Incentive Design for Present-Biased Agents

A Computational Problem in Behavioral Economics

by Dennis Kraft

Technische Universität München
Fakultät für Informatik

Fakultät für Informatik
Lehrstuhl für Algorithmen und Komplexität

Incentive Design for Present-Biased Agents

A Computational Problem in Behavioral Economics

Dennis R. R. Kraft

Vollständiger Abdruck der von der Fakultät für Informatik der Technischen Universität München zur Erlangung des akademischen Grades eines Doktors der Naturwissenschaften genehmigten Dissertation.

Vorsitzender: Prof. Dr. Francisco Javier Esparza Estaun

Prüfer der Dissertation:
1. Prof. Dr. Susanne Albers
2. Prof. Dr. Martin Bichler

Die Dissertation wurde am 27.06.2018 bei der Technischen Universität München eingereicht und duch die Fakultät für Informatik am 30.10.2018 angenommen.

Abstract

The tendency to underestimate future costs and rewards is a wide spread cognitive bias that may severely impair a person's ability to make consistent plans over an extended period of time. People who exhibit such a present-bias are prone to inefficient behavioral patters such as procrastination and the abandonment of partially completed work. As a result, they may require external incentives to reach certain goals. Drawing on a recent graphical model due to Kleinberg and Oren [16], we approach the design of such incentives from an algorithmic perspective. In the first part of this work we consider three commonly used incentives that are based on prohibition, penalties and rewards. We are particularly interested in comparing the conceptual costs of implementing these incentives as well as analyzing the complexity of computing optimal and approximately optimal designs. The presented results summarize and extend our previous work on computing efficient incentives [2, 3]. In the second part of this work we turn our attention to two generalized versions of Kleinberg and Oren's graphical model. The first one addresses incentives for multiple people of a heterogeneous population while the second one is concerned with incentives for people whose present-bias varies over time. Our goal is to quantify the conceptual cost that penalty based incentives incur due to these changes in the setting. Furthermore, we study the complexity of computing optimal or approximately optimal designs in both models. The obtained results have been published in our work on incentive design with imperfect information [4].

Acknowledgments

I would like to express my sincere gratitude to my adviser Prof. Dr. Susanne Albers, who gave me the opportunity to peruse my research interests. I deeply value her trust and support throughout my time as a doctoral student. I am also very grateful to my second adviser Prof. Dr. Martin Bichler for reviewing this thesis and mentoring me on my academic work. Furthermore, I want to thank Prof. Dr. Javier Esparza for taking the time to hold the chair of the examination committee. I am particularly indebted to Angelika Freisberg and Manfred Freisberg for their tireless proofreading. The quality of this work has improved a lot due to their hard work. I also want to extend my thanks to Jasmin Kraft and Glenn Watamanik, who proofread one of the publications leading up to this work. My special thanks go to Dr. Yiannis Giannakopoulos, Marinus Gottschau, Prof. Dr. Harald Räcke, Sebastian Schraink, Dr. Suzanne van der Ster, Richard Stotz, Jeremias Weihmann and all my other colleagues who made the last years an enjoyable and rewarding experience. Finally, I want to thank my family, my partner and my friends for their patience, love and support.

List of Symbols

β	present-bias
$\beta(v)$	present-bias assigned to a node v by the present-bias configuration β
$\mu^*(G, \beta)$	minimum reward or budget for which G admits a motivating subgraph, cost configuration or reward configuration
$\mu(G, \beta)$	minimal reward or budget for which G is motivating
τ	range of the present-bias set B
B	set of present-bias values
$c(e)$	cost of an edge e
$\tilde{c}(e, \beta)$	perceived cost of an edge e
$\tilde{c}(P, \beta)$	perceived cost of a path P
$\tilde{c}(v, \beta)$	perceived cost of a node v
$d(v)$	cost of a cheapest path from a node v to the terminal node t
E	set of edges of G
e	variable denoting an edge of G
F	subset of edges of G
G	task graph
$h(e)$	extra cost cost assigned to the edge e by the cost configuration h
i, j, k, ℓ	numerical indices
n	number of nodes in G
P, Q, R	variables denoting a path within G
$q(G_r, \beta)$	maximum reward collected in the task graph G_r
r	reward awarded at t
$r(v)$	reward awarded at a node v by the reward configuration r
s, t	terminal nodes of G
u, v, w, z	variables denoting nodes of G
V	set of nodes of G
W	subset of nodes of G

Contents

1 Introduction

Saving up money, writing a term paper and losing weight are just a few of many resolutions that people make every day. But despite their best intentions they often find it difficult to see their resolutions through: they squander money, delay work and indulge in unhealthy food. The underlying pattern of behavior like this is generally the same. In order to reach some long-term goal, people plan a certain course of action that they believe maximizes their current and future utility. However, as time goes by, other courses of action are becoming more appealing and the original plan is eventually changed. Sometimes this is the result of unforeseen circumstances. But people also tend to deviate from their original plan even if the circumstances stay the same. In this case, their change of mind often leads to seemingly irrational behavior such as procrastination and the abandonment of partially completed work. It is needless to say that behavioral patterns like these may greatly affect a person's own welfare as well as the welfare of others. Moreover, this type of behavior challenges traditional economic models, which assume that a person's preferences stay consistent over time.

1.1 The Post Office

With the advancement of behavioral economics, insight from psychology and related fields has been incorporated into economic theory to better understand the causes and effects of human behavior. As a result of this process, a behavioral model has been put forward that offers a particularly simple yet compelling explanation for the phenomenon of time-inconsistent behavior. The underlying idea of this model is that people assign disproportionately greater value to the present than to the future. This cognitive bias is also called the *present-bias*. To illustrate how time-inconsistent behavior emerges naturally as the result of a present-biased perception, we have a brief look at a story due to Akerlof [1]. Note that we slightly adapt the story for the sake of a more coherent discussion within the context of this work.

Imagine a person named Alice who is expecting a package. Once the package has arrived at the local post office, Alice has 10 days time to pick it up before it is returned

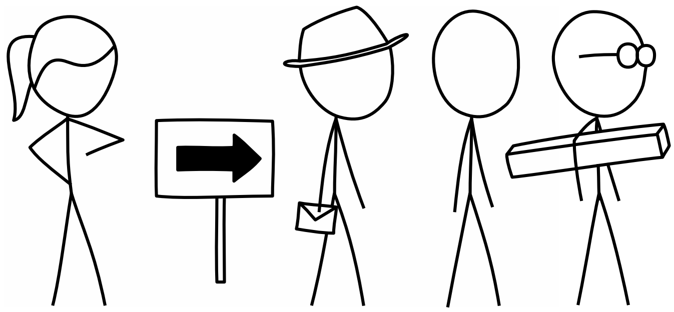

Figure 1.1: Alice waiting in line at the post office

to the sender. Since the package is important to her, assume that Alice pays a price of c for each day that she goes without its content. However, due to the long waiting lines at the post office, which are equally dreadful each day of the week, Alice also has to invest a considerable one-time effort of $c' > c$ to get the package. Figure 1.1 shows an illustration of Alice's plight at the post office.

It seems like the best course of action in Akerlof's story is to collect the package as soon as possible. After all, Alice incurs a cost of c' regardless of the day she goes to the post office. However, if she goes on the day that the package arrives, she pays none of the additional cost of c that becomes due each day after. This simple analysis suggests that Alice would be eager to collect the package right away. However, experience shows that people who face similar situations often behave differently and delay costly tasks even if this delay results in unnecessary cost.

To understand this phenomenon, assume that Alice is present-biased in the sense that she evaluates current costs accurately, but discounts future costs by a factor of $\beta \in (0, 1]$. For the sake of our discussion, let $\beta < 1 - c/c'$ and recall that $c' > c$. The latter assumption ensures that a feasible value of β exists. Now assume that the package has just arrived at the post office. If Alice decides to collect it right away, she incurs an immediate cost of c'. In contrast, if she decides to collect it on the next day, she only incurs an immediate cost of c and anticipates a future cost of $\beta \cdot c'$. Thus, she believes that the cost of running the errand on the current day is c' whereas the cost of running the errand on the next day is $c + \beta \cdot c'$. By choice of β, the second option appears to be strictly less costly to Alice, i.e.,

$$c + \beta \cdot c' < c + \big(1 - \frac{c}{c'}\big) \cdot c' = c'.$$

As a result of this inequality, she decides to wait for a short while and collect the package on the following day.

However, before Alice heads to the post office the next day, she takes the time to briefly reevaluate her plan. Similar to the previous day, she believes that the cost of running the errand immediately is c' while the cost of waiting for another day is $c+\beta\cdot c'$. Since this is the same situation as before, Alice changes her mind and decides to collect the package one day later. Note the inconsistency of this behavior. On the first day Alice does not go to the post office with the good intention to pick up the package one the following day, but then she changes her plan and decides to wait again. The repeated application of this argument implies that Alice fails to pick up the package at any of the first 9 days. The reason is that waiting always seems to be the more preferable option to her. Only on the very last day, when she cannot delay the errand anymore, must she go to the post office. It should be stressed that Alice does not anticipate this behavior, nor does she procrastinate on purpose. Nevertheless, an accumulation of many time-inconsistent choices, each one rather insignificant on its own, eventually causes her to follow the worst possible course of action.

1.2 Temporal Discounting

One of the main difficulties that Alice faces in Akerlof's story is the trade-off between costs and benefits that occur at different points in time. When assessing intertemporal choices like this, economists generally assume that people attach less weight to distant events than to imminent ones. To specify this notion of *temporal discounting* formally, let $u(e)$ be the utility of a certain event e. Furthermore, let $\tilde{u}(e,t)$ be the utility that a person assigns to e whenever e occurs t time units in the future. If the person evaluates $\tilde{u}(e,t)$ according to some form of temporal discounting, then this utility can be expressed as $\tilde{u}(e,t) = D(t)\cdot u(e)$, where $D(t)$ denotes a *discount factor* that decreases monotonically over times. Depending on the particular form of the *discount rate* $D(t)/D(t+1)-1$, economists distinguish between two models of temporal discounting.

In traditional economics the prevailing model is based on the assumption that the discount rate is a constant value $\rho \in [0,\infty)$, i.e., $D(t)/D(t+1)-1=\rho$. Clearly, this assumption implies that the discount factor $D(t)$ decreases exponentially over time. More precisely, it implies that $D(t) = (1/(1+\rho))^t$. The corresponding model of temporal discounting is therefore also known as *exponential discounting*. It can be stated that much of the appeal of exponential discounting is due to the conceptual elegance and mathematical simplicity that go along with a constant discount rate. But despite the popularity of the model, a staggering amount of empirical research casts serious doubts on its descriptive validity. A comprehensive survey on this topic has been compiled by Frederick, Loewenstein and O'Donoghu [11]. To highlight just one discrepancy between the theoretical predictions of exponential discounting and actual human behavior, consider the phenomenon of *preference reversal*. Under certain circumstances, it is well-documented that people change their preference with respect to two events if those events are delayed by a constant amount of time, see e.g. [14]. However, such observations are incompatible with exponential discounting. To see this, assume that a person prefers some event e at time t over another event e' at time t', i.e., $\tilde{u}(e,t) \geq \tilde{u}(e',t')$. In the exponential discounting framework, this immediately implies that the same person also prefers e over e' if both events are delayed by q time units. The reason is that

$$\tilde{u}(e,t+q) = \Big(\frac{1}{1+\rho}\Big)^q \cdot \tilde{u}(e,t) \geq \Big(\frac{1}{1+\rho}\Big)^q \cdot \tilde{u}(e',t') = \tilde{u}(e',t'+q).$$

Clearly, this cannot explain preference reversal. But if exponential discounting fails to predict such a fundamental phenomenon, it certainly cannot shed light on more complex instances of time-inconsistent behavior either.

Due to this shortcoming of exponential discounting, behavioral economists have advocated an alternative model of temporal discounting. Their argument is based on experimental research indicating that people do not only exhibit a monotonically decreasing discount factor, but also a monotonically decreasing discount rate, see e.g. [24]. Temporal discounting that incorporates this observation is called *hyperbolic discounting* because the function obtained from interpolating empirically collected data generally resemble a hyperbolic function. The discount factor $D(t)$ is therefore often defined as $D(t) = 1/(1+\alpha \cdot t)$ for some fixed parameter $\alpha \in [0,\infty)$. It is easy to demonstrate that this type of hyperbolic-discounting indeed predicts a preference reversal in some settings. Consider for instance a person with discount parameter $\alpha = 1$ who needs to choose between two events e and e' with a respective utility of $u(e) = 2$ and $u(e) = 3$. If e occurs at time $t = 0$ and e' occurs at time $t' = 1$, then this person clearly prefers e over e' considering that $\tilde{u}(e,0) = 2 > 3/2 = \tilde{u}(e',1)$. Nevertheless, a delay of 2 time unites reverses this preference since $\tilde{u}(e,2) = 2/3 < 3/4 = \tilde{u}(e',3)$. The intuitive reason for the change of mind is that the discount rate at time $t = 0$ is higher than the discount rate at time $t = 2$. Thus, the person is more sensitive to the time interval between e and e' in the near future than to the same time interval in the far future.

Empirical comparisons between exponential discounting and hyperbolic discounting generally find hyperbolic discounting to be the descriptively superior model, see e.g. [15]. However, the mathematical properties of hyperbolic functions make this framework impractical for a theoretical analysis of complex scenarios. In an attempt to find a compro-

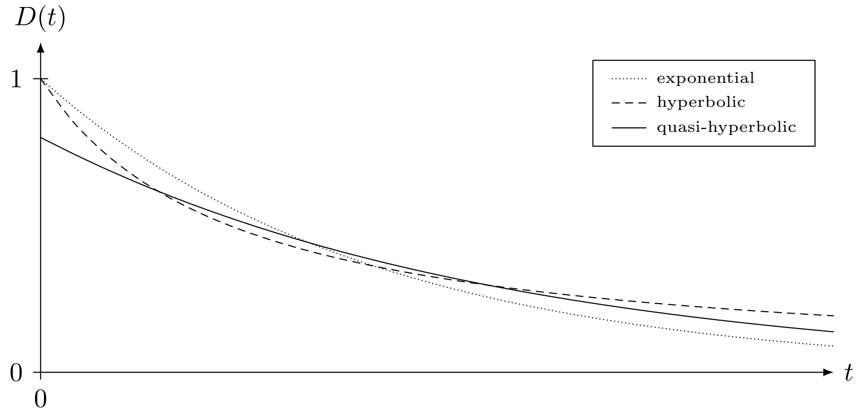

Figure 1.2: Different types of discount factors decreasing over time

mise between descriptiveness and practicality, Laibson has forwarded the idea of *quasi-hyperbolic discounting* [19], which since then has become a standard model of temporal discounting in the field of behavioral economics. To keep the model simple, Laibson approximates human discount rates with a constant $\rho \in [0, \infty)$ at any time $t > 0$. The only exception is at the time $t = 0$ when time the discount rate might be higher. The magnitude of this divination is quantified by the parameter $\beta \in (0, 1]$. More precisely, the discount factor $D(t)$ is defined as $D(t) = \beta \cdot (1/1 + \rho)^t$ whenever $t > 0$ and $D(t) = 1$ otherwise, i.e., $t = 0$. Figure 1.2 depicts the resulting valuations of $D(t)$ for some arbitrary choice of β and ρ and compares them to other forms of hyperbolic discounting and exponential discounting. It is particularly interesting to note the discontinuity of $D(t)$ at $t = 0$ in the quasi-hyperbolic model. This distinctive feature of quasi-hyperbolic discounting can be interpreted as a cognitive bias against immediate cost and toward immediate benefits. If this bias seems familiar, it is because Akerlof's model of temporal discounting is based on a similar idea. In fact, the model is a special case of quasi-hyperbolic discounting that sets $\rho = 0$. As Akerlof's story demonstrates, even this most basic form of quasi-hyperbolic discounting gives rise to relevant behavioral phenomena such as procrastination. This makes quasi-hyperbolic discounting an expressive but also convenient framework for theoretical research.

1.3 The Graphical Model

So far we have given a brief overview of how hyperbolic discounting in general and quasi-hyperbolic discounting in particular may cause people to act in a certain way that is at odds with traditional economic theory. But although an extensive amount of research in behavioral economics is concerned with the effects hyperbolic discounting, most of the obtained results require additional and often quite intricate modeling effort. To unify these results, Kleinberg and Oren propose an elegant graph theoretical model in which many of the behavioral anomalies generally attributed to hyperbolic discounting emerge naturally as different instances of a single computational problem [16].

A formal introduction to Kleinberg and Oren's model is presented in Chapter 2. At this point we only sketch the high level ideas. Kleinberg and Oren's graphical model is essentially a planning problem in which a single agent constructs a path through a directed acyclic graph G. The nodes of G, which include a source node s and a target node t, represent intermediate states of progress toward a certain long-term goal. To move from one node to the next, the agent must complete an atomic task that is associated with the corresponding edge of the graph. Task graphs like this are routinely used in various fields of computer science to model planning and scheduling problems. Due to their expressive power, they also seem to be a suitable tool to model many of the decision environments that behavioral economists are interested in.

While navigating the task graph G, the agent moves forward according to the following simple iterative procedure: When located at a certain node v, she evaluates the cost of all paths from v to t and chooses whichever path appears to be the cheapest. But since she is present-biased, she only evaluates the cost of the outgoing edges of v correctly. All subsequent edges represent future tasks and are therefore discounted by the agent's present-bias β. Once the agent has settled for a certain path $P = v, w, \ldots, t$, she crosses the first edge of P and moves onto w where she reevaluates her plan. For this purpose, she calculates the cost exactly the same way as she did at v. The only difference is that she is now located at w, which may change her perception of some of the edges.

It is easy to see that the agent's decision making process closely resembles Akerlof's behavioral model from Section 1.1. By using Akerlof's ideas, Kleinberg and Oren make some implicit assumptions on the nature of the agent, which deserve further discussion. Consider for instance the way in which the agent evaluates the utility of paths in the task graph. The fact that she perceives immediate edge costs accurately and discounts future edge costs by a factor of β implies that the graphical model is based on a special case of quasi-hyperbolic discounting where the discount rate is set to $\rho = 0$. By shifting the focus onto the agent's present-bias, Kleinberg and Oren settle for a compromise between descriptively validity and formal simplicity. But even though a compromise like this is common in theoretical work, reducing a person's decision making process to a single cognitive bias is of course a gross simplification. To obtain more comprehensive behavioral models, economists distinguish between a wide variety of cognitive biases, see e.g. [9]. It should be noted that, apart from the present-bias, the standard version of the graphical model does not address any other cognitive bias directly. However, an interesting line of research by Kleinberg, Oren and Raghavan incorporates additional biases into the graphical model [18].

Another fundamental assumption of the graphical model relates to the way in which the agent chooses which edge to cross after evaluating all utilities. Recall that Kleinberg and Oren assume that the agent greedily picks an edge that lies on a path that she momentarily believes is a cheapest path to t. Clearly, this decision can only be justified if the agent is unaware of her own present-bias. People who are prone to such a lack of self-awareness are called *naive*, whereas people who are fully aware of their present-bias and plan accordingly are called *sophisticated*. Sometimes behavioral economists also consider a type of person who is aware of her present-bias, but misjudge its extent, see e.g. [22]. Such people are called *partially sophisticated*. Extensive empirical and experimental research suggests that naive reasoning and sophisticated reasoning both play important roles in different contexts of everyday life [9, 11]. However, the agent's decisions process in the Kleinberg and Oren's graphical model only captures naive behavior. An extension to sophisticated and partially-sophisticated agents has been studied by Kleinberg, Oren and Raghavan [17].

We want to underline that our work is primarily based on the original version of the graphical model. In particular, we strictly adhere to Akerlof's assumptions on present-biased behavior. This means we assume that the perception of the agent is affected only by her present-bias and that her decisions are the result of a naive maximization of her currently perceived utility.

1.4 Incentive Design

A recent line of research uses Kleinberg and Oren's graphical model to quantify the effects of present-biased behavior on a person's performance in social and economic settings, see e.g. [13, 16, 17, 18, 23]. In the case of Akerlof's scenario, these effects are widely apparent. For instance, recall that the protagonist Alice repeatedly delays the task of collecting a package from the post office. Since Alice suffers a cost of c for each of the 9 days on which she fails to collect the package, she eventually incurs an unnecessary total cost of $9 \cdot c$. In a slightly modified scenario, the avoidable cost may become even greater, growing exponentially in the length of the delay [16, 23].

Considering the loss of efficiency caused by present-biased choices, economists often try to encourage sensible behavior by adjusting the external circumstance of a given scenario. We call such adjustments *incentives*. To illustrate the idea, assume that the post office in Akerlof's scenario wants people to collect their packages in a more timely manner and reduces the number of days that it stores parcels from 10 to just 1. As a result of the new policy, Alice has no choice but to collect her package right away. Clearly, this sense of urgency is not only beneficial to the post office, but also to Alice since she saves the cost of procrastinating. Deadlines like the one imposed by the post office are a canonical example of how incentives can help people to reach their goals in a more effective way, see e.g. [5]. Other examples of potentially beneficial incentives that are often considered in the economic literature include strategically implemented rewards and penalties [7, 21].

Depending on the particular scenario, some types of incentives are usually more suitable than others. However, to assess the quality of a particular incentive it is important to take its *designer's* objective into account. Behavioral economists generally distinguish between two different types of designers. On the one hand are *self-interested* designers who try to maximize their personal utility. Incentives implemented by such designers include performance bonuses that companies promise their employees to increase productivity or payment plans that retailers provide to attract more customers. Note that these incentives are sometimes at the expense of present-biased people if this benefits the designer. *Benevolent* designers on the other hand are first and foremost interested in the welfare of the people they care for. A common example of incentives implemented by benevolent designers are government policies that try to help people reduce their use of harmful substances such as tobacco, alcohol and drugs.

Drawing on the work of Kleinberg and Oren, we assume that the primary objective of a designer is to ensure that a given person reaches a predefined goal [16]. Note that this objective is usually held by self-interested and benevolent designers alike, although their motives are different. The problem faced by the designer is that her objective is not necessarily the same as that of the person in question. To capture the potential discrepancy, it is sensible to assume that the person is motivated by some form of reward, which she receives upon completing the goal. Kleinberg and Oren capture this idea in the graphical model by assigning a non-negative reward r to the target node t. Furthermore, they assume that the person is free to quit whenever she believes that the reward does not cover her cost. In a setting like this, the designer is inevitably interested in minimizing the person's maximum perceived cost. Otherwise, the designer cannot ensure that the person remains motivated long enough to reach the goal. Note that this objective is different from simply minimizing the total cost experienced by the person. The intuitive reason why the two objectives do not coincide is that under certain circumstances suboptimal behavior such as procrastination keeps a present-biased individual motivated while more cost effective behavior does not. An example of this seemingly paradoxical phenomenon can be found in Section 4.1.

1.5 Our Work

The economic literature is rich in examples of incentives that address certain forms of present-biased behavior. However, most of these examples are limited to very specific scenarios that often require an intricate modeling effort. Based on Kleinberg and Oren's graphical model [16], we present a novel and unifying approach to incentive design incorporating various ideas from the field of algorithms and theoretical computer science. Our main focus is on quantifying the conceptual efficiency of different types of incentives and analyzing the computational challenges of their design.

We first turn our attention to incentives that are based on the prohibition of strategically selected courses of action. A common example of this type of incentive is deadlines that prohibit actions if they violate a predefined time limit. However, more complex incentives may prohibit an arbitrary selection of actions. Kleinberg and Oren capture such prohibition structures in the graphical model by removing the edges of forbidden actions from the task graph G. Based on this framework, they observe that some scenarios admit prohibition structures that improve the agent's performance exponentially with respect to the size of G. More precisely, the performance improves within a factor of β^{-n+2}, where n denotes the number of nodes of G. On top of this conceptual result Kleinberg and Oren raise the algorithmic question of identifying an optional subset of edges to remove from G.

In Chapter 3 we prove that Kleinberg and Oren's observation about the efficiency of prohibition based incentives is tight and argue that the computation of an optimal prohibitions is NP-hard [2]. It should be noted that Tang et al. obtain a similar NP-hardness result [23]. However, our result holds true even if the agent's behavior with respect to an optional prohibition structure is known in advance. This curious property of prohibition based incentives turns out to be a key challenge in the design of efficient solutions. In fact our NP-hardness result can be generalized to prove that any approximation of an optimal prohibition structure within a factor less than $\sqrt{n}/3$ of the optimum is NP-hard as well [2]. We complement this theoretical upper bound on the approximability of prohibition based incentives with a polynomial time algorithm that approximates an optimal solution within a factor of $1 + \sqrt{n}$ [2]. This settles the approximability of prohibition based incentives up to a small constant factor.

