at $1 + 4\beta = 3 < 7\beta = 3.5$. But once it reaches v_1 , it evaluates the cost of completing the path at $4 > 7\beta = 3.5$, and so it quits without reaching t .

An example involving choice reduction

It is useful to describe a more complex example that shows the modeling power of this shortest-path formalism, and also shows how we can use the model to analyze deadlines as a form of beneficial choice reduction. (As should be clear, with a time-consistent agent it can never help to reduce the set of choices; such a phenomenon requires some form of time-inconsistency.) First we describe the example in text, and then show how to represent it as a graph.

Imagine a student taking a three-week short course in which the required work is to complete two small projects by the end of the course. It is up to the student when to do the projects, as long as they are done by the end. The student incurs an effort cost of one from any week in which she does no projects (since even without projects there is still the lower-level effort of attending class), a cost of four from any week in which she does one project, and a cost of nine from any week in which she does both projects. Finally, the student receives a reward of 16 for completing the course, and she has a present-bias parameter of $\beta = 1/2$.

Figure 2(b) shows how to represent this scenario using a graph. Node v_{ij} corresponds to a state in which i weeks of the course are finished, and the student has completed j projects so far; we have $s = v_{00}$ and $t = v_{32}$. All edges go one column to the right, indicating that one week will elapse regardless; what is under the student's control is how many rows the edge will span. Horizontal edges have cost one, edges that descend one row have cost four, and edges that descend two rows in a single hop have cost nine. In this way, the graph precisely represents the story just described.

How does the student's construction of an s - t path work out? From s , she goes to v_{10} and then to v_{20} , intending to complete the path to t via the edge (v_{20}, t) . But at v_{20} , she evaluates the cost of the edge (v_{20}, t) as $9 > \beta r = 16/2 = 8$, and so she quits without reaching t . The story is thus a familiar one: the student plans to do both projects in the final week of the course, but when she reaches the final week, she concludes that it would be too costly and so she drops the course instead.

The instructor can prevent this from happening through a very simple intervention. If he requires that the first project be completed by the end of the second week of the course, this corresponds simply to deleting node v_{20} from the graph. With v_{20} gone, the path-finding problem changes: now the student starting at s decides to follow the path s - v_{10} - v_{21} - t , and at v_{10} and then v_{21} she continues to select this path, thereby reaching t . Thus, by reducing the set of options available to the student—and in particular, by imposing an intermediate deadline to enforce progress—the instructor is able to induce the student to complete the course.

There are many stories like this one about homework and deadlines, and our point is not to focus too closely on it in particular. Indeed, to return to one of the underpinnings of our graph-theoretic formalism, our point is in a sense the opposite: it is hard to reason about the space of possible “stories,” whereas it is much more tractable to think about the space of possible graphs. Thus by encoding the set of stories mechanically in the form of graphs, it becomes feasible to reason about them as a whole.

We have thus seen how a number of different time-inconsistency phenomena arise in simple instances of the path-finding problem. The full power of the model, however, lies in proving statements that quantify over all graphs; we begin this next.

3. THE COST RATIO: A CHARACTERIZATION VIA GRAPH MINORS

Our path-finding model naturally motivates a basic quantity of interest: the *cost ratio*, defined as the ratio between the cost of the path found by the agent and the cost of the shortest path. We work here within the fixed-goal version of the model, in which the agent is required to reach the goal t and the objective is to minimize the cost of the path used.

To fix notation for this discussion, given a directed acyclic graph G on n nodes with positive edge costs, we let $d(v, w)$ denote the cost of the shortest v - w path in G (using the true edge costs, not modified by present bias). Let $P_\beta(v, t)$ denote the the v - t path followed by an agent with present-bias β , and let $c_\beta(v, t)$ be the total cost of this path. The cost ratio can thus be written as $c_\beta(s, t)/d(s, t)$.

A bad example for the cost ratio

We first describe a simple construction showing that the cost ratio can be exponential in the number of nodes n . We then move on to the main result of this section, which is a characterization of the instances in which the cost ratio achieves this exponential lower bound.

