Mathematical Social Sciences 119 (2022) 56–67

Contents lists available at ScienceDirect

Mathematical Social Sciences

journal homepage: www.elsevier.com/locate/mss

Present-biased optimization☆

Fedor V. Fomin [a,*], Pierre Fraigniaud [b], Petr A. Golovach [a]

[a] Department of Informatics, University of Bergen, PB 7803, 5020 Bergen, Norway
[b] IRIF, Université de Paris and CNRS, France

ARTICLE INFO

Article history:
Received 16 June 2021
Received in revised form 6 January 2022
Accepted 3 June 2022
Available online 21 June 2022

Keywords:
Time-inconsistent planning
Present-biased agent
Combinatorial optimization
Hitting set

ABSTRACT

This paper explores the behavior of *present-biased* agents, that is, agents who erroneously anticipate the costs of future actions compared to their real costs. Specifically, we extend the original framework proposed by Akerlof (1991) for studying various aspects of human behavior related to time-inconsistent planning, including procrastination, and abandonment, as well as the elegant graph-theoretic model encapsulating this framework recently proposed by Kleinberg and Oren (2014). The benefit of this extension is twofold. First, it enables to perform fine-grained analysis of the behavior of present-biased agents depending on the *optimization task* they have to perform. In particular, we study covering tasks vs. hitting tasks and show that the ratio between the cost of the solutions computed by present-biased agents and the cost of the optimal solutions may differ significantly depending on the problem constraints. Second, it enables us to study not only the underestimation of future costs, coupled with minimization problems, but also all combinations of minimization/maximization, and underestimation/overestimation. We study the four scenarios, and establish upper bounds on the cost ratio for three of them (the cost ratio for the original scenario was known to be unbounded), providing a complete global picture of the behavior of present-biased agents, as far as optimization tasks are concerned.

1. Introduction

1.1. The framework

Present bias is the term used in behavioral economics to describe the gap between the anticipated costs of future actions and their real costs. A simple mathematical model of present bias was suggested by Akerlof (1991). In this model the cost of an action that will be perceived in the future is assumed to be β times smaller than its actual cost, for some constant $\beta < 1$, called the *degree of present bias*. The model was used for studying various aspects of human behavior related to *time-inconsistent* planning, including *procrastination*, and *abandonment*.

Kleinberg and Oren (2014, 2018) introduced an elegant graph-theoretic model encapsulating Akerlof's model. The approach is based on analyzing how an agent navigates from a source s to a target t in a directed edge-weighted graph G, called *task graph*. At any step, the agent chooses the next edge to traverse from the current vertex v thanks to an estimation of the length of the shortest path from v to t passing through each edge outgoing from v. A crucial characteristic of the model is that the estimation of the path lengths is *present-biased*. More specifically, the model of Kleinberg and Oren includes a positive parameter $\beta < 1$, the degree of present bias, and the length of a path $x_0, \dots, x_k$ from $x_0 = v$ to $x_k = t$ in G is evaluated as $\omega_0 + \beta \sum_{i=1}^{k-1} \omega_i$ where ω_i denotes the weight of edge (x_i, x_{i+1}), for every $i \in \{0, \dots, k-1\}$. As a result, the agent may choose an outgoing edge that is not on any shortest path from v to t, modeling *procrastination* by underestimating the cost of future actions to be performed whenever acting now in some given way. With this effect cumulating along its way from s to t, the agent may significantly diverge from shortest s-t paths, which demonstrates the negative impact of procrastination. Moreover, the *cost ratio*, which is the ratio between the cost of the path traversed by an agent with present bias and the cost of a shortest path, could be arbitrarily large. An illustrating example is depicted on Fig. 1, borrowed from Kleinberg and Oren (2018), and originally due to Akerlof (1991). Among many results, Kleinberg and Oren showed that any graph in which a present-biased agent incurs significantly more cost than an optimal agent must contain a large specific structure as a minor. This structure, called *procrastination structure*, is specifically the one depicted on Fig. 1.

There also could be situations when a researcher might want to design artificially intelligent agents intentionally with a β

☆ A preliminary version of this paper was presented at AAAI 2021. The research received funding from the Research Council of Norway via the project BWCA (grant no. 314528) and from the French National Research Agency (ANR) via the project "DESCARTES".

* Corresponding author.
E-mail addresses: fedor.fomin@uib.no (F.V. Fomin), pierre.fraigniaud@irif.fr (P. Fraigniaud), petr.golovach@uib.no (P.A. Golovach).

https://doi.org/10.1016/j.mathsocsci.2022.06.001

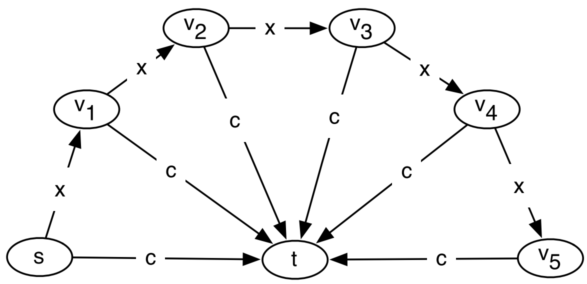

Fig. 1. Procrastination structure as displayed in Kleinberg and Oren (2018); Assuming $x + \beta c < c$, the path followed by the agent is $s, v_1, \ldots, v_5, t$; The ratio between the length of the path followed by the agent and the shortest s-t path can be made arbitrarily large by adding more nodes v_k with $k \geq 5$.

that is different from 1. Let us consider young male adults who tend to adopt conduct with risk, whether in extreme sports or simply driving vehicles. They exhibit overconfidence, which can be viewed as underestimating the probability of bad events. An AI driving a car must not show the same overconfidence. Such an AI's cost function must instead reflect one of prudent and experienced drivers, that is, be based on a relatively high value of the probability of bad events. However, it can be beneficial that AI driving an emergency vehicle (fire trucks, ambulances, etc.) be modified, by biasing the estimation of future actions, e.g., by forcing an underestimation of the probability of bad events, so that such vehicles do not remain stuck in long lines of cars because the AI driver is too careful and does not want to pass the cars in front.

In this paper, we are interested in understanding what kind of *tasks* performed by the agent result in a significant cost ratio. Let us take the concrete example of an agent willing to acquire the knowledge of a set of scientific concepts by reading books. Each book covers a certain number of these concepts, and the agent's objective is to read as few books as possible. More generally, each book could also be weighted according to, say, its accessibility to a general reader or its length. The agent's objective is then to read a collection of books with minimum total weight. Both the weight and the collection of concepts covered by each book are known to the agent a priori. This scenario is obviously an instance of the (weighted) *set-cover* problem. Let us assume, for simplicity, that the agent has access to a *present-biased* oracle providing it with the following information. Given the subset of concepts already acquired by the agent when it queries the oracle, the latter returns to the agent a set $\{b_0, \ldots, b_{k-1}\}$ of books minimizing

$$\omega_0 + \beta \sum_{i=1}^{k-1} \omega_i$$

among all sets of books covering the concepts yet to be acquired by the agent, where $\omega_0 \leq \omega_1 \leq \cdots \leq \omega_{k-1}$ are the respective weights of the books $b_0, \ldots, b_{k-1}$. This corresponds to the procrastination scenario in which the agent picks the easiest book to read now and underestimates the cost of reading the remaining books later. Then the agent moves on by reading b_0 and querying the oracle for figuring out the next book to read for covering the remaining uncovered concepts after having read book b_0. The question is: by how much the agent eventually diverges from the optimal set of books to be read? This set-cover example fits with the framework of Kleinberg and Oren, by defining the vertex set of the task graph as the set of all subsets of concepts, and placing an edge (u, v) of weight ω from u to v if there exists a book b of weight ω such that v is the union of u and the concepts covered by b. In this setting, the agent needs to move from the source $s = \varnothing$ to the target t corresponding to the set of all the concepts to be acquired by the agent. Under this setting, the question can be reformulated as: under which circumstances the set-cover problem yields a large cost ratio?

More generally, let us consider a *minimization* problem where, for every feasible solution S of every instance of the problem, the cost $c(S)$ of S can be expressed as $c(S) = \sum_{x \in S} \omega(x)$ for some weight function ω. This includes, e.g., set-cover, min-cut, minimum dominating set, feedback-vertex set, etc. We then define the biased cost c_β as

$$c_\beta(S) = \omega(x^*) + \beta\, c(S \setminus \{x^*\}), \tag{1}$$

where $x^* = \arg\min_{x \in S} \omega(x)$. Given an instance I of the minimization problem at hand, the agent aims at finding a feasible solution $S \in I$ minimizing $c(S)$. It does so using the following present-biased planning, where $I_0 = I$.

Minimization scenario. For $k \geq 0$, given an instance I_k, the agent computes the feasible solution S_k with minimum cost $c_\beta(S_k)$ among all feasible solutions for I_k. Let $x_k^* = \arg\min_{x \in S_k} w(x)$. The agent stops whenever $\{x_0^*, x_1^*, \ldots, x_k^*\}$ is a feasible solution for I. Otherwise, the agent moves to $I_{k+1} = I_k \setminus \{x_k^*\}$, that is, to the instance obtained from I_k when one assumes x_k^* selected in the solution.

Example. Let us consider the following concrete example, previously sketched above, namely *set-cover*. An agent is aiming at acquiring a collection of concepts related to algorithm design and analysis, like greedy algorithms, dynamic programming, recursive algorithms, flow, linear programs, approximation algorithms, parameterized algorithms, parallel algorithms, distributed algorithms, online algorithms, streaming algorithms, etc. For this purpose, the agent has several options, corresponding to the existing textbooks on these matters. Let us assume that the agent's library offers the following textbooks: x_{all}, a book covering all the aforementioned types of algorithms; x_{seq}, a book covering sequential (centralized) algorithms; x_{par} and x_{dist} covering parallel computing and distributed computing, respectively; And finally, x_{ext} covering all topics related to streaming, on line, etc., dealing with external inputs. (There are many other types of algorithms, but let us ignore them for the sake of simplicity).

We first describe the instance I_0. There are two feasible solutions: $S_0 = \{x_{all}\}$, and $S_0' = \{x_{seq}, x_{par}, x_{dist}, x_{ext}\}$. The weight $\omega(x)$ of a book x is impacted by various aspects such as writeup, pedagogy, length (i.e., number of pages), price, etc. Price is not an issue here, from the perspective of the agent, as the textbooks are accessed for free via a library. For simplicity, let us assume that the length of a book is the main criteria governing its weight, and is reflecting the effort required from the agent to read the book. Now, let us set $\omega(x_{all}) = 16$, and $\omega(x_{seq}) = \omega(x_{par}) = \omega(x_{dist}) = \omega(x_{ext}) = 5$. The fact that $\omega(x_{all}) < \omega(x_{seq}) + \omega(x_{par}) + \omega(x_{dist}) + \omega(x_{ext})$ is motivated from the assumption that each book must contain a section on notations and basic concepts (e.g., asymptotic analysis, big-O notations, etc.) which are common to all topics. While this section is present once in x_{all}, it is repeated in each book $x_{seq}, x_{par}, x_{dist}$, and x_{ext}. The fact that $\omega(x_{seq}) = \omega(x_{par}) = \omega(x_{dist}) = \omega(x_{ext})$ follows from the rough assumption that there is an identical amount of knowledge related to each subfield, whether it be sequential, parallel, distributed, or external algorithms.