To avoid the algorithmic limitations of prohibitions, we introduce a new type of incentive to the graphical model in Chapter 4. The idea is to make certain actions less attractive by imposing artificial penalties. Similar to prohibition such penalty based incentive are frequently used in practice. Consider for instance a library that charges late fees to ensure the timely return of its books or a government that imposes fuel taxes to reduce its citizen's carbon footprints. We model such incentives by assigning non-negative extra cost to the edges of G. This cost can be thought of as a penalty fee that the agent incurs when traversing the corresponding edge. However, we do not assume that the designer benefits directly from this fee. More precisely, we assume that the designer is not interested in maximizing the total amount of penalty fees that the agent accumulates. Instead, her objective remains the same as in the previous setting, namely to keep the agent motivated for as little reward as possible. A conceptual analysis of penalty fees shows that they realize this objective at least as efficiently as prohibition structures do and up to $1/\beta$ times more efficiently in some scenarios [3]. This result is tight. Taking a closer look at the computational properties of penalty fees, we are able to prove that the construction of an optimal allocation of fees is NP-hard [3]. However, in contrast to prohibition based incentives, it is possible to design an almost optimal allocation in polynomial time if the agent's behavior with respect to an optimal solution is known in advance [3]. This computational advantage is also reflected in the 2-approximation algorithm that we present for the design of an efficient allocation of penalty fees [3]. A generalization of our NP-hardness result furthermore yields an asymptotically matching lower bound of 1.08192 on the approximability of the problem [3].

In Chapter 5, we study incentives that are based on intermediate rewards. Their purpose is to increase the appeal of certain transitional stages of a long-term project. Kleinberg and Oren propose to model this type of incentive by assigning non-negative rewards to arbitrary nodes of G. However, they do not investigate the resulting reward allocations in detail. Instead, they pose the formal analysis of reward based incentives as an open research problem [16].

A key observation that should be kept in mind when considering reward based incentives is that present-biased people are occasionally motivated by rewards that they never claim. Assuming that the designer does not have to pay for such rewards, she may use this phenomenon to her advantage and get the agent to perform work for free. Under certain circumstance the designer may even motivate the agent to traverse the entire task graph G without incurring any cost. Note that *exploitative* rewards like this are impossible if the agent is only offered a single reward upon completion of G. Based on these considerations, we argue that the cost efficiency of reward based incentives may exceed the cost efficiency of penalty based incentives by an unbounded factor, at least from the designer's perspective. Conversely, there exist scenarios in which reward based incentives are β^{-n+2} times less cost efficient than penalty based incentives. These results are tight. Apart from the conceptual implications of exploitative rewards there are also some computational ones. A particular curious consequence of exploitative rewards is that it is NP-hard to decide whether a given scenario admits a reward allocation that keeps the agent motivated for free [16]. As a result, it is not only NP-hard to construct optimal reward allocations but it is also NP-hard to approximate them within a bounded factor. Surprisingly, the computational challenges of constructing efficient prohibitions, penalty fees and intermediate rewards immediately disappear if we combine them in a single incentive. We conclude the first part of our work by showing that a combined incentive admits optimal designs in polynomial-time. It should be noted that this result as well as most of the other results about intermediate rewards depend crucially on the assumption that the designer does not have to pay for uncollected rewards. A different setting in which the designer incurs cost even if the agent does not claim a reward has been investigated by Tang et al. [23].

In the second part of this work we set the formal evaluation of different types of incentives aside and turn toward two variations of the graphical model. We lay our main focus on how these variations impact the design of efficient incentives. In particular, we are interested in their impact on penalty fees. The reason why we choose penalty fees rather than prohibition or reward based incentives is partly because of their conceptual power and partly because of their favorable computational properties.

The first variation of the graphical model addresses a research problem proposed by Kleinberg and Oren concerning the design of incentives for a population of present-biased people rather than a single individual [16]. The difficulty of this problem is that the members of the population may have different present-biases. We model this notion of *heterogeneity* by defining a set $B \subset (0, 1]$ that contains each present-bias of the population. The goal is to design an incentive that motives a given member of the population independent of the corresponding present-bias. Settings like this are particularly interesting if individual incentives are not admissible due to equal treatment directives or simply impractical due to the size of the population. Another reason for designing incentives for multiple present-biases arises if precise information about a certain present-bias is unavailable. A fundamental question in this context is whether incentives that are motivating for all present-biases of the set B are inherently more costly than incentives that address only a single present-bias of the set B. We call the qualitative ratio between the two incentives the *price of heterogeneity*.

In Chapter 6 we demonstrate that penalty fees may exhibit a price of heterogeneity that is greater than 1. More precisely, we construct a family of scenarios in which this ratio converges to 1.1 [4]. This implies that a true loss of efficiency is sometimes unavoidable. However, the price of heterogeneity may not become arbitrarily large. To prove this remarkable fact, we revisit the 2-approximation algorithm we have originally designed for a single person. A careful adaptation of the algorithm to multiple individuals does not only retain the approximation factor of two 2, but also yields an allocation of penalty fees that bounds the price of heterogeneity by the same amount, i.e., 2 [4].

The second variation of the graphical model is inspired by the work of Gravin et al. and assumes that the present-bias of a person is not a fixed value, but varies over time [13]. One way to interpret such variations is as artifacts of other cognitive biases that are not directly accounted for in the graphical model. Attempting to keep people motivated despite a varying present-bias may therefore yield more robust incentives. However, the additional robustness may come at the cost of a decreased efficiency. To quantify this loss, we introduce the *the price of variability*. Similar to the price of heterogeneity, this measure reflects the qualitative ratio between incentives that are robust with respect to a variable present-bias and incentives that are designed for a fixed present-bias. Note that this definition assumes that all present-biases are restricted to a predefined set $B \subset (0, 1]$ of potential values.

The close resemblance between the price of variability and the price of heterogeneity may raise the question whether the former can be bounded by a constant factor as well. In Chapter 7 we refute this conjecture for penalty based incentives. Instead, we present evidence that the price of variability is closely connected to the relative range $\tau = \max B / \min B$ of the set B. For this purpose we construct a family of scenarios in which the price of variability approaches $\tau/2$ as τ goes to infinity [4]. A complementing upper bound of $\tau+1$ can be deduced from a further generalization of the 2-approximation algorithm of the previous chapter [4]. Similar considerations also yield a matching approximation ratio of $\tau + 1$ for the problem of optimizing robust incentives [4]. Of course there remains a wide gap between $\tau + 1$ and the NP-hard approximation ratio of 1.08192 implied by our earlier work. To close this gap to some extent, we prove that no constant approximation is possible in polynomial time unless NP = ZPP [4]. Unfortunately, this result implies that it is generally more difficult to work with a variable present-bias than with a fixed present-bias. However, there is one striking exception to this rule. We end our work with this positive exception and present a polynomial time algorithm that constructs optimal and robust penalty fees whenever B contains the value 1, i.e., the agent occasionally loses her present-bias [4].

2 The Formal Framework

The graphical model of Kleinberg and Oren [16] provides a general framework to analyze and quantify the behavior of present-biased agents. This lays the formal foundation of our work. We therefore use the following chapter to thoroughly introduce this framework and settle some of its fundamental computational properties.

2.1 The Graphical Model

Consider an agent working towards some long-term goal and assume that her progress is tracked via the n nodes of a directed acyclic graph that is induced by the node set V and edge set E. Furthermore, assume that V includes a designated source node s and target node t. To move from one node v to another node w, the agent must complete the task associated with the edge (v, w). The corresponding effort is captured by a non-negative edge cost $c(v, w) \geq 0$. Once the agent reaches t, she receives a non-negative reward $r \geq 0$ as compensation for her previous expenses. We call this graphical representation of a given scenario the *task graph* and denote it by $G = (V, E, c, r)$.

Definition 2.1.1 (Task Graph)**.** The task graph $G = (V, E, c, r)$ is a directed acyclic graph with a pair of terminal nodes $s, t \in V$, a non-negative edge cost $c : E \to \mathbb{R}_{\geq 0}$ and a non-negative reward $r \geq 0$.

On her way through G, the agent moves forward according to the following procedure: Located at a node $v \neq t$, she tries to find a cheapest path to t. However, only the initial edge (v, w) of any path $P = v, w, \ldots, t$ must be paid immediately; all other edges are charged at a later point in time. Being prone to temporal discounting, the agent scales the cost these future edges contribute to P by her present-bias $\beta \in (0, 1]$. We call the resulting estimate $\tilde{c}(P, \beta) = c(v, w) + \beta \cdot \sum_{e \in P \setminus \{(v,w)\}} c(e)$ the *perceived cost* of P. Taking the minimum perceived cost of all paths from v to t yields the perceived cost $\tilde{c}(v, \beta)$ of v. Similarly, we define the perceived cost $\tilde{c}(v, w, \beta)$ of (v, w) as the minimum perceived cost of all paths to t with (v, w) as their initial edge.

Definition 2.1.2 (Perceived Node and Edge Cost)**.** The perceived cost of a given node $v \in V \setminus \{t\}$ or edge $(v, w) \in E$ is defined as $\tilde{c}(v, \beta) = \min\{\tilde{c}(P, \beta) \mid P = v, \ldots, t\}$ and $\tilde{c}(v, w, \beta) = \min\{\tilde{c}(P, \beta) \mid P = v, w, \ldots, t\}$ respectively.

Once the agent has determined the perceived cost of her current node v, she compares it to the reward r. However, because she receives r upon reaching t rather than right away, she discounts its actual value by β. Drawing on our previous terminology, we call $\beta \cdot r$ the *perceived reward.* If the agent believes the reward to compensate upcoming expenses ,i.e., $\tilde{c}(v, \beta) \leq \beta \cdot r$, she is willing to continue her walk through G. In this case, she crosses which ever edge (v, w) matches her perceived reward, i.e., $\tilde{c}(v, w, \beta) = \tilde{c}(v, \beta)$, and the entire procedure is repeated at w. Ties between outgoing edges are broken arbitrarily. Otherwise, if $\tilde{c}(v, \beta) > \beta \cdot r$, she deems any further work towards reaching t unprofitable and abandons G prematurely without collecting any reward.

We consider G to be *motivating* if and only if the agent does not abandon G while constructing her path from s to t. Note that more than one such path might exist due to ties in the perceived cost of incident edges. In this case, the agent must not abandon on any of her paths for G to be motivating.

Definition 2.1.3 (Motivating Task Graph)**.** A task graph $G = (V, E, c, r)$ is motivating for an agent with present-bias $\beta \in (0, 1]$ if and only if she does not abandon G on any of the paths she may take from s to t.

Without loss of generality, we often implicitly assume that each node of G is located on a path from s to t. This assumption is justified because the agent cannot plan to construct paths along nodes that do not satisfy this property. Consequently, no such node is relevant to the agent's behavior and we can safely remove them from G, for instance in a pre-processing step.

2.2 Revisiting the Post Office

To become more familiar with the graphical model, we take another look at the story from Section 1.1. Recall that the protagonist of this story is an agent named Alice who has 10 days to pick up a package from the post office. Each day that the package is not in her possession, she pays a cost of c for not being able to use its content. Furthermore, she has to invest a one-time effort of $c' > c$ to collect the package in the first place. For the sake of a simple discussion, we set $c = 1$ and $c' = 2$ and assume that Alice's reward for collecting the package is very large. This way we do not need to consider cases in which she is inclined to leave the package at the post office.

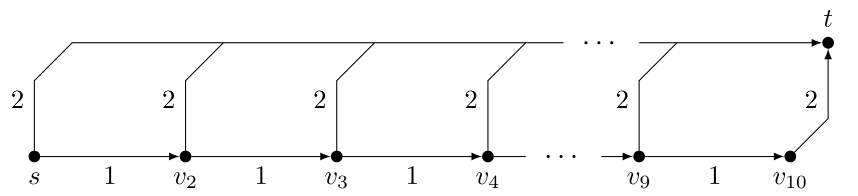

Figure 2.1: Task graph of the post office scenario

The task graph G depicted in Figure 2.1 captures this setting in the following way: Each node v_i, with $s = v_1$, represents one of the days $i \in \{1, \dots, 10\}$ on which Alice may collect the package. To complete this task, she must cross the edge (v_i, t) for a cost of $c(v_i, t) = c' = 2$. Note that Figure 2.1 merges some of the edges (v_i, t) to keep the drawing simple. If $i < 10$, Alice may also defer the errand by crossing the edge (v_i, v_{i+1}). The incurred cost $c(v_i, v_{i+1}) = c = 1$ corresponds to cost Alice pays because she cannot use the package on day i.

With the task graph G in place, we are ready to investigate Alice's behavior for any present-bias $\beta \in (0, 1]$ she may have. According to the rules of the graphical model, Alice decides her next move at a given node v_i based on the perceived cost of the paths from v_i to t. Due to the structure of G, each such path $P_{i,j} = v_i, \dots, v_{i+j}, t$ is uniquely defined by the length $j \in \{0, \dots, 10 - i\}$ of its associated delay. If Alice plans to cross the edge (v_i, t) to collect the package right away, she can only take $P_{i,0}$. The perceived cost of this edge therefore evaluates to

$$\tilde{c}(v_i, t, \beta) = \min\{\tilde{c}(P_{i,0}, \beta)\} = \tilde{c}(P_{i,0}, \beta) = c(v_i, t) = 2.$$

Conversely, as long as $i < 10$, Alice may cross the edge (v_i, v_{i+1}) to wait another day. In this case, she can choose any $P_{i,j}$ for which $j > 0$, resulting in a perceived cost of

$$\begin{aligned}\tilde{c}(v_i, v_{i+1}, \beta) &= \min\{\tilde{c}(P_{i,j}, \beta) \mid 1 \le j \le 10 - i\} \\ &= \min\Big\{c(v_i, v_{i+1}) + \beta \cdot \Big(c(v_{i+j}, t) + \sum_{k=1}^{j-1} c(v_{i+k}, v_{i+k+1})\Big) \;\Big|\; 1 \le j \le 10 - i\Big\} \\ &= \min\{1 + \beta \cdot (j+1) \mid 1 \le j \le 10 - i\} = 1 + \beta \cdot 2.\end{aligned}$$

Depending on β, three different types of behavior may emerge from Alice's preference between the two edges (v_i, t) and (v_i, v_{i+1}): If $\beta > 1/2$, then $\tilde{c}(v_i, t, \beta) < \tilde{c}(v_i, v_{i+1}, \beta)$ and Alice collects the package on the very first day. On the other hand, if $\beta < 1/2$, then $\tilde{c}(v_i, t, \beta) > \tilde{c}(v_i, v_{i+1}, \beta)$ and Alice repeatedly defers the errand until the very last day. Finally, if $\beta = 1/2$, then $\tilde{c}(v_i, t, \beta) = \tilde{c}(v_i, v_{i+1}, \beta)$, implying that Alice is indifferent between the two option. Consequently, she may pick up the package from the post office on any day i within the 10-day time period.

2.3 Computational Considerations

The previous section demonstrates how the graphical model reduces human decision making to a purely mechanical process. The beauty of the model is that we can implement this process to automatically simulate and analyze the behavior of a present-biased person. However, a successful implementation requires a closer look at the evaluation of perceived cost. After all, this cost is at the heart of an agent's decisions.

According to Definition 2.1.2, the perceived cost $\tilde{c}(v, w, \beta)$ of an edge (v, w) is defined as the minimum perceived cost with respect to all paths from v to t across the initial edge (v, w). From a conceptual point of view, this definition seems intuitive. However, it is rather impractical from a computational perspective since the task graph G could contain an exponential number of paths from v to t across the edge (v, w). A brute force search for the minimum perceived cost is therefore intractable. To speed things up, let $d(v) = \min\{\sum_{e \in P} c(e) \mid P = v, \dots, t\}$ be the cost of a cheapest path from v to t. Clearly, $d(v)$ can be computed in polynomial time with respect to the encoding length of G via standard shortest path algorithms. Furthermore, it holds true that

$$\begin{aligned}
\tilde{c}(v, w, \beta) &= \min\{\tilde{c}(P, \beta) \mid P = v, w, \dots, t\} \\
&= \min\Big\{c(v, w) + \beta \cdot \sum_{e \in P \setminus \{(v,w)\}} c(e) \;\Big|\; P = v, w, \dots, t\Big\} \\
&= c(v, w) + \beta \cdot \min\Big\{\sum_{e \in P'} c(e) \;\Big|\; P' = w, \dots, t\Big\} = c(v, w) + \beta \cdot d(w).
\end{aligned}$$

This yields an alternative and more tractable definition of perceived edge cost. Similarly, we can compute the perceived cost of a node $v \in V \setminus \{t\}$ by determining the minimum perceived cost over all outgoing edges (v, w), i.e.,

$$\begin{aligned}
\tilde{c}(v, \beta) &= \min\{\tilde{c}(P, \beta) \mid P = v, \dots, t\} = \min\Big\{\min\{\tilde{c}(P, \beta) \mid P = v, w, \dots, t\} \;\Big|\; (v, w) \in E\Big\} \\
&= \min\{c(v, w) + \beta \cdot d(w) \mid (v, w) \in E\}.
\end{aligned}$$

Having an efficient way to compute perceived cost is relevant for two reasons. First, it implies that the graphical model is compatible with agents who have limited resources to make a decision; a notion that is also known as *bounded rationality* in the behavioral economic literature, see e.g. [6]. Secondly, it allows us to simulate and quantify present-biased behavior in real-world settings. One parameter that is of particular interest to us in this context is the *minimal motivating reward* $\mu(G, \beta)$.

Definition 2.3.1 (Minimal Motivating Reward)**.** The minimal motivating reward $\mu(G, \beta)$ is the smallest reward for which the task graph G becomes motivating for an agent with present-bias $\beta \in (0, 1]$.

Keeping in mind that perceived edge and node cost can be computed in polynomial time with respect to the encoding length of G and β, it is not too hard to see that $\mu(G, \beta)$ can be as well.

Proposition 2.3.1. *The minimal motivating reward $\mu(G, \beta)$ can be computed in polynomial time with respect to the encoding length of G and β.*

Proof. The minimal motivating reward $\mu(G, \beta)$, can be computed via a simple depth first search through G. The search starts at s and only considers edges (v, w) satisfying $\tilde{c}(v, w, \beta) = \tilde{c}(v, \beta)$. Let W be the set of nodes encountered during the search. Clearly, W can be constructed in polynomial time with respect to the encoding length of G and β. Furthermore, it is easy to see that W contains exactly the nodes that an agent with present-bias β may visit if the reward at t is sufficiently motivating. Consequently, the minimal motivating reward is equal to $\mu(G, \beta) = \max\{\tilde{c}(v, \beta) \mid v \in W\}/\beta$, which implies the desired result. □

A direct consequence of this result is that it provides an efficient way to decide whether a certain task graph G is motivating for an agent with present-bias β.

Proposition 2.3.2. *Deciding whether a task graph $G = (V, E, c, r)$ is motivating for an agent with present-bias $\beta \in (0, 1]$ is possible in polynomial time with respect to the encoding length of G and β.*

Proof. By definition of the minimal motivating reward, it is easy to see that G is motivating if and only if $r \geq \mu(G, \beta)$. However, Proposition 2.3.1 implies that $\mu(G, \beta)$ can be computed within the given time bounds. This completes the proof. □

2.4 The Scalability of Task Graphs

We conclude this chapter with a proposition that will be of great help to us later on. To motivate the proposition, imagine a scenario without monetary transactions. In this case it may be difficult to assign cost and reward in absolute terms. Conveniently, the graphical model is invariant to the numerical parameters of the task graph as long as their ratio is preserved. To see this, we define the concept of a *scaled task graph*:

Definition 2.4.1 (Scaled Task Graphs)**.** Let $\lambda > 0$ be an arbitrary positive scaling factor. The task graph $G_\lambda = (V, E, c_\lambda, r_\lambda)$ is a scaled version of $G = (V, E, c, r)$ if and only if $c_\lambda(e) = \lambda \cdot c(e)$ for all $e \in E$ and $r_\lambda = \lambda \cdot r$.

It is not too hard to see that the agent's behavior in G_λ is identical for all scaling parameters $\lambda > 0$. As a result, all versions of G_λ share the same motivational properties. This observation is particularly convenient as it generalizes many complexity theoretic results that we present in the next chapters from specific reward values to arbitrary reward values.

Proposition 2.4.1. *The motivational properties of a task graph $G_\lambda = (V, E, c_\lambda, r_\lambda)$ are independent of the scaling factor $\lambda > 0$.*

Proof. Let $\lambda > 0$ be an arbitrary scaling factor and assume that the agent is located at some node $v \neq t$. It is easy to see that the perceived cost of any path $P = v, w, \ldots, t$ in G_λ is exactly λ times the perceived cost of P in G. More formally, it holds true that

$$\tilde{c}_\lambda(P, \beta) = c_\lambda(v, w) + \beta \cdot \sum_{e \in P \setminus \{(v,w)\}} c_\lambda(e) = \lambda \cdot c(v, w) + \beta \cdot \sum_{e \in P \setminus \{(v,w)\}} \lambda \cdot c(e) = \lambda \cdot \tilde{c}(P, \beta).$$

This means that the agent's preference over the outgoing edges (v, w) in G_λ is identical to her preference in G. Furthermore, the reward r_λ is by definition equal to λ times the reward r. Combining the two observations shows that the agent traverses the same edges and abandons at the same nodes in both graphs. As a result, G_λ must be motivating if and only if G is. This implies the proposition. □

3 Prohibition Based Incentives

In this chapter we focus on incentives that prohibit an agent from performing certain tasks. The goal is to impose these prohibitions in a way that reduces the reward required to keep the agent motivated. We are particularly interested in the computational complexity of computing optimal or approximately optimal solutions for this incentive design problem. However, to introduce the setting we neglect computational considerations for a moment and first take a closer look at the conceptual power of prohibition based incentives.

3.1 The Car Wash

The benefit of strategic prohibition is perhaps best illustrated by a simple example. Consider, for instance, a scenario in which Alice has 10 days time to wash her car. On each day she can either do the chore right away or wait until the next day. If she decides to procrastinate, she incurs no immediate effort. However, the longer she waits the harder it becomes to clean the car. Assuming an exponential increase of effort, let 2^i be the cost of cleaning the car on a particular day $i \in \{1, \ldots, 10\}$. The task graph $G = (V, E, c, r)$ depicted in Figure 3.1 captures this scenario by assigning each day i to a distinct node v_i. In the special case of $i = 1$, let $s = v_1$. Edges of the form (v_i, t) correspond to the task of washing the car on day i, while edges of the form (v_i, v_{i+1}) with $i < 10$ are procrastination edges that postpone the task to the next day.

Assume that Alice has a present-bias of $\beta = 1/2$. When located at a node v_i with $i < 10$, it is easy to see that she is indifferent between washing her car right away or waiting until the next day, i.e.,

$$\tilde{c}(v_i, t, \beta) = 2^i = \beta \cdot 2^{i+1} = \min\{0 + \beta \cdot 2^j \mid i + 1 \leq j \leq 10\} = \tilde{c}(v_i, v_{i+1}, \beta)$$

As a result, it may happen that Alice procrastinates until day 10 at which point she must wash the car to receive the reward. Since the edge (v_{10}, t) has a cost of 2^{10}, this implies that a reward greater or equal to $2^{10}/\beta = 2048$ is required to make G motivating, i.e., $\mu(G, 1/2) \geq 2048$. Clearly, such a high reward is not very cost effective.

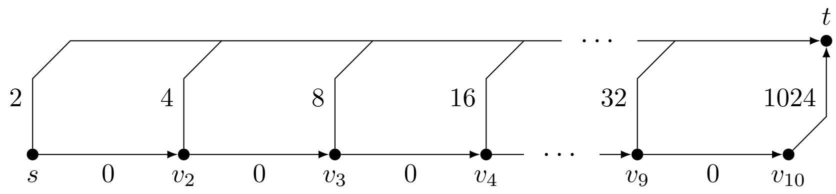

Figure 3.1: Task graph of the car wash scenario

A sensible measure to improve efficiency is to introduce a deadline that prohibits Alice from procrastinating. We can implement such a deadline by removing the edge (s, v_2) from the task graph G. The set $F = \{(s, v_2)\}$ of removed edges induces a new task graph $G_F = (V, E \setminus F, c, r)$. We call this task graph a *subgraph* of G.