Our construction is an adaptation of the Akerlof example from the introduction. We have a graph that consists of a directed path $s = v_0, v_1, v_2, \dots, v_n$, and with each v_i also linking directly to node t . (The case $n = 5$ is the graph in Figure 2(a).) With $b = \beta^{-1}$, we choose any $\mu < b$; we let the cost of the edge (v_j, t) be μ^j , and let the cost of each edge (v_j, v_{j+1}) be 0.

Now, when the agent is standing at node v_j , it evaluates the cost of going directly to t as μ^j , while the cost of the two-step path through v_{j+1} to t is evaluated as $0 + \beta\mu^{j+1} = (\beta\mu)\mu^j < \mu^j$. Thus the agent will follow the edge (v_j, v_{j+1}) with the plan of continuing from v_{j+1} to t . But this holds for all j , so once it reaches v_{j+1} , it changes its mind and continues on to v_{j+2} , and so forth. Ultimately it reaches v_n , and then must go directly to t at a cost of $c_\beta(s, t) = \mu^n$. Since $d(s, t) = 1$ by using the edge directly from s to t , this establishes the exponential lower bound on the cost ratio $c_\beta(s, t)/d(s, t)$. Essentially, this construction shows that the Akerlof example can be made quantitatively much worse than its original formulation by having the cost of going directly to the goal grow by a modest constant factor in each time step; when a present-biased agent procrastinates in this case, it ultimately incurs an exponentially large cost.

The following observation establishes this is the highest possible cost ratio

OBSERVATION 3.1. *Consider an agent currently at v and let u be the next node on $P_\beta(v, t)$. Then $d(u, t) < b \cdot d(v, t)$.*

PROOF. If u is on the shortest path from v to t then clearly the claim holds. Else, since the agent chose to continue to u instead of continuing on the shortest path from v to t we have $c(v, u) + \beta d(u, t) \leq d(v, t)$. This implies that $d(u, t) \leq b \cdot d(v, t)$. $\square$

The observation essentially implies that with each step that the agent takes the cost of the path that it plans to take increases by an extra factor of at most b relatively to $d(s, t)$. Hence, we have a tight upper bound on the cost ratio:

COROLLARY 3.2. *The cost ratio for a graph G with n nodes is at most b^{n-2} .*

A graph minor characterization

We now provide a structural description of the graphs on which the cost ratio can be exponential in the number of nodes n —essentially we show that a constant fraction of the nodes in such a graph must have the structure of the Akerlof example.

We make this precise using the notion of a *minor*. Given two undirected graphs H and K , we say that H *contains a K -minor* if we can map each node κ of K to a connected subgraph S_κ in H , with the properties that (i) S_κ and $S_{\kappa'}$ are disjoint for every two nodes κ, κ' of K , and (ii) if (κ, κ') is an edge of K , then in H there is some edge connecting a node in S_κ to

a node in $S_{k'}$. Informally, the definition means that we can build a copy of K using the structure of H , with disjoint connected subgraphs of H playing the role of “super-nodes” that represent the nodes of K , and with the adjacencies among these super-nodes representing the adjacencies in K . The minor relation shows up in many well-known results in graph theory, perhaps most notably in Kuratowski’s Theorem that a nonplanar graph must contain either the complete graph K_5 or the complete bipartite graph $K_{3,3}$ as a minor.⁶

Our goal here is to show that if G has exponential cost ratio, then its undirected version must contain a large copy of the graph underlying the Akerlof example as a minor. In other words, the Akerlof example is not only a way to produce a large cost ratio, but it is in a sense an unavoidable signature of any example in which the cost ratio is very large.

We set this up as follows. Let $\sigma(G)$ denote the skeleton of G , the undirected graph obtained by removing the directions on the edges of G . Let $\mathcal{F}_k$ denote the graph with nodes $v_1, v_2, \dots, v_k$, and w , and edges (v_i, v_{i+1}) for $i=1, \dots, k-1$, and (v_i, w) for $i=1, \dots, k$. We refer to $\mathcal{F}_k$ as the k -fan.