It follows from our setting that $c(S_0) = 16$, $c(S_0') = 20$, and thus S_0 is the optimal solution, which should lead the agent to pick x_{all}. However, the agent is biased, by believing that being familiar with the notations and basic concepts read in one book enables saving a fraction $\frac{1}{3}$ of the time required to read another book on a related matter. Its degree of present-bias is therefore $\beta = \frac{2}{3}$. As a result, the agent evaluates $c_\beta(S_0) = 16$ correctly, but underestimates the cost of the collection of the other four books, by computing $c_\beta(S_0') = \omega(x_{seq}) + \frac{2}{3}\big(\omega(x_{par}) + \omega(x_{dist}) + \omega(x_{ext})\big) = 15$. It follows that the agent will grab $x_0^* = x_{seq}$ instead of x_{all}.

After having consumed 5 units of time for reading x_{seq}, the agent puts that book back on the shelf, and examines what remains to be done. The instance I_1 involves the original books x_{par}, x_{dist}, and x_{ext}, plus the "sub-book" $x_{all} \setminus x_{seq}$ including all the material of x_{all} not covered by x_{seq}. The weights of the books x_{par}, x_{dist}, and x_{ext} remain 5, because the intersection between the set of topics covered by x_{seq}, and the set of topics covered by $x_{par} \cup x_{dist} \cup x_{ext}$ is null. The weight of $x_{all} \setminus x_{seq}$ however decreases to $\omega(x_{all} \setminus x_{seq}) = \frac{3}{4}\,\omega(x_{all}) = 12$. There are two feasible solutions for I_1, namely $S_1 = \{x_{all} \setminus x_{seq}\}$, and $S_1' = \{x_{par}, x_{dist}, x_{ext}\}$. Again, the agent is biased, and evaluates the cost of these feasible solutions erroneously[1] as $c_\beta(S_1) = 12$ and $c_\beta(S_1') = \omega(x_{par}) + \frac{2}{3}\big(\omega(x_{dist}) + \omega(x_{ext})\big) = \frac{35}{3} < 12$. It follows that the agent grabs $x_1^* = x_{par}$ as the next book to read.

After the agent has read x_{par}, which takes 5 additional units of time, we consider the instance I_2. The books x_{dist} and x_{ext} remain with weight 5, while the sub-book $x_{all} \setminus (x_{seq} \cup x_{par})$ obtained from x_{all} by removing all chapters dedicated to sequential or parallel algorithms has weight $\omega(x_{all} \setminus (x_{seq} \cup x_{par})) = \frac{1}{2}\,\omega(x_{all}) = 8$. There are two feasible solutions for I_2, namely, $S_2 = \{x_{all} \setminus (x_{seq} \cup x_{par})\}$, and $S_2' = \{x_{dist}, x_{ext}\}$. The agent evaluates the cost of these feasible solutions as $c_\beta(S_2) = 8$ and $c_\beta(S_2') = \omega(x_{dist}) + \frac{2}{3}\,\omega(x_{ext}) = \frac{25}{3} > 8$. It follows that the agent eventually grabs $x_2^* = x_{all} \setminus (x_{seq} \cup x_{par})$ as the next book to read (i.e., x_{all}, but skipping the chapters dedicated to sequential and parallel computing).

After reading $x_0^* = x_{seq}$, $x_1^* = x_{par}$, and $x_2^* = x_{all} \setminus (x_{seq} \cup x_{par})$, the agent has completed its task. The agent eventually incurred a total cost of $\omega(x_0^*) + \omega(x_1^*) + \omega(x_2^*) = 5 + 5 + 8 = 18$, while the optimal solution was S_0 with a cost of 16. Thus, the present-biased agent incurred $\frac{18-16}{16} = 12.5\%$ more time than an optimal agent for acquiring all basic concepts in algorithm design and analysis. Note that this additional cost may even lead the agent to abandonment. This could typically happen if the agent's resources (willing to learn algorithms, issues related to the cost of leaving, etc.) are not sufficient to afford an extra 12.5% cost.

1.2. Our objectives

The minimization scenario is captured by the Kleinberg and Oren model, by defining the vertex set of the graph task graph as the set of all "sub-instances" of the instance I at hand, and placing an edge (u, v) of weight w from u to v if there exists an element x of weight ω such that v results from u by adding x to the current solution. The issue is to analyze how far the solution computed by the present-biased agent is from the optimal solution. The first question addressed in this paper is therefore the following.

Question 1. *For which minimization tasks a large cost ratio may appear?*

In the models of Akerlof (1991) and Kleinberg and Oren (2018) the degree β of present bias is assumed to be less than one. However, there are natural situations where underestimating the future costs does not hold. For example, in their influential paper, Loewenstein et al. (2003) gave a number of examples from a variety of domains demonstrating the prevalence of *projection bias*. In particular, they reported an experiment by Jepson et al. (2001) who "asked people waiting for a kidney transplant to predict what their quality of life would be one year later if they did or did not receive a transplant, and then asked those same people one year later to report their quality of life. Patients who received transplants predicted a higher quality of life than they ended up reporting, and those who did not predicted a lower quality of life than they ended up reporting". In other words, there are situations in which people may also *overestimate* the future costs. In the model of Kleinberg and Oren (2018) overestimation bias corresponds to the situation of putting the degree of present bias $\beta > 1$. This brings us to the second question.

Question 2. *Could a large cost ratio appear for minimization problems when the degree of present bias β is more than 1?*

Reformulating the analysis of procrastination, as stated in Question 1, inspires tackling related problems. In Kleinberg and Oren's original framework, procrastination is a priori associated to *minimization* problems. We also investigate *maximization* problems. A present-biased agent aims to maximize its revenue by making a sequence of actions, each providing some immediate gain that the agent maximizes while underestimating the incomes resulting from future actions. As a concrete example, let us consider an instance of Knapsack. The agent constructs a solution gradually by picking the item x_0 of highest value $\omega(x_0)$ in a feasible set $\{x_0, \ldots, x_{k-1}\}$ of items that is maximizing $\omega(x_0) + \beta \sum_{i=1}^{k-1} \omega(x_i)$ for the current sub-instance of Knapsack. In general, given an instance I of a maximization problem, we assume that the agent applies the following present-biased planning, with $I_0 = I$:

Maximization scenario. Given an instance I_k for $k \geq 0$, the agent computes the feasible solution S_k with maximum cost $c_\beta(S_k)$ among all feasible solutions for I_k — where the definition of x^* in Eq. (1) is replaced by $x^* = \arg\max_{x \in S} w(x)$. With $x_k^* = \arg\max_{x \in S_k} w(x)$, the agent stops whenever $\{x_0^*, x_1^*, \ldots, x_k^*\}$ is an inclusion-wise maximal feasible solution for I, and moves to $I_{k+1} = I_k \setminus \{x_k^*\}$ otherwise.

We are interested in analyzing how far the solution computed by the present-biased agent is from the optimal solution. More generally even, we aim at revisiting time-inconsistent planning by considering both cases $\beta < 1$ and $\beta > 1$, that is, not only scenarios in which the agent underestimates the cost of future actions, but also scenarios in which the agent *overestimates* the cost of future actions. The last, more general question addressed in this paper is therefore the following.

Question 3. *For which optimization tasks, and for which time-inconsistency planning (underestimation, or overestimation of the future actions), the solutions computed by a present-biased agent are far from optimal, and for which they are close?*

For all these problems, we study the *cost ratio* $\varrho = \frac{c(S)}{\text{OPT}}$ (resp., $\varrho = \frac{\text{OPT}}{c(S)}$) where S is the solution returned by the present-biased agent, and $\text{OPT} = c(S_{\text{OPT}})$ is the cost of an optimal solution for the same instance of the considered minimization (resp., maximization) problem.

1.3. Our results

Focusing on agents aiming at solving tasks, and not just on agents aiming at reaching targets in abstract graphs, as in the generic model in Kleinberg and Oren (2018), allows us not only to refine the worst-case analysis of present-biased agents, but also to extend this analysis to scenarios corresponding to overestimating the future costs to be incurred by the agents (by setting the degree β of present bias larger than 1), and to maximization problems.

[1] Note that the agent does not learn from his or her previous mistake. Scenarios in which the agent is changing attitude towards estimating the cost of future actions, based on the outcomes of previous actions, is beyond the scope of this paper.

Table 1
Upper bounds on the worst case ratio between the solution cost returned by the present-biased agent and the optimal solution OPT. The symbol ∞ means that the cost ratio can be arbitrarily large, independently of the values of β, and OPT.

	Minimization	Maximization
$\beta < 1$	∞ (Kleinberg and Oren, 2018)	$1/\beta$ [Theorem 5(i)]
$\beta > 1$	β [Theorem 4]	$(1 + \log \beta)\,\frac{\text{OPT}}{\log \text{OPT}}$ [Cor 1]

Minimization & underestimation. In the original setting of minimization problems, with underestimation of future costs (i.e., $\beta < 1$), we show that the cost ratio ϱ of an agent performing k steps, that is, computes a feasible solution $\{x_1^*, \dots, x_k^*\}$, satisfies $\varrho \leq k$. This is in contrast to the general model in Kleinberg and Oren (2018), in which an agent can incur a cost ratio exponential in k when returning a k-edge path from the source to the target. Hence, in particular, our minimization scenarios do not produce the worst cases examples constructed in Kleinberg and Oren (2018), i.e., obtained by considering travels from sources to targets in arbitrary weighted graphs.

On the other hand, we also show that a "minor structure" bearing similarities with the one identified in Kleinberg and Oren (2018) can be identified. Namely, if an agent incurs a large cost ratio, then the minimization problem addressed by the agent includes a large instance of a specific form of minimization problem.

We also study what realistic strategies could reduce the cost of inconsistent planning. We say that an agent is making a superfluous choice by selecting an element x of a feasible solution S, if the set $S \setminus \{x\}$ is also a feasible solution. Thus avoiding superfluous choices guarantees the minimality of the resulting solution. However, the strategy of the agent avoiding superfluous choices does not result in a bounded cost of planning.

We identify another natural class of strategies such that the agent following such a strategy consistently achieves a bounded cost ratio. Recall, that the agent constructs a feasible solution $S = \{x_0, \dots, x_k\}$ by computing for each $i \in \{1, \dots, k\}$ a feasible solution S_i and then selecting $x_i \in S_i$ minimizing the biased cost. We say that the agent's choice of x_i is *compatible* with the agent's previous actions if x_i also belongs to all previous feasible solutions $S_0, \dots, S_{i+1}$. We prove in Theorem 3 that when the agent makes the choices compatible with his previous actions, even if these choices are superfluous, the cost of inconsistent planning is bounded by a constant.