Definition 3.1.1 (Subgraph)**.** The subgraph $G_F = (V, E \setminus F, c, r)$ is a task graph obtained by removing the set $F \subseteq E$ of edges from a given task graph $G = (V, E, c, r)$.

Now, assume that Alice is located at the source node s of the subgraph G_F. Since G_F does not contain the edge (s, v_2) anymore, she does not receive the reward if she postpones the car wash to the next day. In fact, her only option to receive the reward is to traverse the edge (s, t) immediately. The perceived cost of this plan evaluates to $\tilde{c}_F(s, t, \beta) = 2$. Note that we write $\tilde{c}_F$ instead of $\tilde{c}$ whenever we refer to some perceived cost in the subgraph induced by F. Similarly, we write d_F to denote the cost of a cheapest paths in G_F. The above considerations imply that a reward of $2/\beta = 4$ is sufficient to make G_F motivating. In fact, it is not too hard to see that this reward is the best possible among all subgraphs of G. We therefore refer to this reward as the *minimum motivating reward* and denote it by $\mu^*(G, 1/2)$.

Definition 3.1.2 (Minimum Motivating Reward)**.** The minimum motivating reward $\mu^*(G, \beta)$ is the smallest reward for which the task graph G admits a subgraph that is motivating for an agent with present-bias $\beta \in (0, 1]$.

Comparing the minimal motivating reward with the minimum motivating reward shows that the removal of (s, v_2) improves the cost efficiency of G by a factor of

$$\frac{\mu(G, 1/2)}{\mu^*(G, 1/2)} = \frac{2048}{4} = 512.$$

Considering a slightly generalized version of the car wash scenario, it becomes apparent that this *cost efficiency ratio* can grow exponential in the size of the task graph.

Figure 3.2: Alice cleaning her car

Proposition 3.1.1. *The cost efficiency ratio* $\mu(G,\beta)/\mu^*(G,\beta)$ *lies between* 1 *and* β^{-n+2} *for all task graphs* G *and present-bias values* $\beta \in (0,1]$. *This result is tight.*

Proof. Clearly, the ratio $\mu(G,\beta)/\mu^*(G,\beta)$ is never less than 1 since $G_\emptyset$ is a valid subgraph of G. Moreover, it is easy to construct a task graph G in which it is of no advantage to remove edges, e.g. a task graph with just a single edge. This proves that the lower bound of 1 is valid and tight. We therefore focus on the upper bound β^{-n+2}.

As a first step, we check the validity of the upper bound. For this purpose, consider an arbitrary task graph $G = (V, E, c, r)$. By definition of the minimal motivating reward $\mu(G,\beta)$, at least one of the agent's paths from s to t must contain a node $w \neq t$ with a perceived cost of $\tilde{c}(w,\beta) = \beta \cdot \mu(G,\beta)$. Assuming that $w \neq s$, let v be a node that the agent visits before w. Considering that

$$\tilde{c}(v,\beta) = \tilde{c}(v,w,\beta) = c(v,w) + \beta \cdot d(w) \geq \beta \cdot d(w) \geq \beta \cdot \tilde{c}(w,\beta),$$

it should be clear that the perceived cost increases by no more than a factor of β^{-1} as the agent moves from v to w. However, any path that the agent may take from s to w consists of at most $n-1$ nodes. An inductive argument on these nodes immediately implies that the perceived cost of s is at least $\tilde{c}(s,\beta) \geq \beta^{n-2} \cdot \tilde{c}(w,\beta) = \beta^{n-1} \cdot \mu(G,\beta)$.

Conversely, let G_F be a subgraph that is motivating for the minimum motivating reward $\mu^*(G,\beta)$. The perceived cost of s in G_F is a lower bound on the perceived value of $\mu^*(G,\beta)$, i.e., $\beta \cdot \mu^*(G,\beta) \geq \tilde{c}_F(s,\beta)$. Considering that the perceived cost of s can only decrease whenever additional edges are added to G_F, we conclude that

$$\beta^{-n+2} \cdot \mu^*(G,\beta) = \beta^{-n+1} \cdot (\beta \cdot r') \geq \beta^{-n+1} \cdot \tilde{c}_F(s,\beta) \geq \beta^{-n+1} \cdot \tilde{c}(s,\beta) \geq \mu(G,\beta).$$

This establishes the upper bound of β^{-n+2}.

It remains to show that the upper bound is tight. To prove tightness, consider a task graph $G = (V, E, c, r)$ that consist of a directed path $P = v_1, \ldots, v_{n-1}$ and an additional edge (v_i, t) for each node v_i of P. The edges of P are free of charge whereas the edges (v_i, t) have exponentially increasing cost of $c(v_i, t) = \beta^{-i}$. The initial node of P also serves as the source node, i.e., $s = v_1$.

For each $i < n - 1$, it should be easy to see that an agent with present-bias β is indifferent between the two edges (v_i, v_{i+1}) and (v_i, t) when located at v_i. The reason is the same as in the car wash scenario. It may therefore happen that the agent walks all the way from s to v_{n-1} along the path P. However, once she reaches v_{n-1}, the only remaining path to t leads across the edge (v_{n-1}, t). Since this edge has a cost of β^{-n+1}, a reward of $\beta^{-n+1}/\beta = \beta^{-n}$ or more is necessary for G to be motivating, i.e., $\mu(G, \beta) \geq \beta^{-n}$. In contrast, consider a subgraph G_F in which the edge (s, v_2) is removed. Clearly, the only path from s to t in G_F is along the edge (s, t) for a cost of β^{-1}. But this implies that G_F is motivating for a reward of $\beta^{-1}/\beta = \beta^{-2}$, i.e., $\mu^*(G, \beta) \leq \beta^{-2}$. We conclude that the ratio between the minimal motivating reward and minimum motivating reward is at least β^{-n+2} and therefore the upper bound is tight. □

3.2 Computing Motivating Subgraphs

As demonstrated in the previous section, prohibition can improve the cost efficiency of task graphs considerably. However, to exploit the full potential of this incentive design strategy, we need to identify an optimal set F of edges to remove from a given task graph $G = (V, E, c, r)$, i.e., a set F that satisfies $\mu(G_F, \beta) = \mu^*(G, \beta)$. A straight forward approach to this problem is a brute force search through all subsets $F \subseteq E$. Of course, such a search becomes intractable very fast considering that E admits exponentially many subsets with respect to its size. As a result, one may wonder whether more efficient algorithms exist.

To investigate this question, we take a closer look at the computational complexity of finding an optimal set F. As a first step, we define a decision version of the problem called MOTIVATING SUBGRAPH (MSG):

Definition 3.2.1 (MSG)**.** The problem of deciding whether a task graph $G = (V, E, c, r)$ admits a subgraph G_F that is motivating for a given present-bias $\beta \in (0, 1]$.

It is easy to see that MSG is contained in the complexity class NP. The reason is that any solution of a feasible MSG instance can be verified in polynomial time.

Proposition 3.2.1. *MSG is contained in NP.*

Proof. Let $G = (V, E, c, r)$ be the task graph of an MSG instance $\mathcal{I}$ that admits a motivating subgraph $G_F = (V, E \setminus F, c, r)$. To establish membership of MSG in NP, we need a certificate of polynomial size that verifies the existence of a feasible solution in polynomial time. The obvious candidate for such a certificate is of course G_F. On the one hand, the encoding length of G_F is clearly polynomial in that of $\mathcal{I}$ and it is easy to check whether G_F is indeed a subgraph of G. On the other hand, Proposition 2.3.2 states that it can be decided in polynomial time with respect to the encoding length of G_F and β whether G_F is motivating. Because β is part of the given MSG instance $\mathcal{I}$, this completes the proof. □

A more interesting question is whether MSG is contained in the complexity class P, i.e., does there exist a polynomial time algorithm to decide whether a given MSG instance has a feasible solution? In the special case of $\beta = 1$, such an algorithm exists indeed. The reason is that an agent with no present-bias simply follows a cheapest path from s to t. Therefore, it is of no advantage to delete edges and a given MSG instance admits a motivating subgraph if and only if its task graph is motivating itself. According to Theorem 2.3.2, the latter condition can be checked in polynomial time. A similar approach also decides MSG whenever no reward is placed, i.e., $r = 1$.

In the remainder of this section, we argue all other instance of MSG are far more difficult to decide. More precisely, we show that MSG is NP-hard for $\beta \in (0, 1)$ and $r > 0$. In particular this means that MSG is not contained in P unless P = NP. To prove NP-hardness, we construct a polynomial reduction from the well-known, NP-hard 3-SATISFIABILITY (3-SAT) problem, see e.g. [12].

Definition 3.2.2 (3-SAT)**.** The problem of deciding whether ℓ clauses $c_1, \dots, c_\ell$ over m variables $x_1, \dots, x_m$ with 3 literals per clause have a satisfying truth-value assignment.

In the analysis of the reduction, we encounter several paths $P = v_{m+1}, \dots, v_0$ with exponentially increasing edge cost $c(v_i, v_{i-1}) = (1-\beta)^{i-1}$. According to the Lemma 3.2.2, the perceived cost of P evaluates to 1. It is particularly interesting to observe that the same holds true for any non-empty subpath of P ending at v_0. In other words, the perceived cost of P stays invariant as the agent travels along its edges. As it turns out, this is a convenient property, which we use several times through this work.

Lemma 3.2.2. *For any path $P = v_{m+1}, \dots, v_0$ with edge cost $c(v_i, v_{i-1}) = (1 - \beta)^{i-1}$, it holds true that $\tilde{c}(P, \beta) = 1$.*

Proof. According to the geometric series, it is possible to express the sum $\sum_{i=0}^{m-1}(1-\beta)^i$ as $(1 - (1 - \beta)^m)/\beta$ in closed form. Together with the definition of perceived cost, this immediately yields the desired result

$$\tilde{c}(P,\beta) = (1-\beta)^m + \beta \cdot \sum_{i=0}^{m-1} (1-\beta)^i = (1-\beta)^m + \beta \cdot \frac{1-(1-\beta)^m}{\beta} = 1. \qquad \square$$

We are now ready to establish that MSG is NP-hard. The proof of this theorem is an adapted version of a reduction originally presented in our work on penalty based incentives [3]. An alternative proof that was discovered independently of ours can be found in the work of Tang et al. [23].

Theorem 3.2.3. *MSG is NP-hard for any present-bias $\beta \in (0,1)$ and reward $r > 0$.*

Proof. Let $\mathcal{I}$ be an arbitrary instance of 3-SAT, which consists of ℓ clauses $c_1, \dots, c_\ell$ over m variables $x_1, \dots, x_m$. Furthermore, let $\beta \in (0,1)$ be a fixed value independent of $\mathcal{I}$. Our goal is to construct a task graph $G = (V, E, c, r = 1/\beta)$ in polynomial time such that the resulting MSG instance $\mathcal{J}$ has the following two properties:

(a) If $\mathcal{I}$ is satisfiable, then $\mathcal{J}$ must admit a motivating subgraph.

(b) If $\mathcal{J}$ admits a motivating subgraph, then $\mathcal{I}$ must be satisfiable.

This reduction proves the theorem for $r = 1/\beta$. However, the more general result for $r > 0$ follows directly from the scalability of r according to Proposition 2.4.1.

As the first step of the reduction, we specify the structure of G. Figure 3.3 depicts G for a simple sample instance of 3-SAT. Note that the drawing splits G at the node u_1 into a *top* and a *bottom layer*. The top layer contains a node $v_{k,y}$ for each variable x_k and truth-value $y \in \{T, F\}$. We call these nodes *boolean nodes* and the edges between them *regular edges*. The cost of these regular edges is $(1-\beta)^3 - \varepsilon$, where ε denotes a small but positive quantity satisfying

$$\varepsilon < \min\Big\{(1-\beta)^3, \frac{\beta \cdot (1-\beta)^3}{1+\beta}\Big\}.$$

As Figure 3.3 shows, there is a regular edge $(v_{k,y}, v_{k+1,y'})$ for each pair of boolean nodes associated with two consecutive variables x_k and x_{k+1}. Furthermore, all boolean nodes of the first variable x_1 or the last variable x_m are connected to the other nodes of G via a regular edge of the form $(u_2, v_{1,y})$ or $(v_{m,y'}, u_3)$ respectively. The idea behind this construction is that the agent's path along the boolean nodes corresponds to a satisfying truth-value assignment $\tau : \{x_1, \dots, x_m\} \to \{T, F\}$ and vice versa.

Similar to the boolean nodes at the top layer, there is a so-called *literal node* $w_{i,j}$ for each literal $j \in \{1,2,3\}$ of each clause c_i at the bottom layer. The literal nodes are

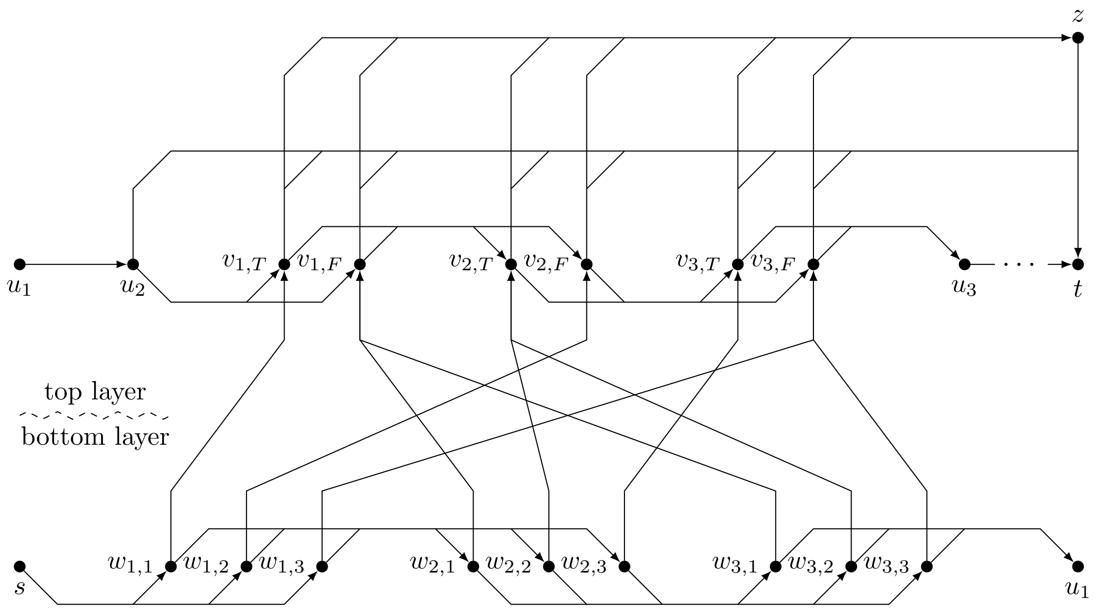

Figure 3.3: Reduction from the 3-SAT instance: $(\neg x_1, x_2, x_3), (x_1, \neg x_2, \neg x_3), (x_1, \neg x_2, x_3)$

connected in the same way as the boolean nodes, i.e., there is a regular edge $(w_{i,j}, w_{i+1,j'})$ of cost $(1-\beta)^3 - \varepsilon$ for each pair of literal nodes associated with two consecutive clauses c_i and c_{i+1} as well as a regular edge $(s, w_{1,j})$ and $(w_{\ell,j'}, u_1)$ for each literal node of the first clause c_1 and last clause c_ℓ. The purpose of the bottom layer is to identify a true literal per clause with respect to the truth-value assignment τ obtained at the top layer. This verifies that τ is indeed satisfying.

In addition to the regular edges, G contains several *shortcuts*. First, there is a shortcut of cost $(1-\beta)^2$ from each literal node $w_{i,j}$ to a distinct boolean node. If the j-th literal of c_i is equal to a variable x_k, this shortcut goes to $v_{k,F}$. Otherwise, if the literal is negated and equal to $\neg x_k$, it goes to $v_{k,T}$. Secondly, there are two shortcuts from each boolean nodes $v_{k,y}$ to the target node t. One of them consists of a single edge of cost $\sum_{i'=0}^{2}(1-\beta)^{i'}$, while the other is routed through the intermediate node z and consists of two edges; the first is free of charge while the second has a cost of $2-\beta$. Finally, there is one more shortcut from u_2 to t via an edge of cost $2-\beta$.

To complete the task graph G, we add four more auxiliary edges (u_1, u_2), (u_3, u_4), (u_4, u_5) and (u_5, t). The cost of these edges is $(1-\beta)^2$, $(1-\beta)^2$, $1-\beta$ and 1 respectively. Clearly, the resulting graph is acyclic. Moreover, since all numerical values are independent of the particular 3-SAT instance $\mathcal{I}$, the graph can be constructed in polynomial time and is of polynomial size with respect to $\mathcal{I}$.

Having specified the task graph G, we proceed to argue that the corresponding MSG instance $\mathcal{J}$ satisfies property (a), i.e., $\mathcal{J}$ admits a motivating subgraph whenever $\mathcal{I}$ is satisfiable. For this purpose, let τ be a satisfying truth-value assignment of $\mathcal{I}$ and consider the subgraph G_F obtained by removing all shortcut edges of the form $(v_{k,\tau(x_k)}, z)$ and $(v_{k,\neg\tau(x_k)}, t)$ from G. This way, all boolean nodes satisfied with respect to τ have a single edge shortcut and all unsatisfied boolean nodes have a double edge shortcut. Furthermore, assume that all regular edges incident to an unsatisfied boolean or literal node are deleted as well. Figure 3.4 shows the resulting subgraph G_F for a satisfying truth-value assignment of the 3-SAT instance given in Figure 3.3.

To prove that the subgraph G_F is indeed motivating, we have to determine the agent's walk through G_F. Starting at s, her only option is to cross a regular edge and move to one of the literal nodes that set the first clause to true. Because τ is a satisfying truth-value assignment, at least one such node $w_{1,j}$ must exist. If the agent plans to take the shortcut at $w_{1,j}$ immediately, Lemma 3.2.2 yields an upper bound of

$$\begin{aligned}\tilde{c}_F(s, w_{1,j}, \beta) &\leq (1-\beta)^3 - \varepsilon + \beta \cdot \big((1-\beta)^2 + 0 + (2-\beta)\big) \\ &= (1-\beta)^3 + \beta \cdot \Big(\sum_{i'=0}^{2} (1-\beta)^{i'}\Big) - \varepsilon = 1 - \varepsilon\end{aligned}$$

for the perceived cost of $(s, w_{1,j})$. This is less than the perceived reward and therefore she is motivated to move forward to $w_{1,j}$ where she faces two new options: Either she takes the immediate shortcut the way she planned at s or she crosses a regular edge to one of the satisfied literal nodes of the next clause. With the help of Lemma 3.2.2 it is easy to see that the perceived cost of the immediate shortcut is 1. The perceived cost of the regular edges on the other hand is at most $1 - \varepsilon$ according to the same reasoning that bounds the perceived cost at s. Consequently, the agent prefers the regular edges over the shortcut and moves to the next literal node. Repeating this argument proves that she travels step by step along the literal nodes of the bottom layer until she reaches a node $w_{\ell,j'}$ of the last clause. At this point, she cannot plan to take a shortcut at a later literal node anymore. However, if she plans to take the shortcut at u_2 instead, the perceived cost of $(w_{\ell,j'}, u_1)$ is still at most $1 - \varepsilon$. As a result, she eventually moves to u_2 via the intermediate node u_1. This completes the bottom layer.

The analysis of the top layer reveals an identical behavior pattern. From the initial node u_2 to the last boolean node $v_{m,\tau(x_m)}$, the agent consistently prefers the next shortcut over the present shortcut. Consequently, she moves from one satisfied boolean node to the next until she reaches $v_{m,\tau(x_m)}$. At this point she prefers to move to t via the nodes

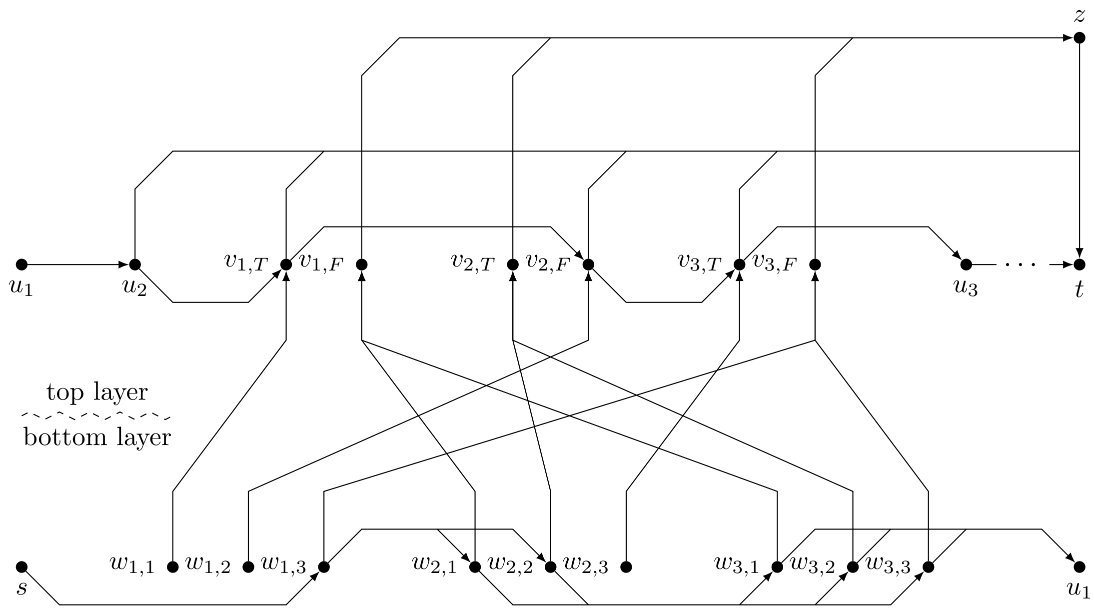

Figure 3.4: Subgraph for the truth-value assignemnt $\tau(x_1) = \tau(3) = T$ and $\tau(x_2) = F$

u_3, u_4 and u_5. According to Lemma 3.2.2, the perceived cost of these nodes never exceeds 1, which proves that G_F is motivating.

All that remains is to show that our reduction also satisfies property (b), i.e., $\mathcal{I}$ has a satisfying truth-value assignment whenever $\mathcal{J}$ admits a motivating subgraph. For this purpose, let G_F be a motivating subgraph and P the path that the agent takes through G_F. It is important to observe that P cannot contain any shortcuts. For the top layer, this follows from the fact that all of its shortcuts contain an edge e of cost $2-\beta$ or more. Clearly, the perceived cost of e is at least $\tilde{c}_F(e, \beta) \geq 2 - \beta > 1$ and therefore not motivating. To verify the observation for the bottom layer, assume that P contains a shortcut edge from some literal node $w_{i,j}$ to a boolean node $v_{k,y}$. The perceived cost of this edge can be at most 1, otherwise the agent loses motivation. By construction of G, there is a distinct cheapest path from $v_{k,y}$ to t, namely the double edge shortcut at $v_{k,y}$. Because the perceived cost of the path from $w_{i,j}$ to t via this shortcut evaluates to 1, we know that the shortcut must be contained in G_F. However, once the agent has reached $v_{k,y}$, the perceived cost of the shortcut decreases to $\beta \cdot (2 - \beta)$. In contrast, any path $P' = v_{k,y}, \ldots, t$ whose initial edge is regular must include a top layer shortcut or pass the nodes u_3, u_4 and u_5. In both cases, the perceived cost of P' is at least

$$\tilde{c}_F(P', \beta) \geq (1 - \beta)^3 - \varepsilon + \beta \cdot (2 - \beta) > \beta \cdot (2 - \beta).$$

The second inequality is valid by choice of ε. As a result, the agent prefers the double edge shortcut at $v_{k,y}$ over all regular edges. This is a contradiction to the previous observation that P does not contain a top layer shortcut.

Because P does not contain shortcuts, we conclude that it contains exactly one literal node $w_{i,j}$ and one boolean node $v_{k,y}$ for each clause c_i and variable x_k. Consequently, we can construct a truth-value assignment τ in the following way: If P includes the node $v_{k,T}$, then $\tau(x_k) = T$. Otherwise, if P includes $v_{k,F}$, then $\tau(x_k) = F$.