We now claim

THEOREM 3.3. *For every $\lambda > 1$, if n is sufficiently large and the cost ratio is greater than λ^n , then $\sigma(G)$ contains an $\mathcal{F}_k$ -minor for $k = \Theta(n)$.*

Proof sketch.^c We now provide a sketch of the proof. The basic idea is to pin down $k = \Theta(n)$ nodes, $v_1, \dots, v_k$, on the path $P_\beta(s, t)$ and let P be the portion of the path from s to v_k . We show that from each v_i the shortest path Q_i intersects with P only at v_i . This is illustrated in Figure 3.

Once we have this structure, we obtain an $\mathcal{F}_k$ minor by partitioning P into segments such that each v_i is in a distinct segment, and then we contract each segment. Since all the Q_i ’s are connected to t , we can contract them together (without the v_i ’s). Finally, as all segments of P are connected to one another, and each segment is connected to t by Q_i , we get a fan $\mathcal{F}_k$.

For simplicity we normalize the edges costs such that $d(s, t) = 1$. We choose the nodes $v_1, \dots, v_k$ such that for every i , v_i is the last node on the path P such that $d(v_i, t) \leq b^i$. With this choice it is not hard to show that the shortest path from v_i to t (Q_i) intersects with P only at v_i . In particular, Q_i can not intersect with P at any node before v_i since this will create

^c This sketch is based on the full proof in our original paper, and also draws on the exposition in Tim Roughgarden’s lecture notes about our paper.¹⁷

Figure 3. The construction of an $\mathcal{F}_k$ -minor in Theorem 3.3.

a directed cycle and our graph is a directed acyclic graph. Furthermore, Q_i can not intersect with P at any node u after v_i since by our choice of v_i , we have $d(u, t) > b^i \geq d(v_i, t)$ for every node u after v_i .

Finally, we should show that indeed for every $1 \leq i \leq k$, there exists a distinct node v_i with the property that (i) $d(v_i, t) \leq b^i$, and (ii) for any node u after v_i on $P_\beta(s, t)$, we have $d(u, t) > b^i$. First observe that since the cost ratio is λ^n and the length of the path is at most n , the path must contain an edge (u, v) of cost at least λ^n/n . Roughly speaking, since n is large enough there exists $\lambda_0 > 1$ such that $\lambda^n/n = \lambda_0^n$. In particular this implies that $d(u, t) \geq \lambda_0^n$. By Observation 3.1 we have that with each step on the path P the cost of the shortest path can increase by at most a factor of b . Thus, there exist k nodes as required.

After the initial publication of our original paper, Tang et al.²⁰ extended Theorem 3.3 as follows:

THEOREM 3.4. *If the cost ratio is greater than b^{k-2} , then $\sigma(G)$ contains an $\mathcal{F}_k$ -minor.*

Both Theorem 3.3 and its tighter counterpart Theorem 3.4 offer some qualitatively relevant advice for thinking about the structure of complicated tasks: to avoid creating inefficient behavior due to present bias, it is better to organize tasks so that they do not contain large fan-like structures. The point to appreciate is that such fan-like structures are not purely graph-theoretic abstractions; they arise in real settings whenever a task has a series of “branches” (as in Akerlof’s story) that allows an agent to repeatedly put off completing the task. The theorems are a way of formalizing the idea that such sets of repeated branches are the crux of the reason why present-biased individuals incur unnecessary inefficiency in completing large tasks. And correspondingly, organizing tasks in a way that breaks this type of branching—for example, with intermediate deadlines or subgoals—can be a way of reducing inefficiency.