Min/maximization & under/overestimation. Interestingly, the original setting of minimization problems, with underestimation of future costs, is far from reflecting the whole nature of the behavior of present-biased agents. Indeed, while minimization problems with underestimation of future costs may result in unbounded cost ratios, the worst-case cost ratios corresponding to the three other settings can be upper bounded, some by a constant independent of the task at hand. Specifically, we show that:

- For any minimization problem with $\beta > 1$, the cost ratio is at most β;
- For any maximization problem with $\beta < 1$, the cost ratio is at most $\frac{1}{\beta}$;
- For any maximization problem with $\beta > 1$, the cost ratio is at most β^c, where $c \leq$ OPT is the cost of a solution constructed by the agent.

Our results are summarized in Table 1.

Let us remark that, for minimization problems with $\beta > 1$, as well as for maximization problems with $\beta < 1$, we have that the cost ratio is bounded by a constant. However, for maximization problems with $\beta > 1$, the cost ratio can be exponential in the cost of the computed solution. We show that this exponential upper bound is essentially tight.

Abandonment. One of the motivations of the original work on procrastination is *abandonment*. We do not put much emphasis on abandonment here. In the "classical" scenario, the abandonment occurs if the agent's budget is finite (it does not depend on the number of steps). For minimization with overestimation, and maximization with underestimation, our results show that, with a budget of at least $\beta \cdot$ OPT, abandonment does not occur.

Approximated evaluations. In many settings, discrete optimization problems are hard. Therefore, for evaluating the best feasible solution according to the biased cost function c_β, an agent may have to solve computationally intractable problems. Thus, in a more realistic scenario, we assume that, instead of computing an optimal solution for c_β at every step, the agent computes an α-approximate solution.

Fine-grained analysis. In contrast to the model of Kleinberg and Oren (2018), our model enables fine-grained analysis of the agents' strategies, that is, it enables identifying different behaviors of the agents as a function of the considered optimization problems. Specifically, there are natural minimization problems for which specific bounds on the cost ratio can be established.

To illustrate the interest of focusing on optimization tasks, we study two tasks in detail, namely *set-cover* and *hitting set*, and show that they appear to behave quite differently. For set-cover, we show that the cost ratio is at most $d \cdot$ OPT, where d is the maximum size of the sets. For hitting set, we show that the cost ratio is at most $d!\,(\frac{1}{\beta}\text{OPT})^d$, again for d equal to the maximum size of the sets. The proofs of these results build on the Sunflower Lemma of Erdős and Rado (1960), the classical result from extremal set theory.

Finally, in Theorem 9, we identify a large class of optimization problems where present-biased agents could obtain an optimum solution. This is the class of problems that can be encoded as the problem of computing a maximum-weight base of a matroid. Due to the expressive power of matroids, this class of problems include the problem of finding a maximum spanning tree in a graph (for graphic matroids), the problem of finding the set of independent vectors of maximum weight (for vector matroids), or the problem of computing the maximum number of disjoint paths connecting two sets in a graph (for gammoids).

1.4. Related work

Our work is directly inspired by Kleinberg and Oren (2014), which was itself motivated by the earlier work by Akerlof (1991). We refer to Kleinberg and Oren (2014, 2018) for a survey of earlier work on time-inconsistent planning, with connections to procrastination, abandonment, and choice reduction. Hereafter, we discuss solely (Kleinberg and Oren, 2014), and the subsequent work. Using their graph-theoretic framework, Kleinberg and Oren reasoned about time-inconsistency effects. In particular, they provided a characterization of the graphs yielding the worst-case cost-ratio, and they showed that, despite the fact that the degree β of present bias can take all possible values in [0, 1], it remains that, for any given digraph, the collection of distinct s-t paths computed by present-biased agents for all degrees of present bias is of size at most polynomial in the number of nodes. They also showed how to improve the behavior of present-biased agents by deleting edges and nodes, and they provided a characterization of the subgraphs supporting efficient agent's behavior. Finally, they analyzed the case of a collection of agents with different degrees of present bias, and showed how to divide the global task to be performed by the agents into "easier" sub-tasks, so that each agent performs efficiently her sub-tasks.

As far as we are aware of, all contributions subsequent to Kleinberg and Oren (2014), and related to our paper, essentially remain within the same graph theoretic framework as Kleinberg and Oren (2014), and focus on algorithmic problems related to this framework. In particular, Albers and Kraft (2019) studied the ability to place rewards at nodes for motivating and guiding the agent. They show hardness and inapproximability results, and provide an approximation algorithm whose performances match the inapproximability bound. The same authors considered another approach in Albers and Kraft (2017a) for overcoming these hardness issues, by allowing not to remove edges but to increase their weight. They were able to design a 2-approximation algorithm in this context. Tang et al. (2017) also proved hardness results related to the placement of rewards, and showed that finding a motivating subgraph is NP-hard. Gravin et al. (2016a) (see Gravin et al., 2016b for the full paper) extended the model by considering the case where the degree of present bias may vary over time, drawn independently at each step from a fixed distribution. In particular, they described the structure of the worst-case graph for any distribution, and derived conditions on this distribution under which the worst-case cost ratio is exponential or constant.

In the same paper, Gravin et al. described two natural conditions on the task graph that lead to smaller procrastination cost ratios. Their first property is the *bounded distance property*, when the weight of the shortest path in the task graph from any node v to the target t is at most the weight of the shortest path from the initial node s to t. Their second property is the *monotone distance property*. Informally, the task graph has a monotone distance property if, for any s-t path $(s = v_0, v_1, \dots, v_{k-1}, t = v_k)$, the shortest path from v_i to t is decreasing in i. These properties are incomparable to the properties we introduce in this paper. First, the property we consider in Section 2.2 (absence of superfluous choices) and in Section 2.3 (choices compatible with previous actions) are the properties of the agent's strategy. In contrast, the bounded and the monotone distance properties of Gravin et al. are the properties of the task graph. Second, the properties of the optimization problems (and hence of the task graph) that we use in Section 4, like d-set cover, or an independent set of a matroid, are also incomparable to the distance properties of Gravin et al. For example, it is easy to construct instances of d-set cover that are not distance monotone and *vice versa*.

Kleinberg et al. (2016, 2017) revisited the model in Kleinberg and Oren (2014). In Kleinberg et al. (2016), they were considering agents estimating erroneously the degree β of present bias, either underestimating or overestimating that degree, and compared the behavior of such agents with the behavior of "sophisticated" agents who are aware of their present-biased behavior in future and take this into account in their strategies. In Kleinberg et al. (2017), they extended the model by considering not only agents suffering from present-biases, but also from *sunk-cost* bias, i.e., the tendency to incorporate costs experienced in the past into one's plans for the future. Albers and Kraft (2017b) considered a model with uncertainty, bearing similarities with (Kleinberg et al., 2016), in which the agent is solely aware that the degree of present bias belongs to some set $B \subset (0, 1]$, and may or may not vary over time. Fomin and Strømme (2020) studied the parameterized complexity of computing a motivating subgraph.

2. Procrastination under minimization problems

This section includes a formal definition of inconsistent planning by present-biased agents, and describes two extreme scenarios: one in which a present-biased agent constructs worst case plannings, and one in which the plannings generated by a present-biased agent are close to optimal.

2.1. Model and definition

We consider minimization problems defined as triples $(\mathcal{I}, F, c)$, where $\mathcal{I}$ is the set of instances (e.g., the set of all graphs), F is a function that returns the set $F(I)$ of feasible solutions for every instance $I \in \mathcal{I}$ (e.g., the set of all edge-cuts of any given graph), and c is a non-negative function returning the cost $c(I, S)$ of every feasible solution $S \in F(I)$ of every instance $I \in \mathcal{I}$ (e.g., the number of edges in a cut). We focus solely on optimization problems, where the task is to find a subset of minimum weight, for which

(i) a finite ground set $\mathcal{S}_I \neq \emptyset$ is associated to every instance I,
(ii) every feasible solution for I is a set $S \subseteq \mathcal{S}_I$, and
(iii) $c(I, S) = \sum_{x \in S} \omega(x)$ where $\omega : \mathcal{S}_I \to \mathbb{N}$ is a weight function.

Moreover, we enforce two properties that are satisfied by classical minimization problems. Specifically we assume that:

- All considered problems are closed downward, that is, for every considered minimization problem $(\mathcal{I}, F, c)$, every $I \in \mathcal{I}$, and every $x \in \mathcal{S}_I$, the instance $I \setminus \{x\}$ defined by the feasible solutions $S \setminus \{x\}$, for every $S \in F(I)$, is in $\mathcal{I}$ with the same weight function ω as for I. This guarantees that an agent cannot be stuck after having performed some task x, as the sub-problem $I \setminus \{x\}$ remains solvable for every x.
- All considered feasible solutions are closed upward, that is, for every minimization problem $(\mathcal{I}, F, c)$, and every $I \in \mathcal{I}$, $\mathcal{S}_I$ is a feasible solution, and, for every $S \in F(I)$, if $S \subseteq S' \subseteq \mathcal{S}_I$ then $S' \in F(I)$. This guarantees that an agent performing a sequence of tasks $x_0, x_1, \dots$ eventually computes a feasible solution.

Inconsistent planning can be rephrased in this framework as follows.

Inconsistent planning. Let $\beta < 1$ be a positive constant. Given a minimization problem $(\mathcal{I}, F, c)$, the biased cost c_β satisfies

$$c_\beta(S) = \omega(x) + \beta\, c(S \setminus \{x\})$$

for every feasible solution S of every instance $I \in \mathcal{I}$, where

$$x = \arg\min_{y \in S} \omega(y).$$

Given an instance I, the agent aims at finding a feasible solution $S \in I$ by applying a present-biased planning defined inductively as follows. Let $I_0 = I$. For every $k \geq 0$, given the instance I_k, the agent computes a feasible solution S_k with minimum cost $c_\beta(S_k)$ among all feasible solutions for I_k. Let $x_k = \arg\min_{y \in S_k} \omega(y)$. The agent stops whenever $\{x_0, x_1, \dots, x_k\}$ is a feasible solution for I. Otherwise, it carries on the construction of the solution by considering $I_{k+1} = I_k \setminus \{x_k\}$.

Observe that inconsistent planning terminates. Indeed, since instances of the considered problem $(\mathcal{I}, F, c)$ are downward-closed, we have $I_k = I \setminus \{x_0, \dots, x_{k-1}\} \in \mathcal{I}$ for every $k \geq 0$. Hence inconsistent planning is well defined. Moreover, since feasible solutions are upward-closed, there exists $k \geq 0$ such that $\{x_0, x_1, \dots, x_k\}$ is a feasible solution for I.

Definition 1 (*Cost of Inconsistent Planning*). The *cost* of inconsistent planning is defined as the ratio

$$\varrho = \frac{c(S)}{\text{OPT}},$$

where $S = \{x_0, x_1, \dots, x_k\}$ is the solution returned by the agent, and $\text{OPT} = c(S_{\text{OPT}})$ is the cost of an optimal solution S_{OPT} for the same instance of the considered minimization problem.