To conclude the proof, we need to show that τ indeed satisfies all clauses. For this purpose, consider an arbitrary clause c_i and let $w_{i,j}$ be the corresponding literal node contained in P. Furthermore, let w' be the node that precedes $w_{i,j}$ with respect to P. We denote the path that the agent plans to take at w' by P'. Clearly, the initial edge of P' must be the regular edge $(w', w_{i,j})$. However, this means that none of the subsequent edges can be regular anymore. The reason is that no matter how P' is chosen, the path always charges a cost of $(1-\beta)^2$ and $2-\beta$ at some point; the former cost must either be payed at a bottom layer shortcut or the edge (u_1, u_2) while the latter cost must be paid at a top layer shortcut or along the nodes u_3, u_4 and u_5. Consequently, any further regular edge added to P' would result in too much perceived cost. More precisely, the perceived cost of P' would amount to a value of at least

$$\begin{aligned}\tilde{c}_F(P', \beta) &\geq (1-\beta)^3 - \varepsilon + \beta \cdot \big((1-\beta)^3 - \varepsilon + (1-\beta)^2 + (2-\beta)\big)\\ &= 1 + (1+\beta) \cdot \Big(\frac{\beta \cdot (1-\beta)^3}{1+\beta} - \varepsilon\Big) > 1\end{aligned}$$

The inequality is valid by choice of ε. It is easy to see that this is not motivating. But if P' only contains one regular edge, it must consist of the direct shortcut from $w_{i,j}$ to the corresponding boolean node $v_{k,y}$ and the double edge shortcut from $v_{k,y}$ to t. Note that P' cannot contain the single edge shortcut at $v_{k,y}$ for the same reason that it cannot contain a second regular edge. As a result, we know that the double edge shortcut at $v_{k,y}$ must be part of G_F.

For the sake of contradiction, assume that P includes the boolean node $v_{k,y}$. As argued earlier, the perceived cost of a regular edge at $v_{k,y}$ is at least $(1-\beta)^3 - \varepsilon + \beta \cdot (2-\beta)$. By choice of ε, this is too expensive to keep the agent from entering the double edge shortcut at $v_{k,y}$ which only has a perceived cost of $\beta \cdot (2-\beta)$. Of course, this violates the observation that the agent must not take shortcuts. Consequently, P needs to contain the boolean node $v_{k,\neg y}$ instead. By construction of G this implies that the truth-value assignment τ satisfies the j-th literal of c_i. Because this holds true for all clauses, τ is a satisfying variable assignment. This completes the proof. □

3.3 Optimal Travel Routes

The fact that MSG is NP-hard immediately implies the same for the design of optimal subgraphs. But what computational challenges give rise to this result? To gain some more insight, consider a scenario in which partial information about the structure of a solution is revealed. More precisely, assume we know which nodes the agent visits in an optimal subgraph G_F for a reward of $\mu(G_F, \beta)$. We call the corresponding set W^* of nodes an *optimal travel route*.

It is interesting to observe that the knowledge of W^* renders the reduction of Theorem 3.2.3 ineffective. The reason is that the variable nodes contained in W^* correspond to a satisfying truth-value assignment whenever the 3-SAT instance of the reduction is satisfiable. Clearly, this implies that W^* is NP-hard to compute. But it also raises the question whether knowledge of W^* simplifies the design of an optimal subgraph. In the next chapter we prove a similar conjecture for penalty based incentives. Unfortunately, this result cannot be carried over to prohibition, i.e., an optimal subgraph remains NP-hard to design even if W^* is known. To prove this, consider the following promise version of MSG, which we call MSG-PROMISE:

Definition 3.3.1 (MSG-PROMISE)**.** The problem of deciding whether a task graph $G = (V, E, c, r)$ admits a motivating subgraph in which the agent visits the nodes of a predefined node set W or does not admit a motivating subgraph at all.

Given an optimal subgraph G_F of G, MSG-PROMISE can be decided in polynomial time whenever $W = W^*$; just check whether $\mu(G_F, \beta) \leq r$. However, by using a reduction from ℓ-DISJOINT CONNECTING PATHS (ℓ-DCP) we can show that MSG-PROMISE is NP-hard, even if $W = W^*$. This implies that knowledge of W^* does not reduce the complexity of computing optimal subgraphs.

Definition 3.3.2 (ℓ-DCP)**.** The problem of deciding whether a graph H with ℓ disjoint node pairs $(s_1, t_1), \ldots, (s_k, t_k)$ admits ℓ mutually node-disjoint paths, one connecting each s_i to the corresponding t_i.

According to Lynch [20], ℓ-DCP is NP-hard in undirected graphs. Conveniently, his reduction adapts seamlessly to directed acyclic graphs, see e.g. [2]. This allows us to emend ℓ-DCP instances into task graphs. In [2] we use such an embedding to prove that MSG is NP-hard. However, the same construction can prove an even stronger result, namely that MSG-PROMISE is NP-hard for $W = W^*$.

Theorem 3.3.1. *MSG-PROMISE is NP-hard for any present-bias* $\beta \in (0, 1)$*, reward* $r > 0$ *and* $W = W^*$*, where* W^* *is an optimal travel route.*

Proof. Let $\mathcal{I}$ be an arbitrary instance of ℓ-DCP that consists of a directed acyclic graph H and ℓ disjoint node pairs $(s_1, t_1), \ldots, (s_\ell, t_\ell)$. Furthermore, assume that the present-bias $\beta \in (0, 1]$ is independent of $\mathcal{I}$. Our goal is to reduce $\mathcal{I}$ to an RMSG instance $\mathcal{J}$ on a task graph $G = (V, E, c, r = 1/\beta)$ such that $\mathcal{J}$ satisfies the following three properties:

(a) If $\mathcal{J}$ admits motivating subgraphs, then the agent travels along a common path in each such subgraph. We call this path the *regular path.*

(b) If $\mathcal{I}$ has ℓ node-disjoint connecting paths, then $\mathcal{J}$ admits a motivating subgraph.

(c) If $\mathcal{I}$ does not have ℓ node-disjoint connecting paths, then $\mathcal{J}$ does not admit a motivating subgraph.

Choosing W to be the set of nodes on the regular path, the above properties prove the theorem for the special case of $r = 1/\beta$. For all other rewards $r > 0$, the theorem follows directly from the scalability of r shown in Proposition 2.4.1.

The task graph G, which is illustrated in Figure 3.5, has the following structure: First, there is the regular path from s to t along the intermediate nodes $v_1, \ldots, v_{\ell+3}$. The cost of the first $\ell + 1$ edges of this path is $(1 - \beta)^3 - \varepsilon$, where ε denotes a positive constant that satisfies

$$\varepsilon < \min\left\{\frac{\beta \cdot (1 - \beta)}{\ell + 1}, \frac{\beta \cdot (1 - \beta)^3}{1 + \beta}\right\}.$$

The last three edges $(v_{\ell+1}, v_{\ell+2})$,$(v_{\ell+2}, v_{\ell+3})$ and $(v_{\ell+3}, t)$ have a cost of $(1 - \beta)^2$, $1 - \beta$ and 1, respectively. Secondly, G contains ℓ *shortcuts* that connect each node v_i with $i \leq \ell$ to t via the embedding of H. More precisely, the i-th shortcut goes from its initial node v_i to a distinct node w_i via an edge of cost $(1-\beta)^2$. From w_i the shortcut continues to the terminal node s_i of H via an edge of cost $(\ell+1-i) \cdot (1-\beta)/(\ell+1)$. After passing through H, the shortcut connects t_i with t via an edge of cost $i \cdot (1 - \beta)/(\ell + 1) + 1$. Note that the prices of (w_i, s_i) and (t_i, t) are complements of each other and sum up to $(1 - \beta) + 1$. The edges of H are free of charge.

The resulting task graph G is acyclic and its construction requires at most polynomial time and space with respect to $\mathcal{I}$. Furthermore, the regular path of G is the only path from s to t whose edges are all of cost less than or equal to 1. This is due to the cost of the final shortcut edges (t_i, t). As a result, the agent must travel along the regular path to stay motivated. This establishes (a).

We continue to show that $\mathcal{J}$ satisfies property (b), i.e., $\mathcal{J}$ admits a motivating subgraph whenever $\mathcal{I}$ has ℓ node-disjoint connecting paths. For this purpose, assume that ℓ node-disjoint connecting paths exist in H and consider the subgraph G_F obtained by

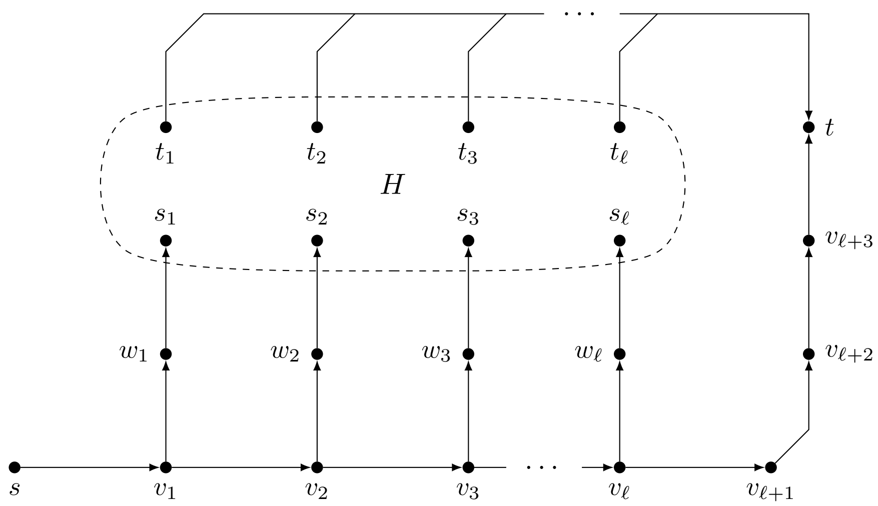

Figure 3.5: Reduction from an ℓ-DCP instance of a graph H

deleting all edges of H that are not part of one of these paths. When located at a node v_i with $i \leq \ell$, the agent has two choices: First, she may take the current shortcut at v_i. According to Lemma 3.2.2, the perceived cost of this option is

$$\begin{aligned}\tilde{c}_F(v_i, w_i, \beta) &= (1-\beta)^2 + \beta \cdot \Big(\frac{(\ell+1-i)\cdot(1-\beta)}{\ell+1} + \frac{i\cdot(1-\beta)}{\ell+1} + 1\Big) \\ &= (1-\beta)^2 + \beta \cdot \big((1-\beta)+1\big) = 1.\end{aligned}$$

Secondly, she may take a regular edge and move to the next node v_{i+1}. Assuming that she plans to continue her walk along the shortcut at v_{i+1}, or the nodes $v_{\ell+1}$, $v_{\ell+2}$ and $v_{\ell+3}$ in the special case of $i = \ell$, the perceived cost of this option is at most $\tilde{c}_F(v_i, v_{i+1}, \beta) \leq 1 - \varepsilon$. Again, this bound is a direct result of Lemma 3.2.2. Similar calculations reveal a perceived cost of at most 1 for each of the edges (s, v_1), $(v_{\ell+1}, v_{\ell+2})$ and $(v_{\ell+2}, v_{\ell+3})$. As a result, all edges going out of nodes of the regular path are motivating. Nevertheless, the agent strictly prefers the regular edges over the shortcuts. This means that she travels along the regular path until she eventually reaches t.

It remains to show that $\mathcal{J}$ satisfies (c), i.e., $\mathcal{J}$ admits no motivating subgraph whenever $\mathcal{I}$ does not have ℓ node-disjoint connecting paths. To prove this statement, let G_F be an arbitrary subgraph. Without loss of generality, we may assume that the agent travels along the regular path of G_F. The reason is that no other type of subgraph can possibly be motivating according to (a). Assuming that H does not have ℓ node-disjoint connecting paths, our goal is to show that G_F is not motivating.

We call the i-th shortcut in G_F *degenerated* if the cost of a cheapest path from v_i to t via w_i is different from the *target value* $\theta = \sum_{i'=0}^{2}(1-\beta)^{i'}$. In particular, the i-th shortcut is degenerated if there is no path from v_i to t via w_i in which case the cost is infinite. Note that all degenerated shortcuts miss the target value by $(1-\beta)/(\ell+1)$ or more. Because H does not admit ℓ node-disjoint connecting paths, at least one degenerated shortcut must exist in G_F. To see this, assume for a moment that no such shortcut exists. This means that there is a cheapest path P_i from v_i to t via w_i for all $i \leq \ell$. By construction of G, it holds true that P_i contains (w_i, s_i). As the total cost of P_i sums up to θ, it follows that the last edge of P_i must be (t_i, t). Furthermore, P_i must be node-disjoint from all other paths P_j with $j < i$. Otherwise, P_i would not be a shortest path from v_i to t considering that the cost of (t_j, t) is less than the cost of (t_i, t). As a result, the subpaths between s_i and t_i correspond to ℓ node-disjoint paths in H. This contradicts the assumption that no ℓ node-disjoint paths exist.

Having established the existence of a degenerated shortcut in G_F, we distinguish two cases: Either there is a degenerated shortcut at v_i such that the cost of a cheapest path P_i from v_i to t via w_i is less than θ, or all degenerated shortcuts cost more than θ. We start with the first case. Let i be the largest index of a degenerated shortcut such that the cost of P_i is less than θ. Because the cost of P_i misses the target value by $(1-\beta)/(\ell+1)$ or more, the perceived cost of (v_i, w_i) is at most

$$\tilde{c}_F(v_i, w_i, \beta) \leq (1-\beta)^2 + \beta \cdot \left(\theta - \frac{1-\beta}{\ell+1}\right) = 1 - \frac{\beta \cdot (1-\beta)}{\ell+1} < 1 - \varepsilon.$$

The second inequality holds true by choice of ε. In contrast, all later shortcuts cost θ or more. This implies that the perceived cost of (v_i, v_{i+1}) is at least $\tilde{c}_F(v_i, v_{i+1}, \beta) \geq 1$. Clearly, this contradicts the fact that the agent stays on the regular path of G_F.

As the first case is impossible, only the second one remains, i.e., all degenerated shortcuts cost more than θ. Let v_i be the initial node of such a shortcut and assume that the agent is currently located at the preceding node v' with respect to the regular path. At this point the agent may plan to take one of two types of paths across the initial edge (v', v_i): The first type of path is of the form $P = v', v_i, w_i, \ldots, t$, i.e., it contains a shortcut at v_i. However, the cost of this shortcut exceeds θ by $(1-\beta)/(\ell+1)$ or more. Consequently, the perceived cost of P is at least

$$\tilde{c}(P, \beta) \geq (1-\beta)^3 - \varepsilon + \beta \cdot \left(\theta + \frac{1-\beta}{\ell+1}\right) = 1 - \varepsilon + \frac{\beta \cdot (1-\beta)}{\ell+1} > 1.$$

The last inequality holds true by choice of ε. The second type of path is of the form

$P' = v', v_i, v_{i+1}, \dots, t$, i.e., it contains a regular edge out of v_i. By construction of G, the path P' must cross some other edges after v_{i+1} whose total cost is at least θ. These edges may either be part of a shortcut or they occur at the nodes $v_{\ell+1}$, $v_{\ell+2}$ and $v_{\ell+3}$. As a result, the perceived cost of P' is at least

$$\tilde{c}(P', \beta) \geq (1-\beta)^3 - \varepsilon + \beta \cdot \big((1-\beta)^3 - \varepsilon\big) = 1 + (1-\beta) \cdot \Big(\frac{\beta \cdot (1-\beta)^3}{1+\beta} - \varepsilon\Big) > 1.$$

Again, the last inequality holds true by choice of ε. This means that no matter whether the agent plans along P or P', the perceived cost is not motivating. Therefore, G_F cannot be motivating either. This completes the proof. □

3.4 Two Complementing Approximation Algorithms

Since the design of optimal subgraphs is NP-hard, we turn our attention to approximate solutions. For real-world applications it seems reasonable to assume that an agent's present-bias cannot become arbitrarily small, but stays relatively close to 1. In such a setting, it might be acceptable to compute a subgraph G_F whose minimal motivating reward $\mu(G_F, \beta)$ approximates the optimal reward $\mu^*(G, \beta)$ within a factor of $1/\beta$. The following algorithm presents a straight forward way to compute such a subgraph:

Algorithm 1: CheapestPathApprox

Input: task graph G, present-bias β
Output: edge set F

```
P ← shortest path from s to t;
F ← E \ P;
return F;
```

The idea of this algorithm is simple. As the name "CheapestPathApprox" suggests, it computes a cheapest path P from s to t and removes all edges apart from the ones contained in P. This way, the agent is forced to travel along P, which results in a $1/\beta$ approximation ratio.

Proposition 3.4.1. *Given a task graph G and present-bias $\beta \in (0, 1]$,* CheapestPathApprox *constructs a subgraph G_F whose minimal motivating reward $\mu(G_F, \beta)$ approximates $\mu^*(G, \beta)$ within a factor of $1/\beta$.*

Proof. Let P be the path computed by CheapestPathApprox. At any node v of P, the perceived cost of v satisfies $\tilde{c}_F(v) \leq d_F(v) \leq d(s)$. The minimal motivating reward of G_F is therefore at most $\mu(G_F, \beta) \leq d(s)/\beta$. Conversely, the perceived cost of s is at least $\tilde{c}_{F'}(s) \geq \beta \cdot d_{F'}(s) \geq \beta \cdot d(s)$ for all subgraphs $G_{F'}$, including optimal ones. As a result it holds true that $\mu^*(G, \beta) \geq d(s)$, which in turn yields the desired ratio

$$\frac{\mu(G_F, \beta)}{\mu^*(G, \beta)} \leq \frac{d(s)/\beta}{d(s)} = \frac{1}{\beta}. \qquad \square$$

Apart from the computational properties of CheapestPathApprox, it is interesting to briefly consider the real-world implications of the algorithm. In particular, note that CheapestPathApprox constructs a prohibition structure that imposes a most cost efficient sequence of tasks onto the agent. Of course, this seems to be a rather dictatorial approach to designing incentives. However, in the case of a moderate present-bias the graphical model suggests that such prohibitions actually yield cost efficient results for the agent as well as for the designer who places the reward.

Assuming that we only face a real-world instance of β, the approximation ratio of CheapestPathApprox might be satisfactory. However, from the point of view of computational complexity the ratio is less impressive. The reason is that the value $1/\beta$ of the ratio can grow exponentially in the encoding length of β. Consequently, we need an approximation strategy that performs well for small values of β.

To come up with a suitable strategy, observe that a small β implies a highly present-biased agent, i.e., an agent who cares almost exclusively about her current cost. It seems sensible to guide such an agent along a path P that minimizes the maximal cost along its edge. We call such a path a *minmax path*. To impose the minmax path P, it suffices to remove all other edges of the task graph. This yields the following algorithm:

Algorithm 2: MinMaxPathApprox

Input: task graph G, present-bias β
Output: edge set F

$P \leftarrow$ minmax path from s to t;
$F \leftarrow E \setminus P$;
return F;

It is easy to see that the above algorithm, which we call MinMaxPathApprox, has a linear approximation ratio of $1 + \beta \cdot n$. From a theoretical point of view, this improves upon the exponential approximation ratio of CheapestPathApprox.

Proposition 3.4.2. *Given a task graph G and present-bias $\beta \in (0, 1]$,* MinmaxPathApprox *constructs a subgraph G_F whose minimal motivating reward $\mu(G_F, \beta)$ approximates $\mu^*(G, \beta)$ within a factor of $1 + \beta \cdot n$.*

Proof. Let P be the minmax path computed by MinMaxPathApprox and denote the cost of its most expensive edge by $c(P) = \max\{c(e) \mid e \in P\}$. By definition of P, the agent must encounter an edge e of cost $c(P)$ or more in any subgraph $G_{F'}$ that connects s with t. The preceived cost of e is lower bounded by $\tilde{c}_{F'}(e) \geq c(e) \geq c(P)$. Since this observation also holds true for an optimal subgraph, we conclude that $\mu^*(G, \beta) \geq c(P)/\beta$. Conversely, each edge e of P charges $c(P)$ or less. Taking into account that P has a length of at most $n-1$, the perceived cost of its nodes v does not exceed

$$\tilde{c}_F(v) \leq c(P) + \beta \cdot (n-2) \cdot c(P) \leq c(P) + \beta \cdot n \cdot c(P).$$

This bounds the minimal motivating reward of G_F by $\mu(G_F, \beta) \leq c(P)/\beta + n \cdot c(P)$ and results in the desired approximation ratio of

$$\frac{\mu(G_F, \beta)}{\mu^*(G, \beta)} \leq \frac{c(P)/\beta + n \cdot c(P)}{c(P)/\beta} = 1 + \beta \cdot n.$$ □

Keeping in mind that CheapestPathApprox performs best for high values of β and MinMaxPathApprox for low values of β, the following combination of the two algorithms suggests itself: Run CheapestPathApprox whenever β is greater or equal to some threshold value θ and MinMaxPathApprox whenever β is less than θ. This approach, which is taken from our work on prohibition based incentive design [2], yields an approximation ratio of $1 + \sqrt{n}$ for a suitable choice of θ.

Theorem 3.4.3. *Given a task graph G and present-bias $\beta \in (0, 1]$, it is possible to approximate an optimal subgraph within a factor of $1 + \sqrt{n}$ in polynomial time with respect to the encoding length of G and β.*

Proof. The approximation ratio of $1+\sqrt{n}$ is a direct result of Propositions 3.4.1 and 3.4.2. In fact, it suffices to set the threshold θ to $1/\sqrt{n}$ in the combined approximation strategy described above. It remains to show that CheapestPathApprox and MinMaxPathApprox can be implemented in polynomial time with respect to the encoding length of G and β. For CheapestPathApprox this should be obvious. To see that MinMaxPathApprox also requires only polynomial time, we need an efficient way to construct a minmax path through G. One possible approach is to start with an empty graph and insert the edges of G in non-decreasing order of cost until s and t become connected for the first time. Any path from s to t in the resulting subgraph is a minmax path. Clearly, this requires only polynomial time. □

3.5 The Approximability of Optimal Subgraphs

The combined approximation ratio of the algorithms presented in the previous section is proportional to the square root of the encoding length of the input. Interestingly, no substantial improvement on this ratio is possible unless P = NP. The reason is that the optimization version of MSG, which we call MSG-OPT, is NP-hard to approximate within a factor less than $\sqrt{n}/3$. This suggests that Theorem 3.4.3 is tight apart from a small constant factor; a result that is particularly surprising considering the relatively simple algorithmic techniques employed to establish the theorem in the first place.

Definition 3.5.1 (MSG-OPT)**.** The problem of computing $\mu^*(G, \beta)$ for a given task graph G and present-bias $\beta \in (0, 1]$.

Our argument is based on a generalized version of the reduction from ℓ-DCP constructed in the proof of Theorem 3.3.1. To facilitate the analysis of the reduction, the following Lemma turns out to be helpful.

Lemma 3.5.1. *For any integer $k \geq 1$ it holds true that*

$$\left(1 - \frac{1}{3 \cdot k + 3}\right)^{3 \cdot k + 3} > \frac{1}{3}.$$

Proof. Note that the function $f(x) = (1 - 1/x)^x$ is monotonically increasing for $x \geq 1$. As a result, it holds true that

$$\left(1 - \frac{1}{3 \cdot k + 3}\right)^{3 \cdot k + 3} = f(3 \cdot k + 3) \geq f(6) = \frac{15625}{46656} > \frac{1}{3}. \qquad \square$$

We are now ready to conclude this chapter by establishing the claimed hardness result for MSG-OPT. As in the previous sections, the proof of the following theorem is taken from our work on prohibition based incentive design [2].

Theorem 3.5.2. *MSG-OPT is NP-hard to approximate within a ratio less than $\sqrt{n}/3$.*

Proof. Let $\mathcal{I}$ be an arbitrary instance of ℓ-DCP that consists of a directed acyclic graph H and ℓ disjoint node pairs $(s_1, t_1), \ldots, (s_\ell, t_\ell)$. Furthermore, let $k > 1$ be an arbitrary positive integer that is polynomial in the encoding length of $\mathcal{I}$. We determine the best choice of k with respect to $\mathcal{I}$ at a later point in time. Our goal is to reduce $\mathcal{I}$ to an MSG instance $\mathcal{J}$ on a task graph $G = (V, E, c, r)$ with the following two properties:

(a) If $\mathcal{I}$ has ℓ node-disjoint connecting paths, then $\mathcal{J}$ has a motivating subgraph for a reward of $r = 1/\beta$, i.e., $\mu^*(G, \beta) \leq 1/\beta$.