4. COLLECTIONS OF HETEROGENEOUS AGENTS

Thus far we have focused on the behavior of a single agent with a given present-bias parameter β . Now we consider all possible values of β , and ask the following basic question: how large can the set $\{P_\beta(s, t) : \beta \in [0, 1]\}$ be? In other words, if for each β , an agent with parameter β were to construct an s - t path in G , how many different paths would be constructed across all the agents? Bounding this quantity tells us how many genuinely “distinct” types of behaviors there are for the instance defined by G . Let $\mathcal{P}(G)$ denote the set $\{P_\beta(s, t) : \beta \in [0, 1]\}$. Despite the fact that β comes from the continuum $[0, 1]$, the set $\mathcal{P}(G)$ is clearly finite, since G only has finitely many s - t paths. The question is whether we can obtain a nontrivial upper bound on the size of $\mathcal{P}(G)$, and in particular one that does not grow exponentially in the number of nodes n . In fact this is possible, and our main goal in this section is to prove the following theorem.

THEOREM 4.1. *For every directed acyclic graph G , the size of*

$\mathcal{P}(G)$ is $O(n^2)$. Moreover, there exists a graph for which the size of $\mathcal{P}(G)$ is $\Omega(n^2)$.

Proof idea. We use the following procedure to “discover” all the paths in $\mathcal{P}(G)$. We start by taking $\beta = 0$ and let P_0 the path that the agent with $\beta = 0$ takes. Now, we gradually increase β till we reach β^* such that an agent with β^* will take a different path. We claim that there is at least one edge (v, u) in P_0 that will not take part in any path that an agent with $\beta > \beta^*$ will take. More generally, each time that we discover a new path, we essentially delete at least one edge from the graph. Hence the number of paths in $\mathcal{P}(G)$ is bounded by the number of edges in the graph.

Theorem 4.1 tells us that the effect of present-bias on the path that an agent takes is in some sense limited as the specific value of β an agent has determines which path will the agent take from a precomputed set of at most $O(n^2)$ different paths. Since this $O(n^2)$ quantity is much smaller in general than the full set of possible paths, it says that the possible heterogeneity in agent behavior based on different levels of present bias is not as extensive as it might initially seem. Furthermore, quantifying this heterogeneity is a first step in designing efficient task graphs for populations of agents that are heterogeneous.

5. MOTIVATING AN AGENT TO REACH THE GOAL

We now consider the version of the model with rewards: there is a reward at t , and the agent has the additional option of quitting if it perceives—under its present-biased evaluation—that the value of the reward is not worth the remaining cost in the path.

Note that the presence of the reward does not affect the agent’s choice of path, only whether it continues along the path. Thus we can clearly determine the minimum reward r required to motivate the agent to reach the goal in G by simply having it construct a path to t according to our standard fixed-goal model, identifying the node at which it perceives the remaining cost to be the greatest (due to present bias this might not be s), and assigning this maximum perceived cost as a reward at t .

A more challenging question is suggested by the possibility of deleting nodes and edges from G ; recall that Figure 2(b) showed a basic example in which the instructor of a course was able to motivate a student to finish the coursework by deleting a node from the underlying graph. (This deletion essentially corresponded to introducing a deadline for the first piece of work.) This shows that even if the reward remains fixed, in general it may be possible for a designer to remove parts of the graph, thereby reducing the set of options available to the agent, so as to get the agent to reach the goal. We now consider the structure of the subgraphs that naturally arise from this process.

Motivating subgraphs: A fundamental example

The basic set-up we consider is the following. Suppose the agent in the reward model is trying to construct a path from s to t in G ; the reward r is not under our control—perhaps it is defined by a third party, or represents an intrinsic reward that we can not augment—but we are able to remove nodes and edges from the graph (essentially by declaring certain

activities invalid, as the deadline did in Figure 2(b)). Let us say that a subgraph G' of G motivates the agent if in G' with reward r , the agent reaches the goal node t . (We will also refer to G' as a *motivating subgraph*.) Note that it is possible for the full graph G to be a motivating subgraph.