Approximated evaluation. It can happen that the considered minimization problem is computationally hard, say NP-hard, and the agent is unable to compute a feasible solution S of minimum cost $c_\beta(S)$ exactly. Then the agent can pick an approximate solution instead. For this situation, we modify the above strategy of the agent as follows. Assume that the agent has access to an α-approximation algorithm $\mathcal{A}$ that, given an instance I, computes a feasible solution S^* to the instance such that $c_\beta(S^*) \le \alpha \min c_\beta(S)$, where minimum is taken over all feasible solution S to I. For simplicity, we assume throughout the paper that $\alpha \ge 1$ is a constant, but our results can be generalized for the case, where α is a function of the input size or OPT.

Again, the agent uses an inductive scheme to construct a solution. Initially, $I_0 = I$. For every $k \ge 0$, given the instance I_k, the agent computes a feasible solution S_k of cost at most $\alpha \min c_\beta(S)$, where the minimum is taken over all feasible solutions S of I_k. Then, exactly as before, the agent finds $x_k = \arg\min_{y\in S_k} \omega(y)$. If $\{x_0, x_1, \dots, x_k\}$ is a feasible solution for I, then the agent stops. Otherwise, we set $I_{k+1} = I_k \setminus \{x_k\}$ and proceed. The *α-approximative cost* of inconsistent planning is defined as the ratio $\varrho_\alpha = \frac{c(S)}{\text{OPT}}$ where $S = \{x_0, x_1, \dots, x_k\}$. Clearly, the 1-approximative cost coincides with ϱ.

2.2. Worst-case present-biased planning

We start with a simple observation. Given a feasible solution S for an instance I of a minimization problem, we say that $x \in S$ is *superfluous* in S if $S \setminus \{x\}$ is also feasible for I. The ability for the agent to make superfluous choices yields trivial scenarios in which the cost ratio ϱ can be arbitrarily large. This is for instance the case of an instance of Set Cover, defined as one set $y = \{1, \dots, n\}$ of weight $c > 1$ covering all elements, and n sets $x_i = \{i\}$, each of weight 1, for $i = 1, \dots, n$. Every solution $S_i = \{x_i, y\}$ is feasible, for $i = 1, \dots, n$, and satisfies $c_\beta(S_i) = 1 + \beta c$. As a result, whenever $1 + \beta c < c$, the present-biased agent constructs the solution $S = \{x_1, \dots, x_n\}$, which yields a cost ratio $\varrho = n/c$, which can be made arbitrarily large as n grows. Instead, if the agent avoids *superfluous* choices, that is, he systematically chooses only *minimal* feasible solutions, then the only feasible solutions $\{y\}$ and $\{x_1, \dots, x_n\}$ can be considered. As a result, the agent will compute the optimal solution $S_{\text{OPT}} = \{y\}$ if $c < 1 + \beta(n-1)$.

However, enforcing the agent to systematically choose *minimal* feasible solutions, i.e., solutions with no superfluous elements, is not sufficient to avoid procrastination. That is, the strategy of avoiding superfluous choices does not guarantee the agent a solution with a low cost ratio.

Example: Set cover instance $I_{SC}^{(n)}$. We denote by $I_{SC}^{(n)}$ the following special instance of the Set Cover problem. The instance of $I_{SC}^{(n)}$ consists of n elements $\{x_1, \dots, x_n\}$ and the following $2n$ subsets of $\{1, \dots, n\}$. Each subset is either a singleton set $\{x_i\}$ of weight 1 or $y_i = \{x_i, \dots, x_n\}$ of weight $c > 1$, for $i \in \{1, \dots, n\}$.

Every minimal feasible solutions of $I_{SC}^{(n)}$ has one of the following forms

- $\{y_1\}$ of weight c,
- $\{\{x_1\}, \dots, \{x_i\}, y_{i+1}\}$ of weight $i + c$ for $i \in \{1, \dots, n-1\}$, and
- $\{\{x_1\}, \dots, \{x_n\}\}$ of weight n.

Whenever $1 + \beta c < c$, a present-biased agent bounded to make only non-superfluous choices constructs the solution $\{\{x_1\}, \dots, \{x_n\}\}$. This is a solution of the cost ratio $\varrho = n/c$, and it grows to infinity with n.

We need the following lemma about biased solutions for minimization problems.

Lemma 1. *Let $\alpha \ge 1$ and let S^* be a feasible solution for a minimization problem, satisfying $c_\beta(S^*) \le \alpha \min c_\beta(S)$, where the minimum is taken over all feasible solutions. Then*

(i) $\omega(x) \le \alpha \cdot \text{OPT}$ for $x = \arg\min_{y\in S^*} \omega(y)$, and
(ii) $c(S^*) \le \frac{\alpha}{\beta}\text{OPT}$.

Proof. Let S be an optimum solution. As $\beta < 1$, it follows that $\omega(x) \le \omega(x) + \beta\cdot\omega(S^*\setminus\{x\}) = c_\beta(S^*) \le \alpha\cdot c_\beta(S) \le \alpha\cdot c(S) = \alpha\cdot\text{OPT}$, and this proves (i). To show (ii), note that $c(S^*) = \omega(x) + \omega(S^* \setminus \{x\}) = \frac{1}{\beta}(\beta\omega(x) + \beta\omega(S^*\setminus\{x\}))$, from which it follows that $c(S^*) \le \frac{1}{\beta}(\omega(x) + \beta\omega(S^* \setminus \{x\})) = \frac{1}{\beta}c_\beta(S^*) \le \frac{\alpha}{\beta}c_\beta(S) \le \frac{\alpha}{\beta}c(S) = \frac{\alpha}{\beta}\text{OPT}$, which completes the proof. □

Lemma 1 has a simple consequence that also can be derived from the results of Gravin et al. (2016b, Claim 5.1). We state it as a theorem despite its simplicity, as it illustrates one major difference between our model and the model in Kleinberg and Oren (2018).

Theorem 1. *For every $\alpha \ge 1$ and every minimization problem, the α-approximative cost ratio ϱ_α cannot exceed $\alpha \cdot k$, where k is the number of steps performed by the agent who is constructing a feasible solution $\{x_1, \dots, x_k\}$ by following the present-biased strategy.*

Proof. By Lemma 1(i), at any step $i \ge 1$ of the construction, the agent adds an element $x_i \in \mathcal{S}_I$ in the current partial solution, and this element satisfies $\omega(x_i) \le \alpha\, c_\beta(S_{\text{OPT}}) \le \alpha\, c(S_{\text{OPT}}) = \alpha \cdot \text{OPT}$. Therefore, if the agent computes a solution $\{x_1, \dots, x_k\}$, then the α-approximative cost ratio for this solution satisfies $\varrho_\alpha = \sum_{i=1}^{k} \omega(x_i)/\text{OPT} \le \alpha\, k$, as claimed. □

Remark. Theorem 1 exhibit the contrast between our model and the model of Kleinberg and Oren (2018), in which an agent performing k steps can incur a cost ratio exponential in k. This is because the model in Kleinberg and Oren (2018) enables to construct graphs with arbitrary weights. In particular, in a graph such as the one depicted on Fig. 1, one can set up weights such that the weight of (v_1, t) is a constant time larger than the weight of (s, t), the weight of (v_2, t) is in turn a constant time larger than the weight of (v_1, t), etc., and still a present-biased agent starting from s would travel via $v_1, v_2, \dots, v_k$ before reaching t. In this way, the sum of the weights of the edges traversed by the agent may become exponential in the number of traversed edges. This phenomenon does not occur when focusing on minimization tasks. Indeed, given a partial solution, the cost of completing this solution into a global feasible solution cannot exceed the cost of constructing a global feasible solution from scratch.

It follows from Theorem 1 that $I_{SC}^{(n)}$ is the worst-case instance. This instance fits with realistic procrastination scenarios in which the agent has to perform a task (e.g., learning a scientific topic T) by either energetically embracing the task (e.g., by reading a single thick book on topic T), or starting first by an easier subtask (e.g., by first reading a digest of a subtopic of topic T), with the objective of working harder later, but underestimating the cost of this postponed hard work. The latter strategy may result in procrastination, by performing a very long sequence of subtasks $x_1, x_2, \dots, x_n$.

In fact, $I_{SC}^{(n)}$ appears to be *the essence of procrastination* in the framework of minimization problems. Indeed, we show that if the cost ratio is large, then the considered instance I contains an instance of the form $I_{SC}^{(n)}$ with large n. More precisely, we say that an instance I contains an instance J as a *minor* if the ground set $\mathcal{S}_J$ associated to J is a collection of subsets of the ground set $\mathcal{S}_I$ associated to I, that is $\mathcal{S}_J \subseteq 2^{\mathcal{S}_I}$, and, for every $\bar{S} \subseteq \mathcal{S}_J$,

$\bar{S}$ is feasible for J if and only if $S = \bigcup_{\bar{x}\in\bar{S}} \bar{x}$ is feasible for I. Moreover, the weight function $\bar{\omega}$ for the elements of $\mathcal{S}_J$ must be induced by the one for $\mathcal{S}_I$ as $\bar{\omega}(\bar{x}) = \sum_{x\in\bar{x}} \omega(x)$ for every $\bar{x} \in \mathcal{S}_J$. Let $J^{(n)}$ be any instance of a minimization problem such that its associated ground set is $\mathcal{S}_{J^{(n)}} = \{x_1, \dots, x_n\} \cup \{y_1, \dots, y_n\}$, and the set of feasible solutions for $J^{(n)}$ is

$$F(J^{(n)}) = \Big\{ \{y_1\}, \{x_1, y_2\}, \{x_1, x_2, y_3\}, \dots, \{x_1, \dots, x_{n-1}, y_n\}, \{x_1, \dots, x_n\} \Big\}.$$

The following result sheds some light on *why* the procrastination structure of Fig. 1 pops up.