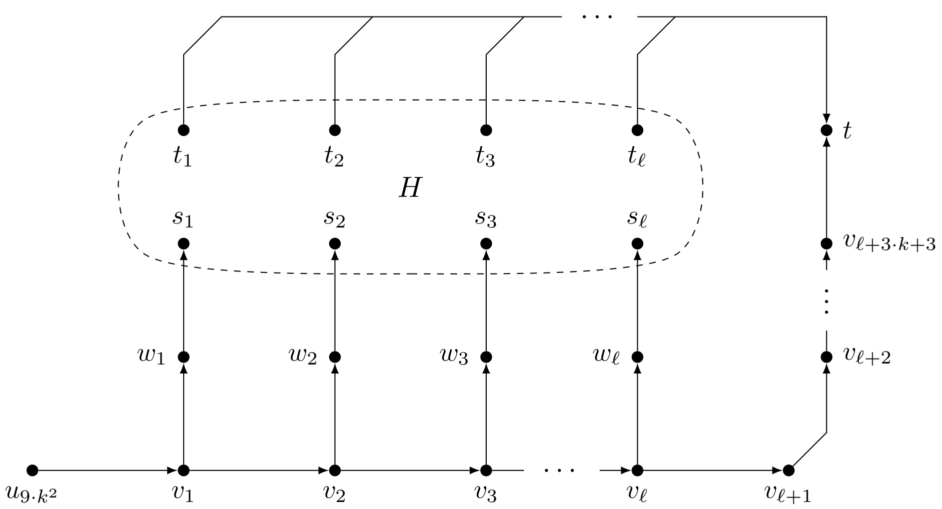

Figure 3.6: Embedding unit of an ℓ-DCP graph H

(b) If $\mathcal{I}$ does not have ℓ node-disjoint connecting paths, then $\mathcal{J}$ has no motivating subgraph for a reward $r \leq k/\beta$, i.e., $\mu^*(G, \beta) \geq k/\beta$..

Consequently, any algorithm approximating the minimal motivating reward $\mu^*(G, \beta)$ within a ratio of k or less must also solve $\mathcal{I}$. Recall that Proposition 3.4.1 proves the existence of a polynomial time algorithm that approximates $\mu^*(G, \beta)$ within a factor of $(1/\beta)$. As a result, the present-bias β cannot be independent of $\mathcal{I}$. Rather, β must be a value less than $1/k$. For convenience, let $\beta = 1/(3 \cdot k + 3)$.

We are now ready to specify the task graph G. In general, G consists of two parts: the *embedding unit* and the *amplification unit.* As Figure 3.6 shows, the structure of the embedding unit is closely related to the reduction presented in the proof of Theorem 3.3.1. In particular, the embedding unit consists of a regular path from the node $u_{9\cdot k^2}$ to the target node t along the intermediate nodes $v_1, \dots, v_{\ell+3\cdot k+3}$ as well as ℓ shortcuts, which are routed through the embedding of H. However, the edge cost of the regular path and its shortcuts is a bit different from Theorem 3.3.1. In case of the regular path the first $\ell + 1$ edges have a cost of $(1-\beta)^{3\cdot k+3} - \varepsilon$, where ε is a positive value satisfying

$$\varepsilon < \min\Big\{\frac{\beta \cdot (1-\beta)^{3\cdot k+1}}{\ell+1}, \frac{\beta \cdot (1-\beta)^{3\cdot k+3}}{1+\beta}, \frac{1}{1+k}, (1-\beta)^{3\cdot k+3} - \frac{1}{3}\Big\}.$$

Note that the term $(1-\beta)^{3\cdot k+3} - 1/3$ in the above expression is a positive quantity according to Lemma 3.5.1. Once the regular path passed the last shortcut, its edge cost steadily increases towards t. More precisely, each edge (v_i, v_{i+1}) with $i > \ell$ and

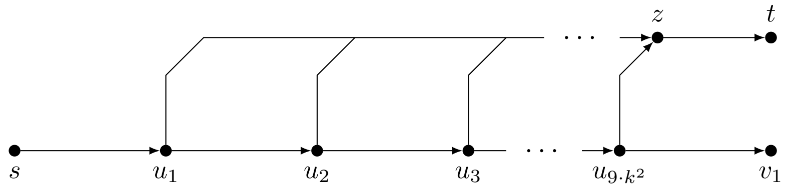

Figure 3.7: Amplification unit

$t = v_{\ell+3\cdot k+4}$ has a cost of $(1-\beta)^{\ell+3\cdot k+3-i}$. To conclude the embedding unit, we consider the edge cost of the individual shortcuts. The initial edge (v_i, w_i) of shortcut i charges $(1-\beta)^{3\cdot k+2}$ while the edges (w_i, s_i) and (t_i, t) have a complementing cost of $(\ell+1-i)(1-\beta)^{3\cdot k+1}/(\ell+1)$ and $i \cdot (1-\beta)^{3\cdot k+1}/(\ell+1) + \sum_{j=0}^{3\cdot k}(1-\beta)^j$ respectively. All edges of the embedding of H are free of charge. The total cost of the shortcut therefore evaluates to $\sum_{j=0}^{3\cdot k+2}(1-\beta)^j$.

Next, we take a closer look at the amplification unit, which is depicted in Figure 3.7. The amplification unit is connected to the embedding unit via the edge $(u_{9\cdot k^2}, v_1)$ and extends the regular path by an additional segment from s to $u_{9\cdot k^2}$ along the nodes $u_1, \ldots, u_{9\cdot k^2-1}$. Similar to the embedding unit, each edge of the regular path in the amplification unit has a cost of $(1-\beta)^{3\cdot k+3} - \varepsilon$. Furthermore, there is a shortcut from each node u_i to t via a common node z. The initial edge (u_i, z) of this shortcut has a cost of $(1-\beta)^{3\cdot k+2}$ while the second edge (z, t) charges $\sum_{j=0}^{3\cdot k+1}(1-\beta)^j$. Just like in the embedding unit the total cost of a shortcut in the amplification unit evaluates to $\sum_{j=0}^{3\cdot k+2}(1-\beta)^j$. It is easy to see that the resulting task graph G is acyclic and its encoding length is polynomial in the encoding length of $\mathcal{I}$.

In the following, we prove that $\mathcal{J}$ satisfies property (a), i.e., $\mathcal{J}$ admits a subgraph that is motivating for a reward of $r = 1/\beta$ whenever $\mathcal{I}$ has ℓ node-disjoint connecting paths. For this purpose, assume that ℓ node-disjoint connecting paths exist in H and consider the subgraph G_F obtained by deleting all edges of H that are not part of such a path. An argument similar to the one given in the proof of Theorem 3.3.1 shows that whenever the agent is located at a node of the regular path with an outgoing shortcut, the perceived cost of this shortcut exceeds the perceived cost of the outgoing regular edge by a value of ε. Therefore, the agent has no incentive to leave the regular path. Furthermore, the perceived cost is at most 1 at any node of the regular path. This means that the given reward is sufficient for G_F to be motivating.

We continue with (b), i.e., no subgraph of $\mathcal{J}$ is motivating for a reward of $r \leq k/\beta$ if $\mathcal{I}$ does not have ℓ node-disjoint connecting paths. To prove this, let G_F be an arbitrary subgraph of G and assume that no ℓ node-disjoint connecting paths exist in H. Our goal is to show G_F cannot be motivating. Without loss of generality, we may assume that the agent does not leave the regular path in G_F. The reason is that all other paths to t contain an edge with a perceived cost strictly greater than $\sum_{j=0}^{3\cdot k}(1-\beta)^j$, namely (z,t) or an edge of the form (t_i,t). According to Lemma 3.5.1, this cost is greater than

$$\sum_{j=0}^{3\cdot k}(1-\beta)^j = \sum_{j=0}^{3\cdot k}\Big(1-\frac{1}{3\cdot k+3}\Big)^j > \sum_{j=0}^{3\cdot k}\Big(1-\frac{1}{3\cdot k+3}\Big)^{3\cdot k+3} > \sum_{j=0}^{3\cdot k}\frac{1}{3} = k,$$

which is not covered by the perceived reward. Therefore, any subgraph G_F in which the agent leaves the regular path cannot be motivating and needs no further consideration. In particular, we may assume that G_F contains the entire regular path.

Similar to the proof of Theorem 3.3.1, we call the i-th shortcut of G_F in the embedding unit degenerated if the cost of a cheapest path from v_i to t via w_i is different from the target value $\theta = \sum_{j=0}^{3\cdot k+2}(1-\beta)^j$. By construction of G, all degenerated shortcuts miss the target value by $(1-\beta)^{3\cdot k+1}/(\ell+1)$ or more. Recall that $\mathcal{I}$ is assumed to have no solution. Following the same argument presented in the proof of Theorem 3.3.1, this implies that some degenerated shortcut must exist in G_F. Furthermore, the proof of Theorem 3.3.1 also implies that these shortcuts cannot cost less than θ if we want the agent to stay on the regular path. As a result, we can make a case distinction on whether the first shortcut of the embedding unit is degenerated or not.

If the first shortcut is not degenerated, then there exists an integer i with $1 < i \leq \ell$ such that the $(i-1)$-st shortcut is not degenerated, but shortcut i is. Let P denote the path that the agent plans to take at v_{i-1}. If the initial edge of P is the shortcut edge (v_{i-1}, w_{i-1}), then the perceived cost of P in G_F evaluates to 1 according to Lemma 3.2.2. Conversely, if the initial edge of P is the regular edge (v_{i-1}, v_i), then P may either contain the shortcut at v_i or another regular edge (v_i, v_{i+1}). Because the i-th shortcut is degenerated and exceeds the target value θ by at least $(1-\beta)^{3\cdot k+1}/(\ell+1)$, the perceived cost of P in the first case is at least

$$\begin{aligned}\tilde{c}_F(P,\beta) &\geq (1-\beta)^{3\cdot k+3} - \varepsilon + \beta\cdot\Big(\frac{(1-\beta)^{3\cdot k+1}}{\ell+1} + \sum_{j=0}^{3\cdot k+2}(1-\beta)^j\Big)\\ &= 1 + \frac{\beta\cdot(1-\beta)^{3\cdot k+1}}{\ell+1} - \varepsilon > 1.\end{aligned}$$

The second inequality holds true by choice of ε. Otherwise, if P does not contain the degenerated shortcut at v_i, then P may or may not be routed along a shortcut at some subsequent node of the regular path. Either way, considering that no shortcut of the subgraph G_F costs less than the target value θ, we deduce that the perceived cost of P must be at least

$$\begin{aligned}\tilde{c}_F(P,\beta) &\geq (1-\beta)^{3\cdot k+3} - \varepsilon + \beta \cdot \Big((1-\beta)^{3\cdot k+3} - \varepsilon + \sum_{j=0}^{3\cdot k+2} (1-\beta)^j\Big) \\ &= 1 + (1+\beta)\Big(\frac{\beta \cdot (1-\beta)^{3\cdot k+3}}{1+\beta} - \varepsilon\Big) > 1.\end{aligned}$$

Again, the second inequality holds by choice of ε. However, this means that the agent strictly prefers the shortcut at v_{i-1} over the regular edge at v_{i-1}. This contradicts the assumption that the agent stays on the regular path.

We now consider the case in which the first shortcut of the embedding unit is degenerated. Let i be the highest index of a node on the regular path of the amplification unit such that u_i is connected to t via a direct shortcut in G_F. The perceived cost of this shortcut is 1 when viewed from u_i. However, according to the previous paragraph, all paths P along (u_i, u_{i+1}), or $(u_{9\cdot k^2}, v_1)$ in the special case of $i = 9 \cdot k^2$, have a perceived net cost greater than 1. This contradicts the assumption that the agent stays on the regular path. But if no u_i has a direct shortcut to t, then the perceived cost of s is

$$\begin{aligned}\tilde{c}_F(s,\beta) &\geq (1-\beta)^{3\cdot k+3} - \varepsilon + \beta \cdot \Big(9 \cdot k^2 \cdot \big((1-\beta)^{3\cdot k+3} - \varepsilon\big) + \sum_{j=0}^{3\cdot k+2} (1-\beta)^j\Big) \\ &= 1 - \varepsilon + \beta \cdot 9 \cdot k^2 \cdot \big((1-\beta)^{3\cdot k+3} - \varepsilon\big).\end{aligned}$$

Taking into account that $\beta = 1/(3 \cdot k + 3)$ we can further simplify this term to

$$\begin{aligned}1 - \varepsilon + \beta \cdot 9 \cdot k^2 \cdot \big((1-\beta)^{3\cdot k+3} - \varepsilon\big) &= 1 - \varepsilon + 9 \cdot k^2 \cdot \frac{1/3 + ((1-\beta)^{3\cdot k+3} - 1/3 - \varepsilon)}{3 \cdot k + 3} \\ &> 1 - \varepsilon + 9 \cdot k^2 \cdot \frac{1/3}{3 \cdot k + 3} \\ &= k + \Big(\frac{1}{1+k} - \varepsilon\Big) > k.\end{aligned}$$

Putting everything together, we may conclude that G_F is not motivating for the given reward. This proves property (b).

To complete the proof, we must determine a suitable value for k. For this purpose let k be the number of nodes in H. Clearly, this value is polynomial in the encoding length of $\mathcal{I}$. The total number of nodes in G is

$$n = 2 + (9 \cdot k^2 + 1) + (k + 2 \cdot \ell + 3 \cdot k + 3).$$

The first term of this sum accounts for s and t, the second for the amplification unit and the third for the embedding unit. Considering that $\ell \leq k$, we get $n \leq 9 \cdot k^2 + 6 \cdot k + 6$. However, this bound on n is less than $(9 + \varepsilon') \cdot k^2$ for any positive value ε' if only H is large enough. In fact, it suffices to assume that H has $k \geq (\sqrt{6 \cdot \varepsilon' + 9} + 3)/\varepsilon'$ nodes. As a result, $n \leq (9 + \varepsilon') \cdot k^2$ holds true and implies $k \geq \sqrt{n/(9 + \varepsilon')}$. But this means that MSG-OPT is NP-hard to approximate within a ratio less than $\sqrt{n}/3$. $\square$

4 Penalty Based Incentives

Having encountered fundamental computational challenges in the design of efficient prohibitions, we turn our attention to more general incentives that are based on penalties. Similar to the previous chapter, our goal is to reduce the reward required to keep the agent motivated. But instead of prohibiting her from performing certain tasks, we now merely assign penalty fees to them. As a result, we obtain a more powerful incentive design tool that exhibits more favorable computational properties at the same time.

4.1 The Seminar

To introduce the concept of penalty based incentives, it is instructive to consider another one of our stories. This time, the protagonist Alice signs up for a 10-week seminar at her university. To pass the seminar, Alice can do one of two things. Either she solves a homework exercise each week or she gives a presentation at the end of the seminar. The homework has a weekly cost of 1 while the presentation has a one-time cost of 3. Due to organizational matters, Alice needs to sign up for the presentation in advance. This procedure has no cost. However, if she omits any of the homework exercises before she signs up for the presentation, she fails the course. Figure 4.1 models this scenario as a task graph $G = (V, E, c, r)$. The nodes v_i of G represent the individual weeks $i \in \{1, \dots, 10\}$ with $s = v_1$ while the edges (v_i, w) and (v_i, v_{i+1}) with $t = v_{11}$ correspond to the choice between signing up for the presentation or solving a homework exercise.

Assuming that Alice has a present-bias of $\beta = 1/3$, she experiences a perceived cost of $\tilde{c}(s, w, \beta) = 0 + \beta \cdot 3 = 1$ if she signs up for the presentation. In contrast, the perceived cost of doing a home work exercise is $\tilde{c}(s, w, \beta) = 1 + \beta \cdot 3 = 2$. This cost is realized if Alice plans to sign up for the presentation one week later. As a result, she opts for the presentation in the very first week and pays a cost of 3 at the end of the seminar. Clearly, this is the cheapest path through G that Alice can take with respect to the total cost she pays. But what about the required reward?

Keeping in mind that Alice's signs up for the presentation immediately, the minimal motivating reward of G is $\mu(G, \beta) = \tilde{c}(w, t, \beta)/\beta = 9$. Furthermore, it is not possible to

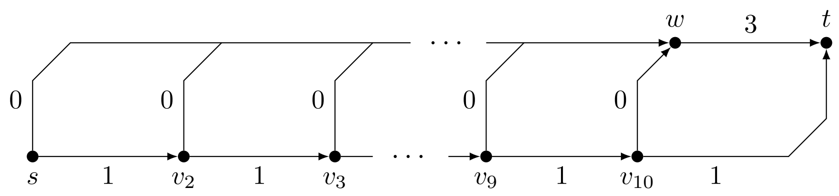

Figure 4.1: Task graph of Alice's seminar

reduce this reward with prohibition based incentives. The reason is that prohibitions can only keep Alice from visiting the node w if they disconnect each path of the form v_i, w, t. However, without these paths, the only path from s to t that remains is $s, v_2, \ldots, v_{10}, t$. The perceived cost of s therefore becomes $1 + \beta \cdot 9 = 4$, which requires a reward of at 12 to be motivating. Clearly, this is more expensive than simply omitting the prohibitions in the first place.

Considering that prohibitions cannot improve the minimal motivating reward of G, the question arises whether penalty fees can. To investigate this question, consider a scenario in which the sign up process for the presentation is not free, but requires to write a lengthy proposal. Let the cost of this proposal be $3/2+\varepsilon$ where ε denotes an arbitrarily small but positive quantity. The cost of the proposal can be interpreted as a penalty fee that is charged whenever the agent tries to sign up for the presentation. In the graphical model, we formalize this notion by assigning an *extra cost* of $h(v_i, w) = 3/2 + \varepsilon$ to each edge (v_i, w). We call the resulting assignment h of non-negative cost to edges a *cost configuration*. Applying h to G yields a new task graph $G_h = (V, E, c + h, r)$.

Definition 4.1.1 (Cost Configuration)**.** Assigning a cost configuration $h : E \to \mathbb{R}_{\geq 0}$ to a task graph $G = (V, E, c, r)$ yields a new task graph $G_h = (V, E, c + h, r)$ whose edge cost is defined as $(c + h)(e) = c(e) + h(e)$.

Taking a closer look at the new task graph G_h, it is easy to see that the perceived cost of signing up for the presentation in any given week i, i.e., the perceived cost of an edge (v_i, w), now evaluates to

$$\tilde{c}_h(v_i, w, \beta) = c(v_i, w) + h(v_i, w) + \beta \cdot \big(c(w, t) + h(w, t)\big) = \frac{3}{2} + \varepsilon + \beta \cdot 3 = \frac{5}{2} + \varepsilon.$$

Conversely, solving the homework exercise of week i has a perceived cost of at most $\tilde{c}_h(v_i, v_{i+1}, \beta) \leq 1 + \beta \cdot (3/2 + \varepsilon + 3) = 5/2 + \beta \cdot \varepsilon$. This cost is realized if Alice plans to sign up for the presentation next week. In the special case of $i = 10$, she may simply complete the seminar by turning in her last homework. Note that we write $\tilde{c}_h$ instead of

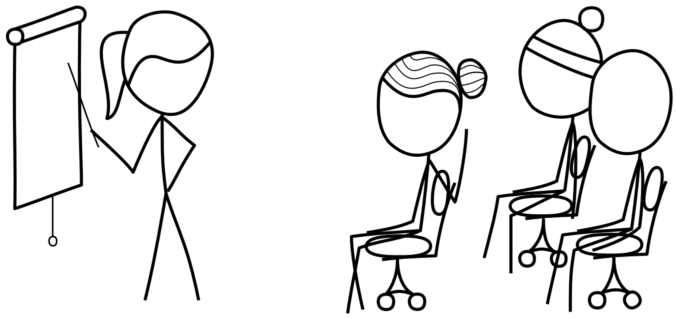

Figure 4.2: Alice giving a presentation

$\tilde{c}$ to denote the perceived cost with respect to a specific cost configuration h. Similarly, we write d_h to denote the cost of a cheapest paths with respect to h. The fact that $\tilde{c}_h(v_i, w, \beta) = 5/2 + \varepsilon > 5/2 + \beta \cdot \varepsilon \geq \tilde{c}_h(v_i, v_{i+1}, \beta)$ implies that Alice always prefers solving the current homework exercise to signing up for the presentation. As a result, her walk through G_h leads her along the path $s, v_2, \ldots, v_{10}, t$. Since she never incurs cost greater than $5/2 + \beta \cdot \varepsilon$ on this path, a reward of $(5/2 + \beta \cdot \varepsilon)/\beta = 15/2 + \varepsilon$ is sufficient to make G_h motivating. Assuming that ε is sufficiently small, i.e., $\varepsilon < 3/2$, this is less expensive than any result we may hope to achieve by prohibiting certain tasks.

4.2 Prohibition versus Penalty Fees

The scenario presented in the previous section suggests that penalty based incentives are sometimes more powerful than prohibition based incentives when it comes to motivating present-biased agents cost efficiently. To formalize this observation, we need to take a closer look at the notion of minimum motivating rewards. In the context of penalty fees, let the minimum motivating reward $\mu^*(G, \beta)$ be defined as follows:

Definition 4.2.1 (Minimum Motivating Reward)**.** The minimum motivating reward $\mu^*(G, \beta)$ is the infimum over all rewards for which the task graph G admits a cost configuration that is motivating for an agent with present-bias $\beta \in (0, 1]$.

A curious technicality of the above definition is its use of an infimum rather than a minimum. In the case of prohibition based incentives this is unnecessary since $\mu^*(G, \beta)$ is drawn from a finite set of values. More precisely, $\mu^*(G, \beta)$ is drawn from the set of all values that correspond to the minimal motivating reward $\mu(G_F, \beta)$ of a particular subgraph G_F. However, penalty based incentives behave differently. The reason is that G admits an infinite number of cost configurations. Consequently, G may not have an optimal cost configuration.

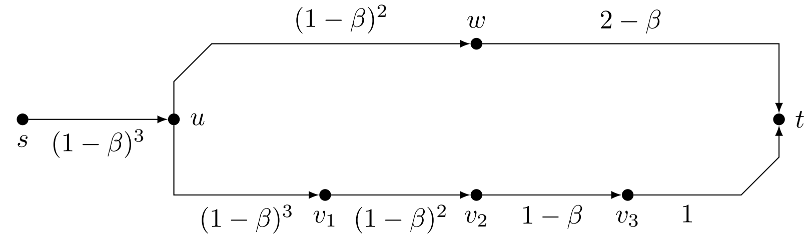

Figure 4.3: Task graph with no optimal cost configuration

To give a concrete example of such a scenario, consider the task graph G depicted in Figure 4.3 and an agent whose present-bias is some value $\beta \in (0, 1)$. By construction of G, the agent is indifferent between the edges (u, v_1) and (u, w) when located at u. In both cases the perceived cost evaluates to 1. If she chooses the edge (u, w), she perceives a cost of $2-\beta$ at the next node w. Conversely, if she crosses (u, v_1), she perceives a cost of 1 at each node v_i. Since $\beta < 1$, (u, v_1) is clearly the better choice. However, to break the tie between (u, w) and (u, v_1), some positive extra cost of ε must be assigned to the path u, w, t. As a result, the perceived cost of s becomes $\min\{1 + \beta \cdot \varepsilon, 1 + \beta(1-\beta)^3\}$. The first term of the minimum corresponds to the path along w while the second one corresponds to the path along v_1, v_2 and v_3. The perceived cost of s implies that G admits a motivating cost configuration for all rewards strictly greater than $1/\beta$ as long as the tie breaker ε is sufficiently small. However, because ε must not become 0, no reward configuration is motivating for a reward of exactly $1/\beta$. Nevertheless we say that $1/\beta$ is the minimum motivating reward of G, i.e., $\mu^*(G, \beta) = 1/\beta$.

Having established the notion of minimum motivating rewards in the context of penalty based incentives as well as prohibition based incentives, we can finally compare the cost efficiency of the two incentives formally. To avoid confusion, let $\mu^*_{\text{pnl}}(G, \beta)$ denote the minimum motivating reward for penalty fees and $\mu^*_{\text{prb}}(G, \beta)$ the minimum motivating reward for prohibitions. We keep this notation until the end of this section. Given a specific task graph G and present-bias β, the cost efficiency ratio $\mu^*_{\text{prb}}(G, \beta)/\mu^*_{\text{pnl}}(G, \beta)$ indicates how a strategic choice of penalty fees compares to optimal prohibitions. According to the following theorem, which is taken from our previous work on penalty based incentives [3], it turns out that penalty fees are at least as efficient as prohibitions and under certain circumstance up to $1/\beta$ times more efficient.