It would be natural to conjecture that if there is any subgraph G' of G that motivates the agent, then there is a motivating subgraph consisting simply of an s - t path P that the agent follows. In fact this is not the case. Figure 4 shows a graph illustrating a phenomenon that we find somewhat surprising *a priori*, though not hard to verify from the example. In the graph G depicted in the figure, an agent with $\beta = 1/2$ will reach the goal t . However, there is no proper subgraph of G in which the agent will reach the goal. The point is that the agent starts out expecting to follow the path s - a - b - t , but when it gets to node a it finds the remainder of the path a - b - t too expensive to justify the reward, and it switches to a - c - t for remainder. With just the path s - a - b - t in isolation, the agent would get stuck at a ; and with just s - a - c - t , the agent would never start out from s . It is crucial the agent mistakenly believes the upper path is an option in order to eventually use the lower path to reach the goal.

It is interesting, of course, to consider real-life analogues of this phenomenon. In some settings, the structure in Figure 4 could correspond to deceptive practices on the part of the designer of G —in other words, inducing the agent to reach the goal by misleading them at the outset. But there are other settings in real life where one could argue that the type of deception represented here is more subtle, not any one party’s responsibility, and potentially even salutary. For example, suppose the graph schematically represents the learning of a skill such as a musical instrument. There’s the initial commitment corresponding to the edge (s, a) , and then the fork at a where one needs to decide whether to “get serious about it” (taking the expensive edge (a, b)) or not (taking the cheaper edge (a, c)). In this case, the agent’s trajectory could describe the story of someone who derived personal value from learning the violin (the lower path) even though at the outset they believed incorrectly that they would be willing to put the work into becoming a concert violinist (the upper path).

The structure of minimal motivating subgraphs

Given that there is sometimes no single path in G that is motivating, how rich a subgraph do we necessarily need to motivate the agent? Let us say that a subgraph G^* of G is a *minimal motivating subgraph* if (i) G^* is motivating, and (ii) no proper subgraph of G^* is motivating. Thus, for example,

Figure 4. A minimal subgraph for getting an agent to reach t .

in Figure 4, the graph G is a minimal motivating subgraph of itself; no proper subgraph of G is motivating.

Concretely, then, we can ask the following question: what can a minimal motivating subgraph look like? For example, could it be arbitrarily dense with edges?

In fact, minimal motivating subgraphs necessarily have a sparse structure, which we now describe in our next theorem. To set up this result, we need the following definition. Given a directed acyclic graph G and a path P in G , we say that a path Q in G is a P -bypass if the first and last nodes of Q lie on P , and no other nodes of Q do; in other words, $P \cap Q$ is equal to the two ends of Q . We now have

THEOREM 5.1. *If G^* is a minimal motivating subgraph, then it contains an s - t path P^* with the properties that*

- (i) *Every edge of G^* is either part of P^* or lies on a P^* -bypass in G^* ; and*
- (ii) *Every node of G^* has at most one outgoing edge that does not lie on P^* .*

Proof sketch. Roughly speaking, there are two types of edges that should be included in a minimal motivating subgraph: edges that the agent will actually take (these are the edges of the path P^*) and edges that at some point the agent (wrongly) believes that it will take (these are the edges on the P^* -bypasses). It is clearly the case that an edge e that the agent never plans to take can be safely removed from the graph without affecting the agent's decisions. Furthermore, since all the bypass edges are only used in shortest path computations it is impossible for a node v in P^* to have two neighbors w_1 and w_2 not on P^* such that the agent at some v_1 on P^* plans to follow a path that includes the edge (v, w_1) and an agent at some v_2 on P^* plans to follow a path that includes the edge (v, w_2) . This is simply because if $(v, w_1) + d(w_1, t) \leq (v, w_2) + d(w_2, t)$ the agent will choose the path that contains (v, w_1) both when standing at v_1 and at v_2 and otherwise it will choose the path that contains (v, w_2) in both cases.