Theorem 2. *Let I be an instance of a minimization problem for which the present-biased agent with parameter $\beta \in (0, 1)$ computes a solution for I with cost $\alpha \cdot \text{OPT}(I)$ for some $\alpha > 1$. Then I contains $J^{(n)}$ as a minor for some $n \ge \alpha$, and the present-biased agent with parameter β computes a solution for $J^{(n)}$ with cost $\alpha \cdot \text{OPT}(J^{(n)})$.*

Proof. Let $S = \{x_1, \dots, x_n\}$ be the final solution selected by the present-biased agent for I, and let ω be the weight function on the set $\mathcal{S}_I$ associated to I. We have $\sum_{i=1}^{n} \omega(x_i) = \alpha\, \text{OPT}(I)$. For every $i \in \{1, \dots, n\}$, let us denote by $\text{OPT}(I \setminus \{x_1, \dots, x_i\})$ the cost of an optimal solution for the instance $I \setminus \{x_1, \dots, x_i\}$, and by $S_{\text{OPT}}(I \setminus \{x_1, \dots, x_i\})$ a corresponding optimal solution. For $i = 0$, $S_{\text{OPT}}(I \setminus \{x_1, \dots, x_i\})$ is an optimal solution for I. For $i \in \{1, \dots, n\}$, we define

$$\bar{x}_i = \{x_i\}, \text{ and } \bar{y}_i = S_{\text{OPT}}(I \setminus \{x_1, \dots, x_{i-1}\}).$$

Let J be the instance with ground set $\{\bar{x}_1, \dots, \bar{x}_n\} \cup \{\bar{y}_1, \dots, \bar{y}_n\}$, and feasible solutions

$$\{\bar{y}_1\}, \{\bar{x}_1, \bar{y}_2\}, \dots, \{\bar{x}_1, \dots, \bar{x}_{n-1}, \bar{y}_n\}, \{\bar{x}_1, \dots, \bar{x}_n\}.$$

Note that $\bar{x}_i \ne \bar{x}_j$ for every $i \ne j$, because $x_i \ne x_j$ for every $i \ne j$. Also, for every $i \in \{1, \dots, n-1\}$ and $j \in \{1, \dots, n\}$, $\bar{x}_i \ne \bar{y}_j$, because otherwise the sequence constructed by the present-biased agent for I would stop at x_k with $k < n$. Therefore, we have $J = J^{(n)}$, and, since $\omega(x_i) \le \text{OPT}(I)$ for every $i = 1, \dots, n$, $n \ge \alpha$ holds.

For analyzing the behavior of a present-biased agent with parameter β acting on J, let us assume that k steps were already performed by the agent, with $0 \le k < n$, resulting in constructing the partial solution $\{\bar{x}_1, \dots, \bar{x}_k\}$. (For $k = 0$, this partial solution is empty). The feasible solutions for $J_k = J \setminus \{\bar{x}_1, \dots, \bar{x}_k\}$ are

$$\{\bar{y}_1\}, \dots, \{\bar{y}_{k+1}\}, \{\bar{x}_{k+1}, \bar{y}_{k+2}\}, \dots, \{\bar{x}_{k+1}, \dots, \bar{x}_{n-1}, \bar{y}_n\}, \{\bar{x}_{k+1}, \dots, \bar{x}_n\}.$$

Note that, for every $i \in \{1, \dots, n\}$, $\bar{\omega}(\bar{x}_i) = \omega(x_i)$, and $\bar{\omega}(\bar{y}_i) = \text{OPT}(I \setminus \{x_1, \dots, x_{i-1}\})$. We claim that $\bar{x}_{k+1}$ is the next element chosen by the agent. Indeed, note first that $\bar{\omega}(\bar{x}_{k+1}) \le \bar{\omega}(\bar{y}_{k+2})$, as, otherwise, we would get $\omega(x_{k+1}) > \text{OPT}(I \setminus \{x_1, \dots, x_{k+1}\})$, contradicting the choice of x_{k+1} by the agent performing on I. As a consequence,

$$c_\beta(\{\bar{x}_{k+1}, \bar{y}_{k+2}\}) = \bar{\omega}(\bar{x}_{k+1}) + \beta\, \bar{\omega}(\bar{y}_{k+2}).$$

It follows from the above that, for every $j = 1, \dots, k+1$, $c_\beta(\{\bar{y}_j\}) \ge c_\beta(\{\bar{x}_{k+1}, \bar{y}_{k+2}\})$, as the reverse inequality would contradict the choice of x_{k+1} by the agent performing on I. For the same reason, for every $\ell \in \{k+1, \dots, n-1\}$, and every $i \in \{k+1, \dots, \ell\}$, we have

$$c_\beta(\{\bar{x}_{k+1}, \bar{y}_{k+2}\}) \le \bar{\omega}(\bar{x}_i) + \beta \Big(\sum_{j\in\{k+1,\dots,\ell\}\setminus\{i\}} \bar{\omega}(\bar{x}_j) + \bar{\omega}(\bar{y}_{\ell+1}) \Big)$$

and

$$c_\beta(\{\bar{x}_{k+1}, \bar{y}_{k+2}\}) \le \bar{\omega}(\bar{y}_{\ell+1}) + \beta \sum_{j\in\{k+1,\dots,\ell\}} \bar{\omega}(\bar{x}_j).$$

As a consequence, the present-biased agent performing on J picks $\bar{x}_{k+1}$ at step $k+1$, as claimed. The cost of the solution computed by the agent is $\sum_{i=1}^{n} \bar{\omega}(\bar{x}_i) = \sum_{i=1}^{n} w(x_i) = \alpha\, \text{OPT}(I)$. On the other hand, by construction, $\text{OPT}(J) = \bar{\omega}(\bar{y}_1) = \text{OPT}(I)$. The cost ratio of the solution computed by the agent for J is thus α, which completes the proof. □

2.3. Quasi-optimal present-biased planning

In the previous section, we have observed that forcing the agent to avoid superfluous choices, by picking minimal feasible solutions only. Such a strategy does not prevent the agent from constructing solutions that are arbitrarily far from the optimal. In this section, we show that, by enforcing the consistency in the sequence of partial solutions constructed by the agent, such a bad behavior does not occur. More specifically, given a feasible solution S for I, we say that the agent makes a choice *incompatible with* S, if he selects an object $x \notin S$. The following result shows that incompatible choices are the reason of a high-cost ratio.

Theorem 3. *An agent using an α-approximation algorithm and who is bounded to avoid incompatible choices with respect to the feasible solutions used in the past for constructing the current partial solution, returns an α/β-approximation of the optimal solution. This holds independently from whether the agent makes superfluous choices or not.*

Proof. Let I be an instance of a minimization problem $(\mathcal{I}, F, c)$. Let $S = \{x_0, \dots, x_k\}$ be the solution constructed by the agent for I, where x_i is the element computed by the agent at step i, for $i = 0, \dots, k$. Let S_i be the feasible solution of $I_i = I \setminus \{x_0, \dots, x_{i-1}\}$ considered by the agent at step i. Since the agent is bounded to avoid incompatible choices with respect to the past, we have $x_i \in \cap_{j=0}^{i} S_j$ for every $i = 0, \dots, k$ because $x_i \notin S_j$ for some $j < i$ would be an incompatible choice. It follows that $S \subseteq S_0$. Therefore, $c(S) \le c(S_0)$. Since the agent uses an α-approximation algorithm, by Lemma 1(ii), $c(S_0) \le \frac{\alpha}{\beta}\text{OPT}$ and the claim follows. □

At first glance, the assumption about compatible choices that we use in Theorem 3 does not look realistic. It implies that after the agent selects the first feasible solution S, he will be selecting only elements from S in his further actions. However, as we show in Section 4.2, for a generic optimization problem of finding a maximum-weight base in a matroid, the present-biased agent makes compatible choices.

3. Min/maximization with under/overestimation

3.1. Minimization with overestimation

We first investigate the cost ratio for minimization problems for the case when $\beta > 1$. Again, given a minimization problem $(\mathcal{I}, F, c)$, the biased cost c_β satisfies

$$c_\beta(S) = \omega(x) + \beta\, c(S \setminus \{x\})$$

for every feasible solution S of every instance $I \in \mathcal{I}$. However, now we have

$$x = \arg\max_{y\in S} \omega(y).$$

Given an instance I, the agent aims at finding a feasible solution $S \in I$ by applying a present-based planning defined inductively as follows. Let $I_0 = I$. For every $k \ge 0$, given the instance I_k, the agent computes a feasible solution S_k with minimum cost $c_\beta(S_k)$ among all feasible solutions for I_k. Let $x_k = \arg\max_{y\in S_k} \omega(y)$. The agent stops whenever $\{x_0, x_1, \dots, x_k\}$ is a feasible solution for I. Otherwise, it carries on the construction of the solution by considering $I_{k+1} = I_k \setminus \{x_k\}$.

The following theorem gives bound on the quality of the solution computed by a present-biased agent. Similar bound was obtained by Kleinberg et al. (see Kleinberg et al., 2016, Theorem 2.1). However, their theorem is about sophisticated agents and cannot be applied in our case directly.

Theorem 4. *Solutions computed by present-biased agents satisfy the following: For any minimization problem with $\beta > 1$, the cost ratio is at most β.*

Proof. For the proof of the theorem it is convenient to switch to the original graph-theoretic model of Kleinberg and Oren (2018). Note that the task graphs corresponding to optimization problems are, in fact, directed acyclic graphs. Hence, we only consider task graphs of this type to avoid dealing with paths of maximum length in the presence of cycles.

Let G be a directed acyclic graph (DAG) with a source s. Let also $\omega : E(G) \to \mathbb{N}$ be a weight function. The aim of the agent is to go from the source s to a sink t of G making present-biased decisions on each step. We assume that G has an s-t path. Let β be a positive constant distinct from 1. Let $c^{\min}(x)$ be the minimum length of an x-t path.

We suppose that the agent is equipped with an algorithm $\mathcal{A}$ that, given a vertex $v \in V(G)$, finds a vertex $x^* \in N_G^+(v)$ such that

$$\omega(vx^*) + \beta \cdot c^{\min}(x^*) = \min_{x \in N_G^+(v)} (\omega(vx) + \beta \cdot c^{\min}(x)).$$

The agent constructs an s-t path as follows: if the agent occupies a vertex $v \neq t$, then he makes the present-biased α-approximate estimation of the length of a shortest v-t path and moves to x^*. Note that since G is a DAG, the agent would eventually arrive to t.

We denote by $\textsc{cost}^{\min}(v)$ the length of a v-t path constructed by the agent from v. Notice that this value is not uniquely defined as the agent may be able to choose distinct vertices that provide α-approximate present-biased evaluations but could give distinct lengths for the constructed paths. Then the proof of the theorem is implied by the following claim.

Claim 3.1. *Let G be a weighted DAG with a weigh function $\omega : E(G) \to \mathbb{N}$ and a sink t. Then for every $v \in V(G)$, if $\beta > 1$, then $\textsc{cost}^{\min}(v) \leq \beta \cdot c^{\min}(v)$.*

The claim is trivial if $v = t$. Assume that $v \neq t$, and that the claim holds for every out-neighbor x of v. Assume that $x^* \in N_G^+(v)$ is computed by $\mathcal{A}$, and let

$$y = \arg\min_{x \in N_G^+(v)} (\omega(vx) + c^{\min}(x)).$$

That is, there is a shortest v-t path that goes through y.