Theorem 4.2.1. *The cost efficiency ratio $\mu^*_{\text{prb}}(G, \beta)/\mu^*_{\text{pnl}}(G, \beta)$ lies between 1 and $1/\beta$ for all task graphs G and present-bias values $\beta \in (0, 1]$. This result is tight.*

Proof. The lower bound of 1 is a direct consequence of the fact that each subgraph G_F

can be emulated by a cost configuration h. For this purpose, it is sufficient to assign large enough extra cost to all edges $e \in F$, e.g. $h(e) = 1 + \sum_{e \in E} c(e)/\beta$. Clearly, this does not change the perceived cost of any path P not using edges of F. Moreover, the agent has no incentive to prefer an alternative path P' over P if it uses edges of F, i.e.,

$$\tilde{c}_h(P', \beta) \geq \beta \cdot \Big(1 + \sum_{e \in E} \frac{c(e)}{\beta}\Big) > \sum_{e \in E} c(e) \geq \sum_{e \in P} c(e) \geq \tilde{c}_h(P, \beta).$$

As a result, the agent is bound to make identical decisions in G_F and G_h whenever s and t are not disconnected by F. Otherwise, if s and t are disconnected, G_h is more cost efficient than G_F anyway. This establishes the lower bound of 1 on the cost efficiency ratio. To see that this bound is tight, consider a task graph G consisting of just a single edge. Because it is of no advantage to delete or assign extra cost to this edge, it immediately follows that $\mu^*_{\text{prb}}(G, \beta)/\mu^*_{\text{pnl}}(G, \beta) = 1$.

Now, we take a closer look at the upper bound. To establish a bound of $1/\beta$, let G be an arbitrary task graph and consider the subgraph G_F whose only edges are those of a cheapest path P from s to t. This way the agent has no choice but to follow P if she wants to reach t. Considering that her perceived cost along the way is bounded by the actual cost of P, i.e., bounded by $d(s)$, we may conclude that $\mu^*_{\text{prb}}(G, \beta) \leq d(s)/\beta$. Next, consider an arbitrary cost configuration h. Because h can only increase but never decrease the edge cost, the perceived cost at s with respect to h is at least $\beta \cdot d(s)$. Consequently, it holds true that $\mu^*_{\text{pnl}}(G, \beta) \geq d(s)$. This yields the desired upper bound of $\mu^*_{\text{prb}}(G, \beta)/\mu^*_{\text{pnl}}(G, \beta) \leq 1/\beta$.

It remains to show that the upper bound is tight. For this purpose, we construct a task graph G that is a slightly modified version of the task graph modeling Alice's seminar, i.e., the task graph depicted in Figure 4.1. More precisely, let $m = \lceil \beta^{-2} \cdot (1-\beta)^{-1} \cdot \varepsilon^{-2} \rceil$ and assume that G contains a path $v_1, \ldots, v_{2 \cdot m+1}$ whose edges are all of cost $(1-\beta) \cdot \varepsilon^2$. We call this the *regular path* and set $s = v_1$ and $t = v_{2 \cdot m+1}$. In addition to the regular path each v_i with $i \leq 2 \cdot m$ has a *shortcut* to t via a common node w. The edges (v_i, w) are free of charge, whereas (w, t) has a cost of $1/\beta$.

We proceed to argue that $\mu^*_{\text{prb}}(G, \beta) \geq 1/\beta^2$. For the sake of contradiction, assume the existence of a subgraph G_F that is motivating for a reward $1/\beta^2$. In this case the agent must not take shortcuts as her perceived cost at w exceeds her perceived reward. Therefore, she must follow the regular path. In particular, she must visit each node v_i on the first half of the path, i.e. $i \leq m+1$. Let $P = v_i, v_{i+1}, \cdots, t$ be the path she plans to take. There are two conceivable versions of P. On the one hand, P may contain a shortcut at some point. In this case P also contains the edge (w, t) and has a perceived cost of at least $\tilde{c}_F(P, \beta) > \beta \cdot c(w, t) = 1$. On the other hand, P may avoid all shortcuts. In this case P is routed via the regular path. But this means that P contains at least m regular edges, each of which contributes a fraction of $\beta \cdot (1 - \beta) \cdot \varepsilon^2$ or more to the perceived cost of P. The total perceived cost of P therefore evaluates to

$$\tilde{c}_F(P, \beta) \geq m \cdot \beta \cdot (1 - \beta) \cdot \varepsilon^2 = \lceil \beta^{-2} \cdot (1 - \beta)^{-1} \cdot \varepsilon^{-2} \rceil \cdot \beta \cdot (1 - \beta) \cdot \varepsilon^2 \geq 1/\beta \geq 1.$$

Due to the perceived cost of P, we conclude that (v_i, v_{i+1}) has a perceived cost of at least 1. However, the perceived cost of the shortcut from v_i to t is also 1 if it is not interrupted. Since the agent must not take a shortcut in G_F, we conclude that all of the first $m+1$ shortcuts are interrupted in G'. As a result, all paths from s to t contain at least m regular edges and the perceived cost of s evaluates to $\tilde{c}_F(s, \beta) \geq m \cdot \beta(1 - \beta) \cdot \varepsilon^2 \geq 1/\beta$. This contradicts the assumption that G_F is motivating for a reward strictly less than $1/\beta^2$ and therefore $\mu^*_{\text{prb}}(G, \beta) \geq 1/\beta^2$ holds true.

To complement this result, we construct a cost configuration h that is motivating for a reward of $(1 + \varepsilon)/\beta$. We achieve this by adding an extra cost of ε to all shortcut edges (v_i, w). To upper bound the perceived cost of (v_i, v_{i+1}), assume that the agent plans to take a shortcut at the next node v_{i+1}. Assuming $\varepsilon < 1$, this yields a perceived cost of

$$\tilde{c}_h(v_i, v_{i+1}, \beta) \leq (1 - \beta) \cdot \varepsilon^2 + \beta \cdot \left(\varepsilon + \frac{1}{\beta}\right) < 1 + \varepsilon$$

for all $i < 2 \cdot m$. In the special case of $i = 2 \cdot m$ the inequality remains valid since $\tilde{c}_h(v_i, v_{i+1}, \beta) = (1 - \beta) \cdot \varepsilon^2 < 1 + \varepsilon$. Conversely, the perceived cost of the shortcut at v_i is at least $\tilde{c}_h(v_i, w, \beta) = \varepsilon + \beta \cdot 1/\beta = 1 + \varepsilon$ for all $i \leq 2 \cdot m$. The agent is therefore never tempted to divert from the regular path. Moreover, she never experiences a cost greater than $1 + \varepsilon$, which implies that $\mu^*_{\text{pnl}}(G, \beta) \leq 1/\beta$. This concludes the proof. □

4.3 Computing Motivating Cost Configurations

Having established the conceptual power of penalty based incentives, we turn our attention to their computational properties. As in the last chapter, our design objective is to keep the agent motivated for as little reward as possible. To express it more formally, given a task graph G and present-bias β, we try to construct a cost configurations h that matches the minimum motivating reward up to a predefined precision-parameter $\varepsilon > 0$, i.e., $\mu(G_h, \beta) - \varepsilon \leq \mu^*(G, \beta)$. We call such a cost configuration *almost optimal*. The reason why we are interested in almost optimal rather than optimal cost configurations lies in the fact that the latter may not exist in G as shown in the previous section.

In order to get a first glimpse of the complexity inherent to the computational task at hand, we formulate a decision version of the problem called MOTIVATING COST CONFIGURATION (MCC):

Definition 4.3.1 (MCC)**.** The problem of deciding whether a task graph $G = (V, E, c, r)$ admits a motivating cost configuration h for a given present-bias $\beta \in (0, 1]$.

In contrast to prohibition based incentives, it is not straight forward to solve MCC via an exhaustive search that considers all possible cost configurations of the given task graph G. The reason is that G admits an infinite number of such cost configurations. At first sight, MCC may therefore seem to be a harder problem than MSG. However, a careful analysis reveals that MCC is contained in the complexity class NP. Keeping in mind that MSG is also contained in NP, this implies that MCC is in fact not harder than MSG from a complexity theoretic perspective.

Proposition 4.3.1. *MCC is contained in NP.*

Proof. Let $G = (V, E, c, r)$ be the task graph of an MCC instance $\mathcal{I}$ and assume that $\mathcal{I}$ admits a motivating cost configuration. Of course, any such cost configuration can be used as a certificate to verify that $\mathcal{I}$ is indeed feasible. However, it is not immediately obvious that one of these certificates can be encoded in polynomial space with respect to $\mathcal{I}$. After all, cost configurations may assign arbitrary cost to the edges of G. To prove the proposition, we must therefore argue that a motivating cost configuration with polynomial encoding length exists.

For this purpose, let h^* be an arbitrary motivating cost configuration and let W be the set of nodes that an agent with present-bias β visits in G_{h^*}. Furthermore, let $P_v = v, \ldots, t$ be a path that minimizes the perceived cost at v with respect to h^*, i.e., $\tilde{c}_{h^*}(v, \beta) = \tilde{c}_{h^*}(P_v, \beta)$. In this case, any cost configuration h that satisfies the following constraints must be a motivating cost configuration:

$$
\begin{aligned}
\tilde{c}_h(P_v, \beta) &\leq \beta \cdot r && \text{for all } v \in W \setminus \{t\} \\
\tilde{c}_h(P_v, \beta) &< \tilde{c}_h(P, \beta) && \text{for all } v \in W \setminus \{t\} \\
& && \text{and all } P = v, u, \ldots, t \text{ such that } u \notin W \\
h(e) &\geq 0 && \text{for all } e \in E.
\end{aligned}
$$

By definition of $\tilde{c}_h$, it should be easy to see that all of the above inequalities are linear with respect to the extra cost h. Consequently, if we replace all strict inequalities in the second set of constraints by non-strict inequalities, we can interpret the result as a linear program without a specific objective function. Clearly, the feasible region of this linear program is non-empty as it contains h^*. As a result, the program also contains a *basic solution*. Since each constraint that is active in this basic solution can be encoded in polynomial space with respect to $\mathcal{I}$, the basic solution itself can also be encoded in polynomial space.

The only thing left to do is to recover the strict inequalities in the second set of constraints. For this purpose, let $h_{v,P}$ be a solution to the linear program that maximizes the difference between $\tilde{c}_h(P,\beta)$ and $\tilde{c}_h(P_v,\beta)$ for a given node $v \in W \setminus \{t\}$ and path $P = v, u, \ldots, t$ such that $u \notin W$. By choice of the objective function, $h_{v,P}$ must satisfy $\tilde{c}_h(P_v,\beta) < \tilde{c}_h(P,\beta)$. Furthermore, we may assume that $h_{v,P}$ can be encoded in polynomial space with respect to $\mathcal{I}$.

Now, let $\bar{h}$ be the arithmetic mean over $h_{v,P}$ for all possible combinations of v and P. Clearly, $\bar{h}$ is a feasible solution of the linear program that satisfies each inequality of the second set of constraints in the strict sense. Consequently, $\bar{h}$ is a suitable certificate for $\mathcal{I}$ if its encoding length is polynomial in $\mathcal{I}$. But because the encoding length of each solution $h_{v,P}$ is polynomial in $\mathcal{I}$ and there are at most exponentially many such solutions, it follows that $\bar{h}$ can indeed be encoded in polynomial space. □

Conversely, MCC does not appear to be substantially easier than MSG either. Similar to MSG, the problem is trivial to solve in polynomial time whenever $\beta = 1$ or $r = 0$. The approach is the same as in Section 3.2. However, for all other parametrizations of β and r MCC turns out to be NP-hard. The following proof of this observation is based on a reduction from 3-SAT that is closely related to the reduction presented in the proof of Theorem 3.2.3. The reduction can also be found in [3].

Theorem 4.3.2. *MCC is NP-hard for any present-bias $\beta \in (0,1)$ and reward $r > 0$.*

Proof. Let $\mathcal{I}$ be a 3-SAT instance, consisting of ℓ clauses $c_1, \ldots, c_\ell$ over m variables $x_1, \ldots, x_m$. To establish the theorem, we reuse the reduction from 3-SAT to MSG from the previous chapter. This means $\mathcal{I}$ is mapped to a task graph $G = (V, E, c, r = 1/\beta)$ similar to the construction of the proof of Theorem 3.2.3. The only difference is that the single edge shortcuts at the boolean nodes of G are omitted this time. Figure 4.4 depicts the resulting MCC instance $\mathcal{J}$ for a simple sample instance of $\mathcal{I}$.

Assuming that $\beta \in (0,1)$ is a fixed value independent of $\mathcal{I}$, our goal is to establish the following two properties:

(a) If $\mathcal{I}$ is satisfiable, then $\mathcal{J}$ must admit a motivating cost configuration.

(b) If $\mathcal{J}$ admits a motivating cost configuration, then $\mathcal{I}$ must be satisfiable.

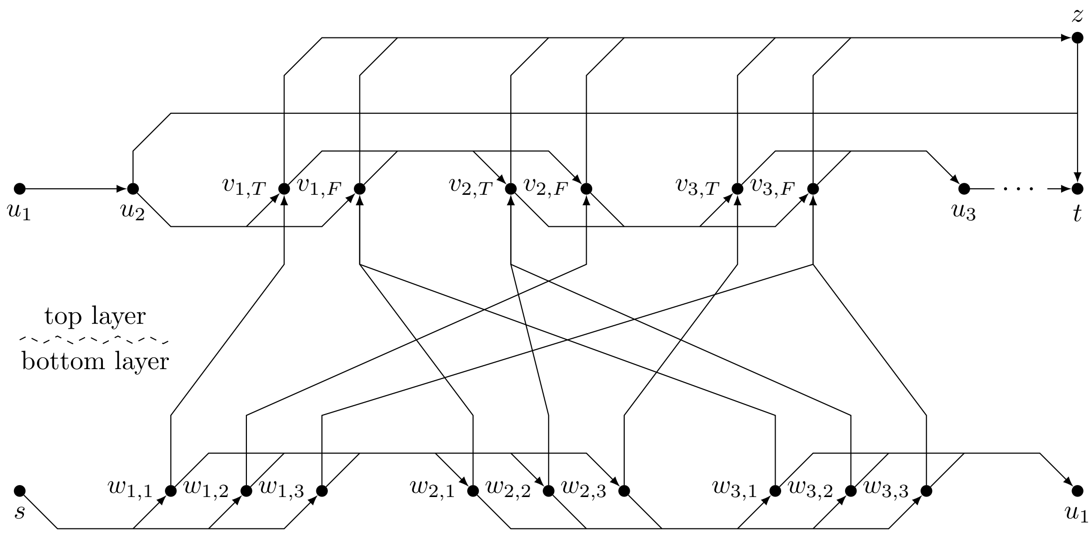

Figure 4.4: Reduction from the 3-SAT instance: $(\neg x_1, x_2, x_3), (x_1, \neg x_2, \neg x_3), (x_1, \neg x_2, x_3)$

This proves the theorem for $r = 1/\beta$. As usual, the more general result for $r > 0$ follows from Proposition 2.4.1.

We begin the proof by showing that our reduction satisfies (a), i.e., $\mathcal{J}$ admits a motivating cost configuration whenever $\mathcal{I}$ has a satisfying truth-value assignment. For this purpose, let τ be a satisfying truth-value assignment and consider the cost configuration h that assigns an extra cost of $(1-\beta)^2$ to all shortcut edges of the form $(v_{k,\tau(x_k)}, z)$. Furthermore, let h charge an extra cost of $2/\beta$ for all regular edges incident to an unsatisfied boolean or literal node. It is easy to see that the resulting task graph G_h is conceptually identical to the subgraph G_F constructed in Theorem 3.2.3. First, the extra cost assigned to the edges $(v_{k,\tau(x_k)}, z)$ corresponds to the removal of the single and double edge shortcuts in G_F. Secondly, the extra cost assigned to the regular edges is large enough to discourage the agent from planning to cross the respective edges at any point in time. Clearly, this is equivalent to removing these edges all together. From the proof of Theorem 3.2.3 we know that G_F is motivating and therefore G_h must be motivating as well.

We proceed to show that our reduction also satisfies (b), i.e., $\mathcal{I}$ has a satisfying truth-value assignment whenever $\mathcal{J}$ admits a motivating cost configuration. To establish this property, let h be a motivating cost configuration and P the path that the agent takes through G_h. A similar argument to that presented in Theorem 3.2.3 proves that P cannot contain any shortcuts, but must pass one literal node $w_{i,j}$ and one boolean node $v_{k,y}$ for each clause c_i and variable x_k. Consequently, we can construct a truth-value assignment

τ in the following way: If P includes the node $v_{k,T}$, then $\tau(x_k) = T$. Otherwise, if P includes $v_{k,F}$, then $\tau(x_k) = F$.

To conclude the proof, we need to show that τ indeed satisfies all clauses. For this purpose, consider an arbitrary clause c_i and let $w_{i,j}$ be the corresponding literal node contained in P. Furthermore, let w' be the node that precedes $w_{i,j}$ with respect to P and P' the path that the agent plans to take at w'. According to the same argument given in the proof of Theorem 3.2.3, P' cannot contain any regular edge apart from $(w', w_{i,j})$. As a result, P' must take the direct shortcut from $w_{i,j}$ to the corresponding boolean node $v_{k,y}$ followed by the shortcut from $v_{k,y}$ to t. The perceived cost of this path is at least $\tilde{c}_h(P', \beta) \geq 1 - \varepsilon$, even if we ignore extra cost. Therefore, h can assign not more than β/ε of extra cost to the shortcut at $v_{k,y}$, otherwise P' would not be motivating.

For the sake of contradiction, assume that P includes the boolean node $v_{k,y}$. As argued in the proof of Theorem 3.2.3, all regular edges at $v_{k,y}$ have a perceived cost of at least $(1-\beta)^3 - \varepsilon + \beta \cdot (2 - \beta)$. Conversely, the perceived cost of the shortcut at $v_{k,y}$ is at most $\varepsilon/\beta + \beta \cdot (2 - \beta)$ due to the bound on the extra cost established in the previous paragraph. By choice of ε, this not enough extra cost to keep the agent from entering the shortcut, considering that

$$\big((1-\beta)^3 - \varepsilon\big) - \frac{\varepsilon}{\beta} = \frac{\beta+1}{\beta} \cdot \Big(\frac{\beta \cdot (1-\beta)^3}{1+\beta} - \varepsilon\Big) > 0.$$

But this violates the fact that the agent must not take shortcuts. Consequently, P must contain the boolean node $v_{k,\neg y}$ instead. By construction of G, this implies that τ satisfies the j-th literal of c_i. As c_i was chosen arbitrarily, τ must be satisfying. □

4.4 Greedy Threats

The NP-completeness of MCC indicates that the design of almost optimal penalty based incentives is as challenging as the design of optimal prohibition based incentives, at least in a general setting. Uncovering the computational benefits of penalty fees therefore requires a more nuanced analysis. As a first step we revisit the notion of optimal travel routes. Recall that the previous chapter defines such a route as the collection W of nodes that the agent may visit in an optimal subgraph. Extending this idea to penalty based incentives, let W be an *almost optimal travel route* if W contains all nodes that the agent may visit in an almost optimal cost configuration.

In the following assume that we have knowledge of an almost optimal travel route W. Our goal is to use this information to reconstruct an almost optimal cost configuration h.

At first sight, this may seem ambitious considering that the same problem is NP-hard for prohibition based incentives, see Theorem 3.3.1. However, the following greedy algorithm constructs the desired cost configuration h in polynomial time assuming that we provide a sufficiently small tie breaker $\varepsilon > 0$.

Algorithm 3: GREEDYTHREATS

Input: task graph G, present-bias β, node set W, tie breaker ε
Output: cost configuration h

foreach $e \in E$ **do** $h(e) \leftarrow 0$;
foreach $v \in W \setminus \{t\}$ **in reverse topological order do**
  $\delta \leftarrow \min\{\tilde{c}_h(v, w, \beta) \mid w \in W\}$;
  foreach $u \notin W$ **do** $h(v, u) \leftarrow \max\{0, \delta - \tilde{c}_h(v, u, \beta) + \varepsilon\}$;
return h;

The basic idea of GREEDYTHREATS is simple. After initializing the cost configuration h to assign zero extra cost to the edges in G, the algorithm iterates through all nodes of $W \setminus \{t\}$ in reverse topological order, i.e., node v is considered before node v' if v is reachable from v' via a directed path. Let v be the node of the current iteration. Furthermore, let w be a successor of v that is also contained in W. Assuming that W is a valid travel route, at least one such node w must exist. The value δ denotes the minimum perceived edge cost over all edges (v, w) with respect to the current cost configuration h. To make sure that the agent stays within the set W, the algorithm greedily assigns an extra cost of $\max\{0, \delta - \tilde{c}_h(v, u, \beta) + \varepsilon\}$ to all edges (v, u) that exit W at v. This way, the perceived cost of (v, u) is greater than δ by a difference of at least ε. Furthermore, the extra cost assigned to (v, u) does not change the perceived cost of any previously considered edge. The reason is that GREEDYTHREATS iterates through $W \setminus \{t\}$ in reverse topological order. An illustrative way to think of h is as a greedily assembled collection of penalties for diverting from the travel route W. However, due to the effectiveness of this construction, the agent never never leaves W and the penalties become mere threats. For this reason, we call the algorithm GREEDYTHREATS.

It remains to show that GREEDYTHREATS yields an almost optimal cost configuration h. In fact we prove a slightly more general result, namely that for any collection W of nodes that the agent may visit in a motivating cost configuration h^*, GREEDYTHREATS constructs a cost configuration h that is motivating for the same reward as h^*. An interesting conceptual interpretation of this result is that a present-biased person's behavior can be just as efficiently influenced by the mere threat of a future penalty as by the actual enforcement of a penalty.

Proposition 4.4.1. *Let h^* be a cost configuration for the task graph $G = (V, E, c, r)$ and W the set of nodes that an agent with present-bias $\beta \in (0, 1]$ may visit in G_{h^*} for a reward of $\mu(G_{h^*}, \beta)$. If the tie breaker is set to*

$$\varepsilon \leq \min\big\{\tilde{c}_{h^*}(v, u, \beta) - \tilde{c}_{h^*}(v, \beta) \;\big|\; v \in W, u \in V \setminus W, (v, u) \in E\big\},$$

then GreedyThreats *constructs a cost configuration h that is motivating for a reward of $\mu(G_{h^*}, \beta)$.*

Proof. From the description of GreedyThreats it should be clear that the cost configuration h provides no incentive to leave the travel route W. This holds true independently of the actual tie break value. Without loss of generality we therefore assume that $\varepsilon \leq \min\{\tilde{c}_{h^*}(v, u, \beta) - \tilde{c}_{h^*}(v, \beta) \mid v \in W, u \in V \setminus W, (v, u) \in E\}$. Note that this is a positive bound. The reason is that $\tilde{c}_{h^*}(v, u, \beta) > \tilde{c}_{h^*}(v, \beta)$ holds true for all edges (v, u) that exit the set W. Otherwise, the agent has an incentive to cross (v, u). But this contradicts the assumption that (v, u) exits the set W.

Because the agent does not leave W with respect to h, it suffices to show that h is motivating for a reward of $\mu(G_{h^*}, \beta)$. To prove this claim, consider an arbitrary node $v \in W \setminus \{t\}$ and assume that all nodes $w \in W \setminus \{t\}$ that precede v in a topological ordering of the task graph satisfy the following two properties:

(a) The perceived cost $\tilde{c}_h(w, \beta)$ is less or equal to the perceived cost $\tilde{c}_{h^*}(w, \beta)$.

(b) The cheapest path cost $d_h(w)$ is less or equal to the cheapest path cost $d_{h^*}(w)$.

Our goal is to show that h also satisfies (a) and (b) for v. Keeping in mind that h^* is by definition motivating for a reward of $\mu(G_{h^*}, \beta)$, property (a) immediately implies the desired result.