After the publication of our original paper, Tang et al.²⁰ and Albers and Kraft² independently showed that determining whether a task graph admits a motivating subgraph is NP-complete. One way of circumventing these hardness results is to identify task graphs that are more common in practice and asking whether on these graphs the motivating subgraph problem can be solved in polynomial time. Alternatively, we can consider approximation algorithms. Albers and Kraft² considered a variant of this question: what is the minimum reward r for which a motivating subgraph exists? They showed that this problem can not be approximated by a factor better than $\sqrt{n}/3$, and they presented a $(\sqrt{n}+1)$ approximation algorithm. Interestingly, the subgraph achieving the $\sqrt{n}+1$ approximation is a path. Surprisingly, in a different paper, Albers and Kraft³ were able to break the $\sqrt{n}$ barrier and presented a 2-approximation algorithm for a more powerful designer who is able to increase the costs of edges.

An alternative way to motivate an agent to reach t is to place intermediate rewards on specific nodes or edges; the agent will claim each reward if it reaches the node or edge

on which it is placed. Now the question is to place rewards on the nodes or edges of an instance G such that the agent reaches the goal t while claiming as little total reward as possible; this corresponds to the designer's objective to pay out as little as possible while still motivating the agent to reach the goal. Following our paper, Albers and Kraft² and Tang et al.²⁰ studied different versions of this question and showed that the problem of assigning intermediate rewards in an optimal way is NP-complete. A version worth mentioning is the one in which the designer only cares about minimizing the rewards that the agent actually takes. Such a formulation of the problem comes with the danger that a designer would be able to create "exploitive" solutions in which the agent is motivated by intermediate rewards that it will never claim, because these rewards are on nodes that the agent will never reach.

6. CONCLUSION AND SUBSEQUENT WORK

We have developed a graph-theoretic model in which an agent constructs a path from a start node s to a goal node t in an underlying graph G representing a sequence of tasks. Time-inconsistent agents may plan an s - t path that is different from the one they actually follow, and this type of behavior in the model can reproduce a range of qualitative phenomena including procrastination, abandonment of long-range tasks, and the benefits of a reduced set of options. Our results provide characterizations for a set of basic structures in this model, including for graphs achieving the highest cost ratios between time-inconsistent agents and shortest paths, and we have investigated the structure of minimal graphs on which an agent is motivated to reach the goal node.

There is a wide range of broader issues for further work. These include finding structural properties beyond our graph-minor characterization that have a bearing on the cost ratio of a given instance; obtaining a deeper understanding of the relationship between agents with different levels of time-inconsistency as measured by different values of β ; and developing algorithms for designing graph structures that motivate effort as efficiently as possible, including for multiple agents with diverse time-inconsistency properties.

In work following the initial appearance of our paper, the results were extended in several subsequent lines of research. We have discussed several of these further results in the text thus far, including a tighter graph-minor characterization of instances with high cost ratio,²⁰ and a set of hardness results and approximation algorithms for motivating subgraphs.^{2, 3, 20} Further results beyond these have considered the behavior of different types of agents. Gravin et al.⁸ studied the behavior of a present-biased agent whose present-bias parameter is not fixed over time; rather, after each step it re-samples the parameter from some fixed distribution. In Kleinberg et al.,¹⁰ the authors of the present paper together with Manish Raghavan studied the behavior of *sophisticated* agents (building on a formalization from O'Donoghue and Rabin¹⁴), who are aware of their present bias and can take it into account in their planning. Such agents are not able to ignore or disregard their present

bias—they still prefer avoiding costs in the present time period—but their sophistication does mean that they can take measures to reduce the future effects of present bias in their plans. Finally Kleinberg et al.,¹¹ studied the behavior of agents that exhibit two biases concurrently: present bias, as in this paper, and *sunk-cost bias*, which is the tendency to reason about costs already incurred in the formulation of plans for the future.

Acknowledgments

We thank Supreet Kaur, Sendhil Mullainathan, and Ted O'Donoghue for valuable discussions and suggestions.