By induction, we have that

$$\textsc{cost}^{\min}(x^*) \leq \beta \cdot c^{\min}(x^*).$$

It follows that

$$\begin{aligned} \textsc{cost}^{\min}(v) &= \omega(vx^*) + \textsc{cost}^{\min}(x^*) \leq \ \omega(vx^*) + \beta \cdot c^{\min}(x^*) \\ &= \min_{x \in N_G^+(v)} (\omega(vx) + \beta \cdot c^{\min}(x)) \leq \omega(vy) + \beta \cdot c^{\min}(y) \\ &\leq \beta \cdot \omega(vy) + \beta \cdot c^{\min}(y) \leq \ \beta \cdot c^{\min}(v). \end{aligned}$$

The last inequality completes the proof of the claim, and of the theorem. □

3.2. Maximization problems

Next, we consider maximization problems. As for minimization problems, given a maximization problem $(\mathcal{I}, F, c)$, the biased cost c_β satisfies

$$c_\beta(S) = \omega(x) + \beta\, c(S \setminus \{x\})$$

for every feasible solution S of every instance $I \in \mathcal{I}$. However, for maximization problems, the element x satisfies

$$\begin{cases} x = \arg\max_{y \in S} \omega(y) & \text{if } \beta < 1, \\ x = \arg\min_{y \in S} \omega(y) & \text{if } \beta > 1. \end{cases}$$

Given an instance I, the agent aims at finding a feasible solution $S \in I$ by applying a present-based planning defined inductively as follows. Let $I_0 = I$. For every $k \geq 0$, given the instance I_k, the agent computes a feasible solution S_k with maximum cost $c_\beta(S_k)$ among all feasible solutions for I_k. Let $x_k = \arg\max_{y \in S_k} \omega(y)$ if $\beta < 1$, and $x_k = \arg\min_{y \in S_k} \omega(y)$ if $\beta > 1$. The agent stops whenever $\{x_0, x_1, \ldots, x_k\}$ is a feasible solution for I. Otherwise, it carries on the construction of the solution by considering $I_{k+1} = I_k \setminus \{x_k\}$.

We establish the following worst-case bounds.

Theorem 5. *Solutions computed by present-biased agents satisfy the following:*

(i) For any maximization problem with $\beta < 1$, the cost ratio is at most $\frac{1}{\beta}$;
(ii) For any maximization problem with $\beta > 1$, the cost ratio is at most β^c, where $c \leq \textsc{opt}$ is the cost of a solution constructed by the agent.

Proof. As in the proof of Theorem 4, we switch to the graph-theoretic model from Kleinberg and Oren (2018). Let G be a directed acyclic graph (DAG) with a source s and weight function $\omega : E(G) \to \mathbb{N}$. The agent aims to go from the source s to a sink t of G making present-biased decisions on each step. We assume that G has an s-t path. Let β be a positive constant distinct from 1, and let $c^{\max}(x)$ be the maximum length of an x-t path.

Let $\alpha \in (0, 1]$. As in Theorem 4, we assume that the agent is equipped with an algorithm $\mathcal{A}$ that, given a vertex $v \in V(G)$, finds a vertex $x^* \in N_G^+(v)$ such that

$$\omega(vx^*) + \beta \cdot c^{\max}(x^*) = \max_{x \in N_G^+(v)} (\omega(vx) + \beta\, c^{\max}(x)).$$

Using $\mathcal{A}$, the agent located in vertex $v \neq t$ constructs an s-t path as follows: the agent computes the present-biased α-approximate estimation of the length of a longest v-t and moves to x^*. We denote by $\textsc{cost}^{\max}(v)$ the length of a v-t path constructed by the agent from v.

Claim 3.2. *Let G be a weighted DAG with a weigh function $\omega : E(G) \to \mathbb{N}$ and a sink t. Then for every $v \in V(G)$,*

(i) *if $\beta < 1$, then $\textsc{cost}^{\max}(v) \geq \beta\, c^{\max}(v)$,*
(ii) *if $\beta > 1$, then $c^{\max}(v) \leq \textsc{cost}^{\max}(v)\, \beta^{\textsc{cost}^{\max}(v)}$.*

As in Theorem 4, we prove the claim by induction. The claim is trivial if $v = t$. Assume that $v \neq t$, and that the claim holds for every out-neighbor x of v. Let x^* be the vertex computed by $\mathcal{A}$ and let

$$y = \arg\max_{x \in N_G^+(v)} (\omega(vx) + c^{\max}(x)).$$

That is, there is a longest v-t path that goes through y.

To show (i), we use the inductive assumption that

$$\textsc{cost}^{\max}(x^*) \geq \beta\, c^{\max}(x^*).$$

We have

$$\begin{aligned}\textsc{cost}^{\max}(v) &= \omega(vx^*) + \textsc{cost}^{\max}(x^*) \geq \omega(vx^*) + \beta\, c^{\max}(x^*)\\ &= \max_{x\in N_G^+(v)} (\omega(vx) + \beta\, c^{\max}(x)) \geq \omega(vy) + \beta\, c^{\max}(y)\\ &\geq \beta\, \omega(vy) + \beta\, c^{\max}(y) \geq \beta\, c^{\max}(v).\end{aligned}$$

To prove (ii), we assume that the following inductive assumption holds:

$$c^{\max}(x^*) \leq \textsc{cost}^{\max}(x^*)\, \beta^{\textsc{cost}^{\max}(x^*)}.$$

It follows that

$$\begin{aligned}c^{\max}(v) &= \omega(vy) + c^{\max}(y) \leq \omega(vy) + \beta\cdot c^{\max}(y)\\ &\leq \max_{x\in N_G(v)} (\omega(vx) + \beta\cdot c^{\max}(x)) \leq \omega(vx^*) + \beta\cdot c^{\max}(x^*)\\ &\leq \omega(vx^*) + \beta\cdot \textsc{cost}^{\max}(x^*)\, \beta^{\textsc{cost}^{\max}(x^*)}\\ &\leq \beta^{\textsc{cost}^{\max}(x^*)+1}(\omega(vx^*) + \textsc{cost}^{\max}(x^*))\\ &= \textsc{cost}^{\max}(v)\, \beta^{\textsc{cost}^{\max}(x^*)+1}\\ &\leq \textsc{cost}^{\max}(v)\, \beta^{\textsc{cost}^{\max}(v)}.\end{aligned}$$

This last inequality completes the proof of Claim 3.2, which immediately gives the bounds for the α-approximate cost ratio claimed in the statement of the theorem. □

We also can write the bound for the cost ratio for $\beta > 1$ in the following form to obtain the upper bound that depends only on the value of OPT.

Corollary 1. *For any maximization problem with $\beta > 1$, the cost ratio is at most $(1 + \log\beta)\frac{\textsc{opt}}{\log \textsc{opt}}$.*

Proof. Let c be the cost of a solution constructed by the agent. By Theorem 5, $\textsc{opt} \leq c\beta^c$. Therefore, $\log \textsc{opt} \leq \log c + c \log \beta \leq (1 + \log\beta)c$, and $\frac{\textsc{opt}}{c} \leq (1 + \log\beta)\frac{\textsc{opt}}{\log\textsc{opt}}$. □

For minimization problems with $\beta > 1$, and maximization problems with $\beta < 1$, we have that the cost ratio is bounded by a constant. This differs drastically with the case of maximization problems with $\beta > 1$, when the cost ratio is still bounded but the bound is exponential. This exponential upper bound is however essentially tight, in the sense that the exponent cannot be avoided.

Theorem 6. *There are maximization problems for which a present-biased agent with $\beta > 1$ returns a solution whose cost ratio is at least $\frac{1}{c}\,\beta^{c-1}$, where c is the cost of the solution constructed by the agent.*

Proof. Let us consider the maximum independent set problem. In this problem, we are given a weighted graph G, and the task is to find an independent set of maximum weight. Let k be a positive integer. We construct the graph G_k as follows (see Fig. 2):

- construct $k+1$ vertices $x_0, \ldots, x_k$, and make them pairwise adjacent,
- construct k vertices $y_1, \ldots, y_k$,
- for each $i \in \{1, \ldots, k\}$, make y_i adjacent to $x_i, x_{i+1}, \ldots, x_k$.

To define the weights, let $\beta \geq 2$. We set $\omega(y_i) = 1$ for every $i \in \{1, \ldots, k\}$, and $\omega(x_i) = \beta^i$ for every $i \in \{0, \ldots, k\}$.

Since $X = \{x_0, \ldots, x_k\}$ is a clique, any independent set has at most one vertex in X. Therefore, the family of sets

$$S_i = \{x_i\} \cup \{y_{i+1}, \ldots, y_k\}$$

for $i \in \{0, \ldots, k\}$ form the family of maximal independent sets. Because $\beta \geq 2$, it is straightforward to verify that the single-vertex set $S_k = \{x_k\}$ is an independent set of maximum weight

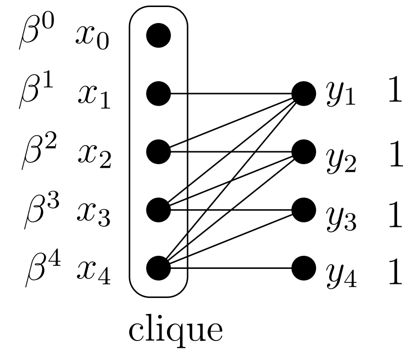

Fig. 2. Construction of G_k for $k = 4$.

β^k, that is, $\textsc{opt} = \beta^k$. Observe that the biased cost of this set is β^k as well.

From the other side, the biased cost of S_{k-1} is

$$\omega(y_k) + \beta\cdot\omega(x_{k-1}) = 1 + \beta\cdot\beta^{k-1} = 1 + \beta^k > \beta^k.$$

Hence, the agent would prefer to select y_k at the first iteration. At the next iteration, the agent considers the graph obtained from G_k by the deletion of y_k and its neighborhood, that is, G_{k-1}. Applying the same arguments inductively, we conclude that the agent will end up with the set $S_0 = \{x_0, y_1, \ldots, y_k\}$ with $\omega(S_0) = k+1$. We obtain that $\textsc{opt} = \beta^{c-1}$, where c is the cost of a solution constructed by the agent. □

Remark. An example similar to the one in the proof of Theorem 6 can be constructed for the knapsack problem. Recall that in this problem, we are given n objects with positive integer *values* v_i, and *weights* w_i, for $i \in \{1, \ldots, n\}$, and $W \in \mathbb{N}$. The task is to find a set of objects $S \subseteq \{1, \ldots, n\}$ of maximum value with the total weight at most W. Let k be a positive integer. We let $n = 2k+1$, $W = n$ and $\beta \geq 2$. We define

$$v_i = \beta^{k+1-i},$$

and

$$w_i = W - (i-1)$$

for every $i \in \{1, \ldots, k+1\}$, and we set $v_i = 1$ and $w_i = 1$ for $i \in \{k+2, \ldots, n\}$. Using the same arguments as for the maximum independent set problem, we obtain that the optimum solution has cost β^k while a present-biased agent would select a solution of cost $k+1$.

4. Fine-grained analysis of specific problems

In Section 3, we demonstrated upper bounds for the cost ratio, and in Section 2.2, we pointed that the ratio cannot be bounded by any function of OPT in the case of minimization with underestimation. However, for some specific problems, we can improve these results.