We begin with (a), i.e., we need to show that $\tilde{c}_h(v, \beta) \leq \tilde{c}_{h^*}(v, \beta)$. By definition of W it is possible to express $\tilde{c}_{h^*}(v, \beta)$ as $c(v, w) + h^*(v, w) + \beta \cdot d_h(w)$ for some successor $w \in W$ of v. In the special case of $w = t$, the term $d_{h^*}(t)$ evaluates 0. Together with the observation that GreedyThreats assigns no extra cost to edges within W, i.e., $h(v, t) = 0$, we obtain the desired result

$$\tilde{c}_{h^*}(v, \beta) = c(v, t) + h^*(v, t) \geq c(v, t) + h(v, t) \geq \tilde{c}_h(v, \beta).$$

Next, assume that $w \neq t$. In this case hypothesis (b) implies $d_h(w) \leq d_{h^*}(w)$. Because GreedyThreats sets $h(v, w) = 0$, it once more follows that

$$\tilde{c}_{h^*}(v, \beta) = c(v, w) + h^*(v, w) + \beta \cdot d_{h^*}(w) \geq c(v, w) + h(v, w) + \beta \cdot d_h(w) \geq \tilde{c}_h(v, \beta).$$

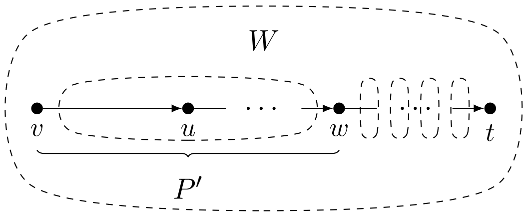

Figure 4.5: Structure of P assuming that $u \neq w \neq t$

We continue with (b), i.e., we need to show $d_h(v) \leq d_{h^*}(v)$. For this purpose, let $P = v, u, \ldots, t$ be a cheapest path to t with respect to h^*, i.e., $\sum_{e \in P} c(e) + h^*(e) = d_{h^*}(v)$. Furthermore, let $P' = v, u, \ldots, w$ be the subpath of P that goes from v to the first node $w \neq v$ that is contained in W. Note that such a node w must exist because P ends at t, which is contained in W. On the other hand, it might hold true that $w = u$ or $w = t$. Figure 4.5 illustrates the structure of P for distinct nodes u, w and t. It is crucial to observe that h cannot not charge any extra cost within the path segment P' except at (v, u). The reason is that GreedyThreats only assigns an extra cost to edges going from a node within W to a node outside of W. However, by choice of w no such node exists in P'. Next, we make a case distinction on whether $h(v, u)$ is a positive value.

First, assume that $h(v, u) = 0$. In this case, h does not charge any extra cost for the path segment P'. Consequently, we can bound the cheapest path cost $d_{h^*}(v)$ by

$$d_{h^*}(v) = \sum_{e \in P} c(e) + h^*(e) = \Big(\sum_{e \in P'} c(e) + h^*(e)\Big) + d_{h^*}(w) \geq \Big(\sum_{e \in P'} c(e) + h(e)\Big) + d_{h^*}(w).$$

Moreover, an argument similar to that given in the proof of the inductive step (a) implies that $d_h(w) \leq d_{h^*}(w)$, which yields the desired result

$$\Big(\sum_{e \in P'} c(e) + h(e)\Big) + d_{h^*}(w) \geq \Big(\sum_{e \in P'} c(e) + h(e)\Big) + d_h(w) \geq d_h(v).$$

Secondly, assume that $h(v, u) > 0$. In this case, GreedyThreats assigns an extra cost to (u, v) in such a way that $\tilde{c}_h(v, u, \beta) = \tilde{c}_h(v, \beta) + \varepsilon$. Consequently, it holds true that

$$\begin{aligned} d_h(v) &\leq c(v, u) + h(v, u) + d_h(u) = \tilde{c}_h(v, u, \beta) + (1 - \beta) \cdot d_h(u) \\ &= \tilde{c}_h(v, \beta) + \varepsilon + (1 - \beta) \cdot d_h(u). \end{aligned}$$

However, according to the result of (a) it holds true that $\tilde{c}_h(v, \beta) \leq \tilde{c}_{h^*}(v, \beta)$. Together with the choice of ε, we get

$$\begin{aligned}\tilde{c}_h(v,\beta)+\varepsilon+(1-\beta)\cdot d_h(u) &\leq \tilde{c}_{h^*}(v,\beta)+\big(\tilde{c}_{h^*}(v,u,\beta)-\tilde{c}_{h^*}(v,\beta)\big)+(1-\beta)\cdot d_h(u)\\ &= \tilde{c}_{h^*}(v,u,\beta)+(1-\beta)\cdot d_h(u).\end{aligned}$$

At his point it is essential to recall that GreedyThreats assigns no extra cost to the edges of P' except for the first one, i.e., $h(e) \leq h^*(e)$ for all $e \in P'\backslash\{(v,u)\}$. Furthermore, it holds true that $d_h(w) \leq d_{h^*}(w)$. Consequently, we can bound $d_h(u)$ by

$$d_h(u) \leq d_h(w) + \sum_{e\in P'\backslash\{(v,u)\}} c(e)+h(e) \leq d_{h^*}(w) + \sum_{e\in P'\backslash\{(v,u)\}} c(e)+h^*(e) = d_{h^*}(u),$$

which in turn implies that

$$\tilde{c}_{h^*}(v,u,\beta)+(1-\beta)\cdot d_h(u) \leq \tilde{c}_{h^*}(v,u,\beta)+(1-\beta)\cdot d_{h^*}(u) = d_{h^*}(v).$$

The last equality holds true because the edge (v,u) is by assumption part of a cheapest path from v to t with respect to h^*. All in all, this proves that $d_h(v) \leq d_{h^*}(v)$. □

4.5 The Copied Cost Approximation

The previous section suggests that penalty fees have a certain computational advantage over prohibitions. This observation becomes particularly apparent in the light of Theorem 3.5.2. Recall that this theorem establishes a hardness of approximation result for the design of optimal prohibitions. However, the proof of the theorem relies heavily on the fact that the knowledge of an optimal travel route does not simplify the computational problem at hand. Considering that this key property of prohibition based incentives does not apply to penalty based incentives, we may hope that latter type of incentive admits more practical approximation algorithms than the former type.

Consider for instance CopiedCostApprox, an algorithm that takes a task graph G and an agent with present-bias $\beta \in (0,1]$ as its input and returns a cost configuration h that approximates any almost optimal cost configuration within a factor of 2. The main idea of CopiedCostApprox, which we also present in [3], is simple: First, the algorithm identifies a path P from s to t whose maximum perceived edge cost is a lower bound on $\beta \cdot \mu(G_{h'},\beta)$ for any conceivable cost configuration h'. Secondly, it assigns an extra cost to all outgoing edges e of P such that the perceived cost of e either discourages the agent from crossing e or the perceived cost of e exceeds any future cost that the agent may experience before reentering P. Thus, the agent has no incentive to quit outside of P. Furthermore, the extra cost is assigned in such a way that the perceived cost of the edges of P is at most doubled. This yields the desired 2-approximation.

The reason why we call the algorithm CopiedCostApprox is that the extra cost $h(e)$ it assigns to an outgoing edge e of P is identical to the cost $c(e')$ of some other edge e', i.e., $h(e) = c(e')$ for some edge e' of G. This peculiar detail about the structure of h is particularly useful for the implementation of h in a real-world setting. After all, it suffices to impose the tasks associated with e' as additional tasks for e. Note that this strategy even works in settings without money.

Algorithm 4: CopiedCostApprox

Input: Task graph G, present-bias β
Output: Cost configuration h

$P \leftarrow$ minmax path from s to t with respect to $\tilde{c}$; $\alpha \leftarrow \max\{\tilde{c}(e, \beta) \mid e \in P\}$;
foreach $v \in V \setminus \{t\}$ **do**
  $\sigma(v) \leftarrow$ successor node of v on a cheapest path from v to t;
$T = \{(v, \sigma(v)) \mid v \in V \setminus \{t\}\}$;
foreach $(v, w) \in E$ **do**
  if $(v, w) \notin P \cup T$ **then**
    $h(e) \leftarrow 2 \cdot \alpha/\beta + 1$;
  else if $v \in P$ **and** $w \notin P$ **then**
    $P' \leftarrow v, \sigma(v), \sigma(\sigma(v)), \ldots, t$;
    $u \leftarrow$ first node of P' different from v that is also a node of P;
    $h(v, w) \leftarrow$ cost of a most expensive edge of P' between v and u;
  else $h(e) \leftarrow 0$;
return h;

For a more detailed discussion of CopiedCostApprox assume that each edge e is labeled with its perceived cost $\tilde{c}(e, \beta)$. Furthermore, let h' be an arbitrary cost configuration and P' the path that the agent takes from s to t in $G_{h'}$ for a reward of $\mu(G_{h'}, \beta)$. Our goal is to bound $\beta \cdot \mu(G_{h'}, \beta)$ from below by some value α. For this purpose, we set

$$\alpha = \min\bigl\{\max\{\tilde{c}(e, \beta) \mid e \in P\} \bigm| P = s, \ldots, t\bigr\} \leq \max\{\tilde{c}(e, \beta) \mid e \in P'\}$$
$$\leq \max\{\tilde{c}_{h'}(e, \beta) \mid e \in P'\} \leq \beta \cdot \mu(G_{h'}, \beta).$$

In other words, α is the maximum perceived edge cost of a minmax path P from s to t with respect to $\tilde{c}$. Note that P can be computed in polynomial-time by adding the edges of G in non-decreasing order of perceived cost to an initially empty graph until s and t become connected for the first time. Any path from s to t through the resulting subgraph of G is a suitable minmax path.

We continue with the construction of h. To facilitate this task, CopiedCostApprox sets up a *cheapest path successor relation* σ. More precisely, it assigns a distinct successor node $\sigma(v)$ to each node $v \in V \setminus \{t\}$. The successor is chosen in such a way that $(v, \sigma(v))$ is the initial edge of a cheapest path from v to t. Since we may assume that t is reachable from each node of G, all $v \neq t$ must have at least one suitable successor. By construction of σ, any path $P' = v, \sigma(v), \sigma(\sigma(v)), \ldots, t$ is a cheapest path from v to t. We call P' the *σ-path* of v and $T = \{(v, \sigma(v)) \mid v \in V \setminus \{t\}\}$ a *cheapest path tree*.

Once σ has been created, CopiedCostApprox starts to assign extra cost to the edges (v, w) of G. For this purpose, the algorithm distinguishes between three types of edges: First, (v, w) may neither be part of the minmax path P or a σ-path. In this case, (v, w) is irrelevant to the approximation scheme. To ensure that the agent does not plan along (v, w), CopiedCostApprox imposes an extra cost of $h(v, w) = 2 \cdot \alpha/\beta + 1$. As a result, any path along (v, w) has a perceived cost of at least $2 \cdot \alpha + 1/\beta$. Clearly, this cost is not covered by the anticipated reward of $(2 \cdot \alpha)/\beta$. Alternatively, the edge (v, w) may simply be removed if prohibition is possible. Secondly, (v, w) might be an edge that exits the minmax path P, i.e., $v \in P$ but $w \notin P$. To determine an appropriate extra cost for (v, w), CopiedCostApprox considers the σ-path P' of v. Let u be the first common node between P and P' that is different from v. As P and P' both end in t, such a node must exist. Moreover, let e be the most expensive edge of P' between v and u. The algorithm assigns an extra cost of $h(v, w) = c(e)$. As we show in Theorem 4.5.1, this cost is sufficient to ensure that the agent does not encounter a perceived cost greater than $\tilde{c}_h(v, w, \beta)$ should she decide to take a shortcut to u along the σ-path of w. Finally, if neither of the above two cases applies to (v, w), then (v, w) must either be an edge of P or an internal edge of a σ path, i.e., a σ-edge that does not exit P. In this case CopiedCostApprox adds no extra cost to (v, w).

Clearly, CopiedCostApprox can be implemented to run in polynomial-time with respect to the encoding length of G and β. It remains to show that the algorithm returns a cost configuration h that achieves an approximation ratio of 2.

Theorem 4.5.1. *Given a task graph G and present-bias $\beta \in (0, 1]$,* CopiedCostApprox *constructs a cost configuration h whose minimal motivating reward $\mu(G_h, \beta)$ is less than $2 \cdot \mu(G_{h'}, \beta)$ for any other cost configuration h'.*

Proof. Recall that α denotes the maximum perceived edge cost along the minmax path P. Furthermore, it should be evident that $\alpha/\beta \leq \mu(G_{h'}, \beta)$ for any cost configuration h'. To prove the theorem, we need to show that CopiedCostApprox returns a cost configuration h that is motivating for a reward of $2 \cdot \alpha/\beta$.

As a first step we argue that a cheapest path from any node v to t in G_h costs at most twice as much as in G. More formally, we show that $d_h(v) \leq 2 \cdot d(v)$. For this purpose let P' be the σ-path of v. By construction of σ, P' is a cheapest path from v to t. It is crucial to observe that CopiedCostApprox assigns an extra cost to an edge $(v', \sigma(v'))$ of P' only if v' is located on P. Consequently, there is at most one edge with extra cost between any two consecutive intersections of P and P'. Moreover, this extra cost matches the cost of an edge of P' between v' and the next intersection of P and P'. Therefore, each edge of P' can contribute at most once to the total extra cost assigned to P'. This means that h can at most double the price of P'. Because the price of P' is an upper bound on $d_h(v)$, it follows that $d_h(v) \leq 2 \cdot d(v)$.

We proceed to investigate the agent's walk through G_h. Our goal is to show that the perceived cost is at most $2 \cdot \alpha$ at each node v of her walk. This establishes the theorem. Our analysis is based on the following case distinction: First, assume that v is located on P and let w be the immediate successor of v on P. Since h assigns no extra cost to (v, w), we conclude that v has a perceived cost of

$$\begin{aligned}\tilde{c}_h(v, \beta) &\leq \tilde{c}_h(v, w, \beta) = c(v, w) + \beta \cdot d_h(w) \leq c(v, w) + \beta \cdot 2 \cdot d(w) \\ &\leq 2 \cdot \big(c(v, w) + \beta \cdot d(w)\big) = 2 \cdot \tilde{c}(v, w, \beta) \leq 2 \cdot \alpha.\end{aligned}$$

The last inequality is valid by definition of α.

Secondly, assume that v is not located on P. In this case, let v' be the last node of P that the agent visited before v. Since the agent is only motivated to travel along edges of P and T, we know that v is located on the σ-path of v'. In particular, this implies that the agent must have crossed the edge $(v', \sigma(v'))$ to get to v. Consequently, it holds true that $\tilde{c}_h(v', \sigma(v'), \beta) \leq 2 \cdot \alpha$ and $d_h(\sigma(v')) \leq 2 \cdot \alpha/\beta$. Taking into account that all paths from $\sigma(v')$ to t either visit v or contain an edge that charges extra cost of $2 \cdot \alpha/\beta + 1$, we furthermore know that

$$d_h(\sigma(v')) \geq \min\Big\{\frac{2 \cdot \alpha}{\beta} + 1, d_h(\sigma(v))\Big\}.$$

But $d_h(\sigma(v'))$ cannot be greater than $2 \cdot \alpha/\beta$, otherwise a reward of $2 \cdot \alpha/\beta$ would not be sufficiently motivating for the agent to cross the edge $(v', \sigma(v'))$. Consequently, it holds true that $d_h(\sigma(v')) \geq d_h(\sigma(v))$. To complete the proof, recall that $(v, \sigma(v))$ is located on P' between v' and the next intersection of P and P'. By construction of h we know that $h(v', \sigma(v')) \geq c(v, \sigma(v))$. Furthermore, h assigns no extra cost to $(v, \sigma(v))$. All of this together finally yields the desired inequality

$$\begin{aligned}\tilde{c}_h(v, \beta) &\leq \tilde{c}_h(v, \sigma(v), \beta) = c(v, \sigma(v)) + \beta \cdot d_h(\sigma(v)) \leq h(v', \sigma(v')) + \beta \cdot d_h(\sigma(v')) \\ &\leq c(v', \sigma(v')) + h(v', \sigma(v')) + \beta \cdot \tilde{c}_h(\sigma(v'), \beta) = \tilde{c}_h(v', \sigma(v'), \beta) \leq 2 \cdot \alpha. \qquad \square\end{aligned}$$

4.6 The Approximability of Cost Configurations

We conclude this chapter by taking a closer look at the theoretical limitations that underlie the approximability of almost optimal penalty fees. For this purpose, we restate the decision problem MCC as the following optimization problem MCC-OPT.

Definition 4.6.1 (MCC-OPT)**.** The problem of computing $\mu^*(G, \beta)$ for a given task graph G and present-bias $\beta \in (0, 1]$.

By reusing the reduction from the proof of Theorem 4.3.1, we are able to show that MCC-OPT is NP-hard to approximate within a ratio of 1.08192. This result, which is taken from [3], is particularly interesting as it implies that the design of almost optimal penalty fees does not admit a PTAS unless P = NP.

Theorem 4.6.1. *MCC-OPT is NP-hard to approximate within a ratio less or equal to* 1.08192.

Proof. Given a 3-SAT instance $\mathcal{I}$ of ℓ clauses $c_1, \ldots, c_\ell$ over m variables $x_1, \ldots, x_m$, we consider the same reduction as in the proof of Theorem 4.3.2. The only difference is that our choice of ε is slightly more restrictive this time, as we assume that

$$\varepsilon < \min\Big\{\beta \cdot (1-\beta)^3, \frac{\beta^2 \cdot (1-\beta)^3}{1+\beta}\Big\}.$$

Let $G = (V, E, c, r)$ be the task graph of the resulting MCC instance $\mathcal{J}$. To prove the theorem we need to verify that the following two properties of $\mathcal{J}$ hold true:

(a) If $\mathcal{I}$ has a satisfiable truth-value assignment, then $\mathcal{J}$ admits a motivating cost configuration for a reward of $r = 1/\beta$.

(b) If $\mathcal{I}$ has no satisfiable truth-value assignment, then $\mathcal{J}$ does not admit a motivating cost configuration for a reward of $r \leq (1 + \beta \cdot (1-\beta)^4)/\beta$.

Consequently, any algorithm that approximates the minimal reward $\mu(G, \beta)$ within a ratio of $1 + \beta \cdot (1 - \beta)^4$ or less must also solve $\mathcal{I}$. Choosing $\beta = 1/5$ maximizes this ratio and we obtain the desired approximation bound, namely $1 + (1 - 1/5)^4/5 = 1.08192$.

It remains to show that $\mathcal{J}$ satisfies (a) and (b). However, the correctness of (a) is an immediate consequence of the proof of Theorem 4.3.2. We therefore focus on (b). For the sake of contradiction assume that there exists a cost configuration h that is motivating for a reward of at most $(1 + \beta(1 - \beta)^4)/\beta$, but yet $\mathcal{I}$ has no solution. Let P be a path that corresponds to the agent's walk from s to t.

Similar to the proof of Theorem 3.2.3, we first observe that P cannot contain any shortcut. Recall that all shortcuts at the top layer have an edge e of cost $2-\beta$. However, the perceived reward is at most $1+\beta(1-\beta)^4$. Because $2-\beta = 1+(1-\beta) > 1+\beta(1-\beta)^4$, the agent has no incentive to traverse e. To verify the observation for the bottom layer, assume that P contains a shortcut edge from some literal node $w_{i,j}$ to a boolean node $v_{k,y}$. Let P' be the agent's planned path when located at $w_{i,j}$. We distinguish between two scenarios. First, P' might include a regular edge after $(w_{i,j}, v_{k,y})$. In this case, the perceived cost of P' is at least

$$\begin{aligned}\tilde{c}_h(P', \beta) &\geq (1 - \beta)^2 + \beta \cdot \big((1 - \beta)^3 - \varepsilon + (2 - \beta)\big) \\ &> (1 - \beta)^2 + \beta \cdot \big((1 - \beta)^3 - \beta \cdot (1 - \beta)^3 + (2 - \beta)\big) = 1 + \beta \cdot (1 - \beta)^4,\end{aligned}$$

even if we neglect extra cost. The second inequality holds true by choice of ε. Because the perceived cost of P' exceeds the perceived reward, this scenario is not possible. Secondly, P' might contain the shortcut from $v_{k,y}$ to t. In this case, the perceived cost of P' is at least 1. Consequently, h may assign an extra cost of no more than $(\beta \cdot (1 - \beta)^4)/\beta = (1 - \beta)^4$ to the edges of P'. This holds particularly true for the edges of the shortcut from $v_{k,y}$ to t. Therefore, the perceived cost of taking the shortcut at $v_{k,y}$ is at most $(1 - \beta)^4 + \beta \cdot (2 - \beta)$. Conversely, any path P'' from $v_{k,y}$ along a regular initial edge must include a top layer shortcut or pass the nodes u_3, u_4 and u_5 to get to t. In both cases, the perceived cost of P'' is at least

$$\begin{aligned}\tilde{c}_h(P'', \beta) &\geq (1 - \beta)^3 - \varepsilon + \beta \cdot (2 - \beta) \\ &> (1 - \beta)^3 - \beta \cdot (1 - \beta^3) + \beta \cdot (2 - \beta) = (1 - \beta)^4 + \beta \cdot (2 - \beta).\end{aligned}$$

As a result, the agent prefers the shortcut at $v_{k,y}$ over all regular edges. This is a contradiction to the previous observation that P does not contain a top layer shortcut.

Because h does not guide the agent onto a shortcut, we conclude that P contains one literal node $w_{i,j}$ and one boolean node $v_{k,y}$ for each clause and variable of $\mathcal{I}$. Similar to the proof of Theorem 3.2.3, we use P to construct a variable assignment τ in the following way: If the agent visits $v_{k,T}$ along P, we set $\tau(x_k) = T$. Otherwise, if she visits $v_{k,F}$, we set $\tau(x_k) = F$. To conclude the proof we argue that τ satisfies all clauses of $\mathcal{I}$. This is a contradiction to our initial assumption that $\mathcal{I}$ has no solution.

Consider an arbitrary clause c_i and let $w_{i,j}$ be the corresponding literal node in P. Furthermore, let w' be the node that precedes $w_{i,j}$ with respect to P and P' the path that the agent plans to take at w'. As argued in the proof of Theorem 3.2.3, P' may only contain one regular edge. Otherwise, the perceived cost of P' is at least

$$\begin{aligned}\tilde{c}_F(P', \beta) &\geq (1-\beta)^3 - \varepsilon + \beta \cdot \big((1-\beta)^3 - \varepsilon + (1-\beta)^2 + (2-\beta)\big) \\ &= 1 + \beta \cdot (1-\beta)^3 - (1+\beta) \cdot \varepsilon > 1 + \beta \cdot (1-\beta)^3 - (1+\beta) \cdot \frac{\beta^2 \cdot (1-\beta)^3}{1+\beta} \\ &= 1 + \beta \cdot (1-\beta)^4.\end{aligned}$$

By choice of ε, this is not motivating. But if P' only contains one regular edge, it must consist of the direct shortcut from $w_{i,j}$ to the corresponding boolean node $v_{k,y}$ and the shortcut from $v_{k,y}$ to t. In this case, the perceived cost of P' is at least $1 - \varepsilon$. This leaves an extra cost of no more than $(\varepsilon + \beta \cdot (1-\beta)^4)/\beta = \varepsilon/\beta + (1-\beta)^4$ to place on the shortcut from $v_{k,y}$ to t.

Finally, assume that P also includes $v_{k,y}$. The perceived cost for taking a regular edge at $v_{k,y}$ is at least $(1-\beta)^3 - \varepsilon + \beta \cdot (2-\beta)$. Conversely, the perceived cost of the shortcut at $v_{k,y}$ is at most $\varepsilon/\beta + (1-\beta)^4 + \beta \cdot (2-\beta)$ due to the bound on the extra cost established in the previous paragraph. By choice of ε this not enough extra cost to keep the agent from entering the shortcut considering that

$$\Big((1-\beta)^3 - \varepsilon\Big) - \Big(\frac{\varepsilon}{\beta} + (1-\beta)^4\Big) = \frac{\beta+1}{\beta} \cdot \Big(\frac{\beta^2 \cdot (1-\beta)^3}{1+\beta} - \varepsilon\Big) > 0.$$

Of course, this contradicts the fact that the agent cannot take shortcuts. As a result, the agent cannot visit $v_{k,y}$, but must visit $v_{k,\neg y}$ instead. By construction of G this implies that τ satisfies the j-th literal of clause c_i and τ must be satisfying. $\square$

5 Reward Based Incentives

The previous two chapters deal with incentives that influence a person's behavior by making undesirable courses of action less appealing, either by prohibition or by imposing penalty fees. The incentives presented in this chapter are based on the opposite idea, namely to make desirable courses of action more appealing. For this purpose, we offer rewards at intermediate states of progress. The resulting incentives are surprisingly budget efficient in some scenarios. However, their design is computationally difficult. To resolve this problem, we combine intermediate rewards with penalty fees and prohibition. This yields a very powerful, yet computationally tractable incentive design tool.