References

1. Akerlof, G.A. Procrastination and obedience. *American Econ. Rev.: Papers and Proc.* 81 (1991).
2. Albers, S., Kraft, D. Motivating time-inconsistent agents: A computational approach. In *Intl. Conf. Web and Internet Econ.* (2016).
3. Albers, S., Kraft, D. On the value of penalties in time-inconsistent planning. In *Proc. 44th Intl. Colloq. on Automata, Languages and Programming* (2017).
4. Ariely, D., Wertenbroch, K. Procrastination, deadlines, and performance: self-control by precommitment. *Psych. Sci.* 13 (2002).
5. Carstensen, P. The complexity of some problems in parametric linear and combinatorial programming. PhD thesis, U. Michigan (1983).
6. Diestel, R. *Graph Theory*, 3 edn. Springer (Berlin/Heidelberg, 2005).
7. Frederick, S., Loewenstein, G., O'Donoghue, T. Time discounting and time preference. *J. Econ. Lit.* 40, 2 (June 2002), 351–401.
8. Gravin, N., Immorlica, N., Lucier, B., Pountourakis, E. Procrastination with variable present bias. In *Proc. ACM Conf. Econ. Comp.* (2016).
9. Kaur, S., Kremer, M., Mullainathan, S. Self-control and the development of work arrangements. *American Econ. Rev.: Papers and Proceedings* 100 (2002).
10. Kleinberg, J., Oren, S., Raghavan, M. Planning problems for sophisticated agents with present bias. In *Proc. ACM Conf. Econ. Comp.* (2016).
11. Kleinberg, J., Oren, S., Raghavan, M. Planning with multiple biases. In *Proc. ACM Conf. Econ. Comp.* (2017).
12. Laibson, D. Golden eggs and hyperbolic discounting. *Q. J. Econ.* 112, 2 (1997), 443–478.
13. Nikolova, E., Kelner, J.A., Brand, M., Mitzenmacher, M. Stochastic shortest paths via quasi-convex maximization. In *Proc. 14th European Symposium on Algorithms* (2006), 552–563.
14. O'Donoghue, T., Rabin, M. Doing it now or later. *American Econ. Rev.* 89 (1999).
15. O'Donoghue, T., Rabin, M. Procrastination on long-term projects. *J. Econ. Behav. Org.* 66 (2008).
16. Pollak, R.A. Consistent planning. *Rev. Econ. Studies* 35, 2 (Apr. 1968), 201–208.
17. Roughgarden, T. Lecture 19: Time-inconsistent planning, 2016. <http://theory.stanford.edu/tim/f16/l19.pdf>.
18. Russell, S.L., Norvig, P. *Artificial Intelligence: A Modern Approach*. Prentice Hall (Upper Saddle River NJ, USA, 1994).
19. Strotz, R.H. Myopia and inconsistency in dynamic utility maximization. *Rev. Econ. Studies* 23 (1955).
20. Tang, P., Teng, Y., Wang, Z., Xiao, S., Xu, Y. Computational issues in time-inconsistent planning. In *Proc. 31st AAAI Conf. Artificial Intelligence* (2017).

Jon Kleinberg (kleinber@cs.cornell.edu), Cornell University, Ithaca, NY, USA.

Sigal Oren (sigal3@gmail.com), Ben Gurion University of the Negev, Be'er Sheva, Israel.

© 2018 ACM 0001-0782/18/3 \$15.00

He was the conscience of the computing industry...and paid for it.

The first full-length biography of Edmund Berkeley, computing pioneer, social activist, and founding member of the ACM. It is an historical narrative of a man ultimately in favor of engineering peace, instead of war, and how his career was ultimately damaged by politicians determined to portray him as a Communist sympathizer. Berkeley's life work provides a lens to understand social and political issues surrounding the early development of electronic computers which ties directly to current debates about the use of autonomous intelligent systems.

ISBN: 978-1-970001-36-5 DOI: 10.1145/2787754
<http://books.acm.org>
<http://www.morganclaypoolpublishers.com/berkeley>