4.1. Minimum set-cover and hitting set problems with size constraints

In Section 2.2, we have seen instances of the set-cover problem whose cost ratio cannot be bounded by any function of OPT. The same obviously holds for the hitting-set problem. Recall that an instance of hitting-set is defined by a collection Σ of subsets of a finite set V, and the objective is to find the subset $S \subseteq V$ of minimum size, or minimum weight, which intersects (hits) every set in Σ. However, set-cover problems, and hitting set problems behave differently when the sizes of the sets are bounded. Throughout this subsection we consider the case of underestimation, that is, it is assumed that $\beta < 1$. First, we consider the d-set cover problem.

The d-set cover problem. Let d be a positive integer. The task of the d-set cover problem is, given a collection Σ of subsets with size at most d of a finite set V, and given a weight function $\omega\colon \Sigma \to \mathbb{N}$, find a set $S \subseteq \Sigma$ of minimum weight that covers V, that is, $\bigcup_{X\in S} X = V$.

Theorem 7. *Let $\alpha \geq 1$. For any instance of the d-set-cover problem, the α-approximative cost ratio is at most $\alpha \cdot d \cdot \textsc{opt}$.*

Proof. Let $I = (\Sigma, V, \omega)$ be an instance of the d-set cover problem. Let $|V| = n$. Denote by $S_1, \ldots, S_p$ a sequence of solutions computed by the present-biased agent avoiding superfluous solutions, and let $S = \{X_1, \ldots, X_p\}$ be the obtained solution for I. That is, S is a set cover such that

$$X_i = \arg\min_{Y\in S_i} \omega(Y)$$

for $i \in \{1, \ldots, p\}$. Clearly, $p \leq n$. Note that for each iteration $i \in \{1, \ldots, p\}$, the agent considers the instance $I_i = (\Sigma_i, V_i, \omega_i)$, where, as superfluous solutions are avoided,

$$\begin{array}{rcl} V_i &=& V \setminus (\bigcup_{j=1}^{i-1} X_i),\\ \Sigma_i &=& \{X \cap V_i \mid X \in \Sigma \setminus \{X_1, \ldots, X_{i-1}\}\},\\ \omega_i &=& \omega|_{\Sigma_i}. \end{array}$$

We have that, for every $i \in \{1, \ldots, p\}$,

$$\textsc{opt} = \textsc{opt}(I) = \textsc{opt}(I_1) \geq \cdots \geq \textsc{opt}(I_p),$$

and, by Lemma 1(ii),

$$\omega(X_i) \leq \alpha\, \textsc{opt}(I_i).$$

Since each set of Σ covers at most d elements of V, $\textsc{opt}(I) \geq n/d$. Therefore

$$c(S) = \sum_{i=1}^{p} \omega(X_i) \leq \alpha\, \textsc{opt}(I) n \leq \alpha d\, \textsc{opt}(I)^2.$$

It follows that the cost ratio is at most $\alpha d\, \textsc{opt}$. □

The d-hitting set problem. Let d be a positive integer. We are given a collection Σ of subsets with size d of a finite set V, a weight function $\omega\colon V \to \mathbb{N}$. The task is to find a set $S \subseteq V$ of minimum weight that hits every set of Σ.

We use the classical *Sunflower Lemma* of Erdős and Rado (1960). We state this result in the form given in Cygan et al. (2015). A *sunflower* with k *petals* and a *core* X is a collection of pairwise distinct sets $S_1, \ldots, S_k$ such that $S_i \cap S_j = X$ for all distinct $i, j \in \{1, \ldots, k\}$. Note that the core may be empty, that is, a collection of k pairwise disjoint sets is a sunflower.

Lemma 2 (*Sunflower Lemma, Erdős and Rado, 1960*)**.** *Let $\mathcal{A}$ be a family of pairwise distinct sets over a universe U such that for every $A \in \mathcal{A}$, $|A| = d$. If $|\mathcal{A}| > d!(k-1)^d$, then $\mathcal{A}$ contains a sunflower with k petals.*

Theorem 8. *Let $\alpha \geq 1$. For any instance of the d-hitting-set problem, the α-approximative cost ratio is at most $\alpha d!(\frac{\alpha}{\beta}\textsc{opt})^d$.*

Proof. Let $I = (\Sigma, V, \omega)$ be an instance of the d-hitting set problem. Denote by $S_1, \ldots, S_p$ a sequence of solutions computed by the present-biased agent avoiding superfluous solutions, and let $S = \{v_1, \ldots, v_p\}$ be the obtained solution for I, that is, $S \subseteq V$ is a hitting set such that

$$v_i = \arg\min_{v\in S_i} \omega(v)$$

for every $i \in \{1, \ldots, p\}$. Since the agent avoids superfluous solutions, the agent considers the instance $I_i = (\Sigma_i, V_i, \omega_i)$ at each iteration $i \in \{1, \ldots, p\}$, where

$$\begin{array}{rcl} V_i &=& V \setminus \{v_1, \ldots, v_{i-1}\},\\ \Sigma_i &=& \{X \in \Sigma \mid X \cap \{v_1, \ldots, v_{i-1}\} = \emptyset\},\\ \omega_i &=& \omega|_{V_i}. \end{array}$$

Let $i \in \{1, \ldots, p\}$. Since S_i is a minimal hitting set for Σ_i, there is $X_i \in \Sigma_i$ such that $v_i \in X_i$, and, for every $v \in S_i \setminus \{v_i\}$, $v \notin X_i$. We say that X_i is a *private* set for v_i. Observe that

$$\textsc{opt} = \textsc{opt}(I) = \textsc{opt}(I_1) \geq \cdots \geq \textsc{opt}(I_p)$$

by the construction of the instances. The following claim is crucial for the proof of the theorem.

Claim 4.1. $p \leq d!(\frac{\alpha}{\beta}\textsc{opt})^d$.

Let us assume, for the purpose of contradiction, that $p > d!(\frac{\alpha}{\beta}\textsc{opt})^d$. Consider the private sets $X_1, \ldots, X_p \in \Sigma$ for $v_1, \ldots, v_p$, respectively, and let $\mathcal{A} = \{X_1, \ldots, X_p\}$. Note that $X_1, \ldots, X_p$ are pairwise distinct, since $X_h \in \Sigma_h$ and $X_h \notin \Sigma_{h+1}$ for every $h \in \{1, \ldots, p-1\}$. Let

$$k = \left\lfloor \frac{\alpha}{\beta}\textsc{opt} \right\rfloor + 1.$$

We have that $|X_h| = d$ for every $X_h \in \mathcal{A}$, and $|\mathcal{A}| > d!(k-1)^d$. Hence, $\mathcal{A}$ contains a sunflower with k petals by the sunflower lemma. Denote by $X_{h_1}, \ldots, X_{h_k}$ the sets of the sunflower, and let Y be its core.

Suppose that $Y = \emptyset$. Then, $X_{h_1}, \ldots, X_{h_k}$ are pairwise disjoint. Note also that because $\alpha \geq 1$, $0 < \beta < 1$ and $\textsc{opt}$ is an integer, $\lfloor\frac{\alpha}{\beta}\textsc{opt}\rfloor \geq \textsc{opt}$. Therefore,

$$\textsc{opt}(I) \geq k = \left\lfloor \frac{\alpha}{\beta}\textsc{opt} \right\rfloor + 1 > \textsc{opt},$$

which is a contradiction. Therefore, $Y \neq \emptyset$.

We show that, for every hitting set R for Σ_{h_1} of weight at most $\frac{\alpha}{\beta}\textsc{opt}$, $R \cap Y \neq \emptyset$. Assume that this is not the case, that is, $R \cap Y = \emptyset$. Since R is a hitting set, there exists

$$u_\ell \in X_{h_\ell} \setminus Y$$

such that $u_\ell \in R$ for every $\ell \in \{1, \ldots, k\}$. Because $\{X_{h_1}, \ldots, X_{h_k}\}$ is a sunflower, $u_1, \ldots, u_k$ are distinct. It follows that

$$\omega(R) \geq \sum_{\ell=1}^{k} \omega(u_\ell) \geq k = \left\lfloor \frac{\alpha}{\beta}\textsc{opt} \right\rfloor + 1 > \frac{\alpha}{\beta}\textsc{opt}.$$

The latter strict inequality is contradicting the fact that the weight of R is at most $\frac{\alpha}{\beta}\textsc{opt}$. We conclude that $R \cap Y \neq \emptyset$.

Recall that S_{h_1} is a feasible solution for I_{h_1}, and, by Lemma 1 (ii),

$$\omega(S_{h_1}) \leq \frac{\alpha}{\beta}\textsc{opt}(I_{h_1}) \leq \frac{\alpha}{\beta}\textsc{opt}.$$

Then $S_{h_1} \cap Y \neq \emptyset$. The set X_{h_1} was chosen to be a private set for v_{h_1}, that is, $S_{h_1} \cap X_{h_1} = \{v_{h_1}\}$. Note that $v_{h_1} \notin X_{h_2}$. Hence, $v_{h_1} \notin Y$, and thus $S_{h_1} \cap Y = \emptyset$. This is a contradiction, which completes the proof of the claim.

By Claim 4.1, $p \leq d!(\frac{\alpha}{\beta}\textsc{opt})^d$. By Lemma 1(i),

$$\omega(v_i) \leq \alpha\, \textsc{opt}(I_i) \leq \alpha\, \textsc{opt}$$

for every $i \in \{1, \ldots, p\}$. Therefore,

$$c(S) = \sum_{i=1}^{p} \omega(v_i) \leq \alpha\, \textsc{opt}\, p \leq \alpha\, \textsc{opt}\, d!(\frac{\alpha}{\beta}\textsc{opt})^d.$$

It follows that $\frac{c(S)}{\textsc{opt}} \leq \alpha d!(\frac{\alpha}{\beta}\textsc{opt})^d$. □

4.2. Independent sets in matroids of maximum and minimum weights

In this subsection, we identify a large class of optimization problems for which the present-biased agent obtains an optimum solution for both $\beta < 1$ and $\beta > 1$. These are the optimization problems solvable exactly by greedy algorithms. The main intuition why the present-biased agent is able to find an optimal solution is that for such problems the greedy algorithm makes an optimal selection at each step. In particular, this means that the present-biased agent does not make incompatible choices. We show that the choice is compatible with an optimum solution.

To define the problem of finding a maximum-weight base of a *matroid*, we need some definitions. We refer to the textbook of Oxley (2011) for the introduction to matroid theory, and we only recall the basics of matroids. A pair $M = (E, \mathcal{I})$, where E is a set called *ground set*, and $\mathcal{I}$ is a family of subsets of E, called *independent sets of* M, is a *matroid* if it satisfies the following conditions, called *independence axioms*:

(I1) $\emptyset \in \mathcal{I}$,
(I2) if $A \subseteq B$ and $B \in \mathcal{I}$ then $A \in \mathcal{I}$,
(I3) if $A, B \in \mathcal{I}$ and $|A| < |B|$, then there is $e \in B \setminus A$ such that $A \cup \{e\} \in \mathcal{I}$.

A set of $\mathcal{I}$ that is maximal for the inclusion is called a *base*. All bases of M have the same cardinality, which is called the *rank* of M. As it is common, we assume that matroids are given by their *independence oracles*, where, given a set $X \subseteq E$, the independence oracle for M answers the query whether X is independent or not in unit time.

The task of the maximum-weight base problem is, given a matroid $M = (E, \mathcal{I})$ and a weight function $\omega\colon E \to \mathbb{Z}$, find a base of the maximum weight. Similarly, the task of the minimum-weight base problem is, given a matroid $M = (E, \mathcal{I})$ and a weight function $\omega\colon E \to \mathbb{Z}$, find a base of the minimum weight.

The greedy algorithm for this problem constructs a base of the input matroid by the iterative addition of a new element to the independent set X which is initially defined as $X := \emptyset$. In each iteration, the algorithm finds an element $e \in E \setminus X$ of maximum weights such that $X \cup \{e\} \in \mathcal{I}$ and updates $X := X \cup \{e\}$. The algorithm stops when it is unable to find an element which can be included in X. Then it returns X. By the classical result of Edmonds (1971), the greedy algorithms compute a base of maximum weight. We show that the present-biased agents obtains an optimal solution in both under and overestimation cases.

Theorem 9. *For every positive β, the present-biased agent obtains an optimal solution for the problem of finding a maximum-weight base of a matroid.*

Proof. Let (M, ω) with $M = (E, \mathcal{I})$ be an instance of the problem of finding a maximum-weight base of a matroid. Observe that the present-biased agent constructs a solution S as follows. Initially, $S := \emptyset$. Then on each iteration, the agent finds a set $X^* \subseteq E \setminus S$ and element $e^* \in X^*$ such that the maximum value of $\omega(e) + \beta\, \omega(X \setminus \{e\})$ is achieved for $X^* = X$ and $e^* = e$, where the maximum is taken over all $X \subseteq E \setminus S$ such that $S \cup X \in \mathcal{I}$. The algorithm stops when the agent fails to find a nonempty $X \subseteq E \setminus S$ satisfying the condition that $S \cup X$ is independent. By axioms (I1)–(I3), S is a base of M. We claim that S is a base of the maximum weight. The claim is straightforward if $\beta = 1$, because the algorithm of the agent is exactly the greedy algorithm. We prove the claim for $\beta \neq 1$.

Underestimation. If $\beta < 1$, then the agent's solution is also constructed by the greedy algorithm. Indeed, given the already constructed partial solution S and $X \subseteq E \setminus S$ such that $S \cup X \in \mathcal{I}$, the maximum value of $\omega(e) + \beta\, \omega(X \setminus \{e\})$ is achieved for $e = \arg\max_{e' \in X} \omega(e')$. Now, if $e^* \in E \setminus S$ is an element of maximum weight such that $S \cup \{e^*\} \in \mathcal{I}$, then, thanks to the result of Edmonds (1971), the following holds. If $Y \subseteq E \setminus (S \cup \{e^*\})$ is a set of maximum weight such that $S \cup \{e^*\} \cup Y \in \mathcal{I}$, then $X = \{e^*\} \cup Y \subseteq E \setminus S$ is a set of maximum weight such that $S \cup X \in \mathcal{I}$. This means that the agent chooses e^* for inclusion in S.

Overestimation. The case $\beta > 1$ is more complicated due to the fact that for S and $X \subseteq E \setminus S$, the maximum value of $\omega(e) + \beta\, \omega(X \setminus \{e\})$ is achieved for $e = \arg\min_{e' \in X} \omega(e')$. Suppose that S is a partial solution of the agent such that S is not a base of M, but $S \subset B$ for some maximum-weight base B. Moreover, assume that the agent chooses $e^* \in E \setminus S$ for inclusion in the solution.

Claim 4.2. *$S \cup \{e^*\}$ is a subset of a base of maximum weight.*

To establish the claim, suppose, for the purpose of contradiction, that $S \cup \{e^*\}$ is not a subset of a base of maximum weight. Then for $X^* \subseteq E \setminus S$ such that $S \cup X^* \in \mathcal{I}$ chosen by the agent, $\omega(S \cup X^*) < W$ where W is the maximum weight of a base. Also we have that $|X^*| \geq 2$ because, for singleton sets $X = \{e\}$, the agent would merely choose an element $e^* = \{X^*\}$ with $\omega(S \cup \{e^*\}) = W$. By Edmonds (1971), there is $Y \subseteq E \setminus S$ such that $S \cup Y$ is a base of maximum weight W, where Y is constructed by the greedy algorithm. Since $\omega(S \cup X^*) < W$, we have $\omega(Y) > \omega(X^*)$. Let $Y = \{e_1, \ldots, e_k\}$, where the elements are ordered by their inclusion in Y during the execution of the greedy algorithm (e_i is included before e_{i+1}). Note that $\omega(e_1) \geq \cdots \geq \omega(e_k)$, that is, e_k is an element of minimum weight in Y. Since $|X^*| \geq 2$, we have $k \geq 2$. By the properties of the greedy algorithm (see Edmonds, 1971), the set $Z = \{e_1, \ldots, e_{k-1}\}$ is a subset of maximum weight among of all subsets $Z' \subseteq E \setminus S$ such that $S \cup Z' \in \mathcal{I}$ and $|Z'| = k-1$. Since $|X^* \setminus \{e^*\}| = k - 1$, we have $\omega(X^* \setminus \{e^*\}) \leq \omega(Z)$. It follows that

$$\begin{aligned}
\omega(e^*) + \beta\, \omega(X^* \setminus \{e^*\}) &= \omega(X^*) - \omega(X^* \setminus \{e^*\}) + \beta\, \omega(X^* \setminus \{e^*\}) \\
&= \omega(X^*) + (\beta - 1)\, \omega(X^* \setminus \{e^*\}) \\
&< \omega(Y) + (\beta - 1)\, \omega(Z) \\
&= \omega(Y) + (\beta - 1)\, \omega(Y \setminus \{e_k\}) \\
&= \omega(e_k) + \beta\, \omega(Y \setminus \{e_k\}).
\end{aligned}$$

This strict inequality contradicts the strategy of the agent who should prefer Y and e_k over X^* and e^*. This proves the claim.

Applying the claim iteratively, we conclude that the agent should select a base of maximum weight. □

Finally, let us remark that because we assume that the weights of the elements of the matroid are from $\mathbb{Z}$, Theorem 9 (by multiplying all weights by -1) also implies that the present-biased agent obtains an optimal solution for the problem of finding a minimum-weight base of a matroid.

5. Conclusion

We introduced a framework for time-inconsistent planning. Such a framework enables to perform fine-grained analysis of the behavior of present-biased agents depending on the optimization problem they have to solve. Our case study concerns two optimization problems: set cover and hitting set. It would be fascinating to provide a fine-grained analysis for other fundamental optimization problems (NP-hard and polynomial-time solvable). The incomplete list of open questions includes: matchings (perfect, maximum, stable, etc.), integer linear programming, max-flow, knapsack, traveling salesman, and max-cut. More generally, it would be fascinating to obtain a deeper understanding of the impact of optimization problems on the worst case ratio of present-biased agents.

When an agent is required to solve an NP-hard problem, it is natural to assume that some heuristics or approximation algorithms are used due to computational limitations. Thus, another exciting direction for further research is to investigate the influence of approximation on present-biased agents' behavior. In our work, we made only the first steps in this direction. A very concrete example: one can adapt the upper bound on the cost ratio in Theorem 4 to the assumption that the agent uses an α-approximation algorithm. However, the bound blows-up by the factor α^s, where s is the size of the solution obtained by the agent (informally, we pay the factor α on each iteration). However, the examples in Section 4 show that this is not always the case. Are there cases when this exponential blow-up unavoidable?

References

Akerlof, G.A., 1991. Procrastination and obedience. Am. Econ. Rev. Pap. Proc. 81 (2), 1–19.

Albers, S., Kraft, D., 2017a. On the value of penalties in time-inconsistent planning. In: 44th International Colloquium on Automata, Languages, and Programming, ICALP. pp. 10:1–10:12.

Albers, S., Kraft, D., 2017b. The price of uncertainty in present-biased planning. In: 13th Int. Conference on Web and Internet Economics, WINE. pp. 325–339.

Albers, S., Kraft, D., 2019. Motivating time-inconsistent agents: A computational approach. Theory Comput. Syst. 63 (3), 466–487.

Cygan, M., Fomin, F.V., Kowalik, L., Lokshtanov, D., Marx, D., Pilipczuk, M., Pilipczuk, M., Saurabh, S., 2015. Parameterized Algorithms. Springer.

Edmonds, J.R., 1971. Matroids and the greedy algorithm. Math. Program. 1 (1), 127–136. http://dx.doi.org/10.1007/BF01584082.

Erdős, P., Rado, R., 1960. Intersection theorems for systems of sets. J. Lond. Math. Soc. 35, 85–90.

Fomin, F.V., Strømme, T.J.F., 2020. Time-inconsistent planning: Simple motivation is hard to find. In: Proceeding of the 34th AAAI Conference on Artificial Intelligence, AAAI. AAAI Press, pp. 9843–9850, URL https://aaai.org/ojs/index.php/AAAI/article/view/6537.

Gravin, N., Immorlica, N., Lucier, B., Pountourakis, E., 2016a. Procrastination with variable present bias. In: ACM Conference on Economics and Computation, EC. pp. 361.

Gravin, N., Immorlica, N., Lucier, B., Pountourakis, E., 2016b. Procrastination with variable present bias. CoRR, arXiv:1606.03062.

Jepson, C., Loewenstein, G., Ubel, P., 2001. Actual versus estimated differences in quality of life before and after renal transplant. In: Working Paper. Department of Social and Decision Sciences, Carnegie Mellon University.

Kleinberg, J.M., Oren, S., 2014. Time-inconsistent planning: A computational problem in behavioral economics. In: ACM Conference on Economics and Computation, EC. pp. 547–564.

Kleinberg, J.M., Oren, S., 2018. Time-inconsistent planning: A computational problem in behavioral economics. Commun. ACM 61 (3), 99–107.

Kleinberg, J.M., Oren, S., Raghavan, M., 2016. Planning problems for sophisticated agents with present bias. In: ACM Conference on Economics and Computation, EC. pp. 343–360.

Kleinberg, J.M., Oren, S., Raghavan, M., 2017. Planning with multiple biases. In: ACM Conference on Economics and Computation, EC. pp. 567–584.

Loewenstein, G., O'Donoghue, T., Rabin, M., 2003. Projection bias in predicting future utility. Q. J. Econ. 118 (4), 1209–1248.

Oxley, J., 2011. Matroid theory, second ed. In: Oxford Graduate Texts in Mathematics, vol. 21, Oxford University Press, Oxford, p. xiv+684. http://dx.doi.org/10.1093/acprof:oso/9780198566946.001.0001.

Tang, P., Teng, Y., Wang, Z., Xiao, S., Xu, Y., 2017. Computational issues in time-inconsistent planning. In: Proceedings of the 31st Conference on Artificial Intelligence. AAAI, AAAI Press.