5.1 Extending the Graphical Model

Incorporating the idea of reward based incentives in the graphical model requires the placement of multiple rewards at various nodes of a given task graph $G = (V, E, c, r)$.to For this purpose, let $r : V \to \mathbb{R}_{\geq 0}$ be an assignment of non-negative rewards to the nodes of G rather than the value of a single reward placed at the target node t. We call such an assignment a *reward configuration*. To emphasize the role of reward configurations as incentives, we write G_r instead of G whenever we study a particular reward configuration r.

Definition 5.1.1 (Reward Configuration)**.** The reward configuration $r : V \to \mathbb{R}_{\geq 0}$ of a task graph $G_r = (V, E, c, r)$ assigns a non-negative reward $r(v)$ to each node $v \in V$.

Note that the extension to intermediate rewards may cause the agent to collect different amounts of reward depending on the way she takes through the task graph G_r. As a result, her preferences are not solely determined by perceived path cost anymore. Instead, she bases her decisions on the more intricate notion of *perceived net cost*, i.e., the perceived cost of a path minus the perceived reward of the same path. More formally, the perceived net cost $\tilde{c}_r$ of a path $P = v, w, \ldots, t$ is defined as

$$\tilde{c}_r(P, \beta) = c(v, w) + \beta \cdot \Big(\sum_{e \in P \setminus \{(v,w)\}} c(e) - \sum_{u \in P \setminus \{v\}} r(u) \Big).$$

Similar to the standard notion of perceived cost, this definition extends to the nodes and edges of G_r in the following way: Given a node v, the perceived net cost $\tilde{c}_r(v,\beta)$ is the minimum perceived net cost of all paths from v to t. Moreover, the perceived net cost $\tilde{c}_r(v,w,\beta)$ of an edge (v,w) is the minimum perceived net cost of all paths whose initial edge is (v,w).

Definition 5.1.2 (Perceived Node and Edge Net Cost)**.** The perceived net cost of a node $v \in V \setminus \{t\}$ or edge $(v,w) \in E$ for a reward configuration $r : V \to \mathbb{R}_{\geq 0}$ is defined as $\tilde{c}_r(v,\beta) = \min\{\tilde{c}_r(P,\beta) \mid P = v,\ldots,t\}$ and $\tilde{c}_r(v,w,\beta) = \min\{\tilde{c}_r(P,\beta) \mid P = v,w,\ldots,t\}$.

While constructing her path from s to t, the agent only crosses edges (v,w) whose perceived net cost minimizes the perceived net cost among all outgoing edges of v, i.e., $\tilde{c}_r(v,w,\beta) = \tilde{c}_r(v,\beta)$. Furthermore, the perceived net cost of (v,w) must not be positive. Otherwise, the agent loses motivation and quits. We call the reward configuration r motivating if the agent does not quit on any of her paths through G_r.

To decide whether a given reward configuration r is motivating, we need an efficient way to compute the perceived edge net cost. For this purpose, let

$$d_r(w) = \min\Big\{\sum_{e\in P} c(e) - \sum_{u\in P} r(u) \;\Big|\; P = w,\ldots,t\Big\}$$

be the lowest net cost of any path from w to t. It is be easy to see that $d_r(w)$ can be computed in polynomial time with respect to the encoding length of G_r. Consider for instance a modified version G' of G_r in which each edge (v,w) has a cost of $c(v,w) - r(v)$ instead of $c(v,w)$. By construction of G' any cheapest path from v to t has a cost of $d_r(v) + r(t)$. Consequently, it is possible to compute $d_r(v)$ via a standard cheapest path algorithm that can handle the potentially negative edge weight of G'. Based on the value of $d_r(v)$ the perceived net cost of an edge (v,w) can be deduced as follows:

$$\begin{aligned}
\tilde{c}_r(v,w,\beta) &= \min\{\tilde{c}_r(P,\beta) \mid P = v,w,\ldots,t\} \\
&= \min\Big\{c(v,w) + \beta\cdot\Big(\sum_{e\in P\setminus\{(v,w)\}} c(e) - \sum_{u\in P\setminus\{v\}} r(u)\Big) \;\Big|\; P = v,w,\ldots,t\Big\} \\
&= c(v,w) + \beta\cdot\min\Big\{\sum_{e\in P'} c(e) - \sum_{u\in P'} r(u) \;\Big|\; P' = w,\ldots,t\Big\} \\
&= c(v,w) + \beta\cdot d_r(w).
\end{aligned}$$

Using the above method to compute the perceived net cost of edges enables us to simulate the agent's behavior and decide whether a given reward configuration is motivating.

Proposition 5.1.1. *Deciding whether a given task graph $G_r = (V, E, c, r)$ is motivating for an agent with present-bias $\beta \in (0, 1]$ is possible in polynomial time with respect to the encoding length of G_r and β.*

Proof. Consider a depth first search through G_r that starts at s and only traverses edges whose perceived net cost satisfies $\tilde{c}_r(v, w, \beta) = \tilde{c}_r(v, \beta)$. Let W be the set of nodes encountered during the search. Keeping in mind that the perceived net cost of the edges of G_r can be computed in polynomial time with respect to the encoding length of G_r and β, it is easy to see that the set W can be computed within the same time bounds as well. Furthermore, W contains exactly those nodes that the agent may visit if she does not abandon G_r prematurely. As a result, G_r is motivating if and only if no node of W has a positive perceived net cost. Clearly, this condition can be checked in polynomial time, which completes the proof. □

5.2 Exploitative Incentives

The construction of motivating reward configurations is not a particularly challenging design problem in itself. After all, any connected task graph becomes motivating if a sufficiently large reward is placed at its target node t. However, our goal is not just to construct motivating reward configurations, but to construct motivating reward configurations that are cost efficient at the same time. Similar to the previous incentive design settings, we measure the cost efficiency of a certain reward configuration based on the amount of reward that is collected by the agent. The only difference is that the current setting admits solutions that are motivating even if some of the allocated rewards are not paid out. Consider for instance the following scenario:

To prepare students for the final exam, Alice's professor offers two voluntary review courses at the end of his lecture series. The first course summarizes the basic content while the second course is dedicated to more advanced topics. Due to the different levels of difficulty, Alice expects to incur an effort of 1 and 3 for attending the respective courses. Unfortunately, like many other students Alice also believes to gain no value from attending the courses. Her professor, who is aware of this common misconception, therefore grants students a bonus on the final exam if they attend both review courses. However, he is already satisfied if they attend at least the first of the two courses. Figure 5.1 captures Alice's situation from the professor's perspective in the graphical model. The edges (s, v) and (v, w) correspond to the two review courses. After each course Alice completes the task graph free of charge via the edge (v, t) or (w, t). This reflects the professor's indifferent attitude towards attendance of the second course.

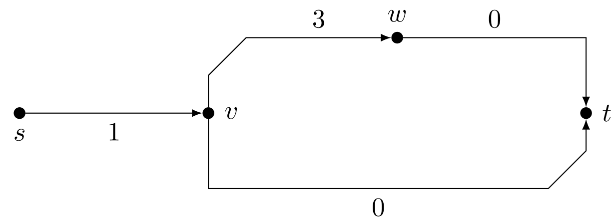

Figure 5.1: Task graph of Alice's training course

Assume that Alice estimates the value of the bonus to be 8, i.e., $r(w) = 8$. Furthermore, let her present-bias be $\beta = 1/3$. Under these assumptions, Alice is motivated to attend the first review course. The reason is that her perceived net cost for taking both review courses is $-2/3$ when located at s. More formally, it holds true that

$$\tilde{c}_r(s, \beta) = \min\{\tilde{c}_r(P, \beta) \mid P = s, \ldots, t\} = \min\{1 + \beta \cdot (3 - 8), 1 + \beta \cdot 0\} = -\frac{2}{3}.$$

However, once Alice reaches v, her perceived net cost for attending the second course becomes positive, i.e., $\tilde{c}_r(v, w, \beta) = 3 - \beta \cdot 8 = 1/3$. Clearly, this is not motivating. As a result, Alice abandons the reward at w and takes the edge (v, t) for a perceived net cost of $\tilde{c}_r(v, t, \beta) = 0$ instead. This completes her walk.

Note that the reward at w motivates Alice to move forward through the task graph, although she eventually fails to claim it. We call rewards with this property *exploitative*. Since exploitative rewards are not paid out, their contribution to the total cost of a reward configuration is not immediately clear. Tang et al. [23] suggest to add the value of exploitative rewards to the overall cost of the reward configuration. A conceivable application of this accounting scheme are scenarios in which rewards must be installed beforehand. However, many settings admit rewards that can be supplied on demand. Consider for instance the previous scenario. Clearly, the professor incurs no cost for the mere promise of a bonus on the final exam. Keeping in mind that Alice does not earn the bonus, it seems reasonable to omit the bonus from the overall cost of the reward configuration. To generalize this notion, let $q(G_r, \beta)$ be the maximum total reward that an agent collects on her way through a given task graph G_r. We call this quantity the *collected reward*. In Alice's case, the collected reward is 0.

Definition 5.2.1 (Collected Reward)**.** The collected reward $q(G_r, \beta)$ is the maximum total reward that an agent with present-bias β may collect on her walk through G_r.

It is instructive to think of $q(G_r, \beta)$ as the budget required for a particular reward configuration r whenever the rewards can be raised on demand. The latter assumption is particularly natural if r uses monetary rewards. Consequently, we believe that $q(G_r, \beta)$ provides a broadly applicable measure to quantify the cost efficiency of r that unlike Tang et al.'s approach also captures the phenomenon of exploitative rewards. However, to use this measure in practice, we need an efficient way to compute $q(G_r, \beta)$. The following theorem presents a possible approach:

Proposition 5.2.1. *The collected reward $q(G_r, \beta)$ can be computed in polynomial time with respect to the encoding length of G_r and β.*

Proof. Let G' be the subgraph of G containing those edges that an agent with present-bias β may cross in the task graph G_r on her way from s to t. Clearly, G' can be computed in polynomial time with respect to the encoding of G_r and β via a simple depth first search that starts at s and considers only edges (v, w) satisfying $\tilde{c}_r(v, w, \beta) = \tilde{c}_r(v, \beta) \leq 0$. To determine the value of $q(G_r, \beta)$, assume that each edge (v, w) of G' is labeled with a cost of $c'(v, w) = r(w)$. By construction of c', it holds true that $q(G_r, \beta) + r(s)$ is equal to the cost of a most expensive path in G' that starts at s. Since G' is acyclic, it is possible to compute such a path in polynomial time. This concludes the proof. □

5.3 Intermediate Reward versus Penalty Fees

The previous section provides tools to determine the cost efficiency of reward based incentives. But how do reward based incentives compare to other incentive design strategies such as penalty fees? To investigate this question, let G be an arbitrary task graph and $\beta \in (0, 1]$ the present-bias of the corresponding agent. Furthermore, recall that the minimum motivating reward $\mu^*(G, \beta)$ provides a tight lower bound on the reward needed to construct a motivating cost configuration. Keeping in mind that penalty based incentives are at least as cost efficient as prohibition based incentives, it seems reasonable to use $\mu^*(G, \beta)$ as a benchmark for our analysis. To compare reward based incentives to this benchmark, we adapt the notion of $\mu^*(G, \beta)$ to reward configurations as follows:

Definition 5.3.1 (Minimum Motivating Collected Reward)**.** The minimum motivating collected reward $\mu^*(G, \beta)$ is the infimum over the collected reward $q(G_r, \beta)$ of all reward configurations r that motivate an agent with present-bias $\beta \in (0, 1]$ to traverse G.

Note that the minimum motivating collected reward is defined in terms of an infimum rather than a minimum. The reason for this peculiar technicality is that $\mu^*(G, \beta)$ is drawn from an infinite set of reward configurations. Similar to cost configurations, this

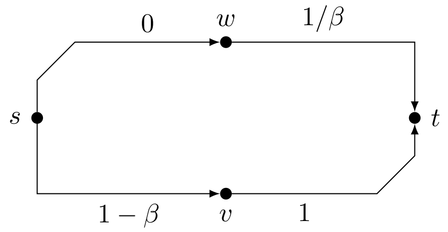

Figure 5.2: Task graph with no optimal reward configuration

implies that some choices of G and β may not admit an optimal motivating reward configuration. Consider for instance the task graph depicted in Figure 5.2

When located at the nodes v or w, a reward of $1/\beta$ or $1/\beta^2$ must be placed at t to keep the agent motivated. Since the agent eventually collects this reward, it is clearly more cost efficient to let her travel along v instead of w whenever $\beta < 1$. However, when located at s, the perceived cost of the paths s, v, t and s, w, t is identical, i.e., they both evaluate to 1. Independent of the reward at t, this implies that a positive reward of ε must be placed at v to break the tie in favor of the lower path. Consequently, G admits motivating reward configurations whose collected rewards are arbitrarily close to $1/\beta$, but no reward configuration whose collected reward is $1/\beta$ can be motivating due to the need of a tie breaker at v.

Having established the notion of a minimum motivating collected reward, we proceed to compare the cost efficiency of reward and penalty based incentives. For this purpose, we consider the ratio between the minimum motivating collected reward and the minimum motivating reward of the respective incentives. As in Theorem 4.2.1, which uses a similar measure to compare penalty and prohibition based incentives, we call this ratio the cost efficiency ratio. However, unlike the results of Theorem 4.2.1, we fail to identify a strict hierarchy between reward and penalty based incentives with respect to their cost efficiency. Instead, we witness scenarios in which reward based incentives outperform penalty based incentives and vice versa. According to the following proposition, this gap in performance may in fact become arbitrarily large in both directions. To facilitate the formal proof of the proposition, let $\mu^*_{\text{rwd}}(G, \beta)$ denote the minimal motivating collected reward and $\mu^*_{\text{pnl}}(G, \beta)$ the minimum motivating reward.

Proposition 5.3.1. *The cost efficiency ratio $\mu^*_{\text{rwd}}(G, \beta)/\mu^*_{\text{pnl}}(G, \beta)$ lies between 0 and β^{-n+2} for all task graphs G and present-bias values $\beta \in (0, 1]$. This result is tight.*

Proof. The lower bound of 0 is trivial because the cost efficiency ratio cannot become a negative number. To verify that this bound is also tight, it suffices to consider the task graph G depicted in Figure 5.1. Recall that this graph admits a motivating reward configuration whose collected reward is 0 for a present-bias of $\beta = 1/3$, i.e., $\mu^*_{\text{rwd}}(G, 1/3) = 0$. Furthermore, it is easy to see that any motivating cost configuration must place a reward of at least 3 onto t to motivate the agent to cross (s, v). This implies that $\mu^*_{\text{pnl}}(G, 1/3) = 3$, which in turn yields a cost efficiency ratio of 0.

We proceed with the upper bound of β^{-n+2}. The correctness of this bound follows from the observation that the minimum motivating collected reward $\mu^*_{\text{rwd}}(G, \beta)$ is always less or equal to the minimal motivating reward $\mu(G, \beta)$, i.e., the reward that needs to be spent if the only admissible incentive is a reward placed at t. According to Proposition 3.1.1, it holds true that $\mu(G, \beta)/\mu^*_{\text{pnl}}(G, \beta) \leq \beta^{-n+2}$. However, this immediately implies the claimed upper bound

$$\frac{\mu^*_{\text{rwd}}(G, \beta)}{\mu^*_{\text{pnl}}(G, \beta)} \leq \frac{\mu(G, \beta)}{\mu^*_{\text{pnl}}(G, \beta)} \leq \beta^{-n+2}.$$

To prove tightness of this bound, we revisit the task graph introduced in the proof of Proposition 3.1.1. Recall that this task graph G consists of a directed path $P = v_1, \ldots, v_{n-1}$ and a target node t. To save cost, it is essential that the agent does not procrastinate, but moves from P to t as soon as possible. Using penalty based incentives, this can be achieved easily by assigning a large enough extra cost to the initial edge of P, i.e., (v_1, v_2). As a result, the minimum motivating reward of G is $\mu^*_{\text{pnl}}(G, \beta) = \tilde{c}(s, t, \beta)/\beta = \beta^{-2}$. Conversely, the topology of G makes it impossible to assign intermediate rewards to G that incentivize the agent to leave P. The reason is that intermediate rewards can only be placed on P itself. However, this would only increase the agent's incentive to stay on P. Thus, we may assume that the minimum motivating collected reward equals the minimal motivating reward, i.e., $\mu^*_{\text{rwd}}(G, \beta) = \mu(G, \beta)$. According to Proposition 3.1.1, we get $\mu(G, \beta) \geq \beta^{-n}$ and conclude that the cost efficiency ratio of G is β^{-n+2}. This proves tightness of the upper bound. □

5.4 Computing Motivating Reward Configurations with Budget Constraints

So far we have gained insight into the conceptual power of reward based incentives. As always, we continue with the computational complexity of destining this type of incentive. Our goal is to design motivating reward configurations r whose collected reward matches the minimum motivating collected reward within an arbitrarily small precision parameter $\varepsilon > 0$, i.e., $q(G_r, \beta) - \varepsilon \leq \mu^*(G, \beta)$ for a given task graph G and an agent with present-bias β. Similar to the previous chapter, we call such reward configurations almost optimal. The reason why we consider almost optimal reward configurations instead of optimal ones is that the latter may not exist in G according to the previous section.

To assess the complexity of designing almost optimal reward configurations, it is instructive to take a closer look at the following decision problem called MOTIVATING REWARD CONFIGURATION (MRC).

Definition 5.4.1 (MRC)**.** The problem of deciding whether a task graph $G = (V, E, c, r)$ admits a reward configuration $r : E \to \mathbb{R}_{\geq 0}$ that is motivating for a given present-bias $\beta \in (0, 1]$ and stays within a fixed budget $b \in \mathbb{R}_{\geq 0}$, i.e., $q(G_r, \beta) \leq b$.

Similar to penalty based incentives, it is not straight forward to solve MRC via an exhaustive search since G admits an infinite number of possible reward configurations. Nevertheless, a slight adaptation of the algorithmic ideas laid out in the proof of Proposition 4.3.1 confirms that MRC $\in$ NP.

Proposition 5.4.1. *MRC is contained in NP.*

Proof. Let $G = (V, E, c, r)$ be the task graph of an MRC instance $\mathcal{I}$ and assume that $\mathcal{I}$ admits a motivating reward configuration that stays within a given budget b. If this reward configuration can be encoded in polynomial space with respect to $\mathcal{I}$, then it can also serve as a certificate to verify that $\mathcal{I}$ is feasible. To establish the proposition, it therefore suffices to show that I admits a motivating reward configuration that stays within budget and has polynomial encoding length.

Similar to Proposition 4.3.1 we prove the existence of such a reward configuration by modeling $\mathcal{I}$ as a linear program. For this purpose let r^* be an arbitrary motivating reward configuration that stays within budget and let W be the set of nodes that an agent with present-bias β visits in G_{r^*}. Furthermore, let $P_v = v, \ldots, t$ be a path that minimizes the perceived net cost at v with respect to r^*, i.e., $\tilde{c}_{r^*}(v, \beta) = \tilde{c}_{r^*}(P_v, \beta)$. Clearly, any reward configuration r that satisfies the following constraints is also motivating and stays within the given budget

$$
\begin{aligned}
\sum_{v \in P} r(v) &\leq b && \text{for all } P = s, \ldots, t \subseteq W \\
\tilde{c}_r(P_v, \beta) &\leq 0 && \text{for all } v \in W \setminus \{t\} \\
\tilde{c}_r(P_v, \beta) &< \tilde{c}_r(P, \beta) && \text{for all } v \in W \setminus \{t\} \\
& && \text{and all } P = v, u, \ldots, t \text{ such that } u \notin W \\
r(e) &\geq 0 && \text{for all } e \in E.
\end{aligned}
$$

By definition of $\tilde{c}_r$ all of the above inequalities are linear with respect to the extra cost r. Replacing the strict inequalities in the third set of constraints by non-strict inequalities therefore yields a linear program without a specific objective function. Clearly, the feasible region of this linear program is non-empty as it contains r^*. As a result, the program also contains a *basic solution*. Since each constraint that is active in this basic solution can be encoded in polynomial space with respect to $\mathcal{I}$, the basic solution itself can also be encoded in polynomial space.

To complete the proof, we need to recover the strict inequalities in the third set of constraints. For this purpose, let $r_{v,P}$ be a solution to the linear program that maximizes the difference between $\tilde{c}_r(P,\beta)$ and $\tilde{c}_r(P_v,\beta)$ for a given node $v \in W \setminus \{t\}$ and path $P = v, u, \ldots, t$ such that $u \notin W$. By choice of the objective function, $r_{v,P}$ must satisfy $\tilde{c}_r(P_v,\beta) < \tilde{c}_r(P,\beta)$. Furthermore, we may assume that $r_{v,P}$ can be encoded in polynomial space with respect to $\mathcal{I}$.

Now, let $\bar{r}$ be the arithmetic mean over $r_{v,P}$ for all possible combinations of v and P. Clearly, $\bar{r}$ is a feasible solution of the linear program that satisfies each inequality of the third set of constraints in the strict sense. Moreover, $\bar{r}$ can be encoded in polynomial space since it is composed of at most exponentially many solutions $r_{v,P}$. As a result, $\bar{r}$ is a suitable certificate for $\mathcal{I}$. □

The more interesting question is whether MRC ∈ P. In the case of $\beta = 1$, MRC is indeed solvable in polynomial time. To see this, recall that an agent with a present-bias of $\beta = 1$ simply follows a cheapest path from s to t. Without loss of generality, all rewards may therefore be placed at the target node t. As a result, it is easy to see that the minimum motivating collected reward is exactly the cost of a cheapest path from s to t. Clearly, this cost can be computed in polynomial time. However, in the more general case of $\beta \in (0,1)$, MRC is much less likely to admit a polynomial time algorithm. To substantiate this claim, we devise a reduction from SET PACKING (SP) [12].

Definition 5.4.2 (SP)**.** The problem of deciding whether a collection of finite sets $S_1, \ldots, S_\ell$ contains $k \leq \ell$ mutually disjoint sets.

Note that the reduction presented in the proof of the following theorem is taken from our work on penalty and reward based incentives [2].

Theorem 5.4.2. *MRC is NP-hard for any bias factor $\beta \in (0,1)$, even if $b = 0$.*

Proof. Let $\mathcal{I}$ be an arbitrary instance of SP that consists of ℓ sets $S_1, \ldots, S_\ell$ and an integer $k \leq \ell$. Furthermore, assume that $\beta \in (0,1)$ is an arbitrary fixed value independent of $\mathcal{I}$. Our goal is to construct a task graph $G = (V, E, c, r)$ such that the resulting MRC instance $\mathcal{J}$ satisfies the following two properties:

(a) If $\mathcal{I}$ has k mutually disjoint sets, then $\mathcal{J}$ admits a motivating reward configuration for a budget of 0.

(b) If $\mathcal{J}$ admits a motivating reward configuration for a budget of 0, then $\mathcal{I}$ has k mutually disjoint sets.

This establishes NP-hardness of MRC in the special case of $b = 0$. At the end of the proof, we briefly sketch a method to adjust this reduction to an arbitrary $b > 0$ that is independent of $\mathcal{I}$.

As the first step of the proof, we construct the task graph G. Figure 5.3 depicts G for a small sample instance of SP. In general, G consists of a source s, a target t and k *layers* of nodes $v_{i,j}$ with $1 \leq i \leq k$ and $1 \leq j \leq \ell$. For each $v_{i,j}$ with $i < k$ there is a so-called *regular edge* to each node $v_{i+1,j'}$ of the next layer. To maintain readability, Figure 5.3 omits these edges. In addition to the regular edges between layers, there is also a regular edge from s to each node $v_{1,j}$ on the bottom layer and an edge from each node $v_{k,j'}$ on top layer to t. The idea behind this construction is that the agent walks from s to t along the regular edges of G in such a way that the nodes $v_{1,j}, \dots, v_{k,j'}$ of her walk correspond to a collection of k mutually disjoint sets $S_j, \dots, S_{j'}$. All regular edges that do not end in t charge a cost of $1 - \beta - \varepsilon$, where ε is a small but positive value satisfying

$$\varepsilon < \min\Big\{\frac{(1-\beta)^2}{k}, \frac{\beta - \beta^2}{k - 1 + \beta}\Big\}.$$

The other regular edges, i.e., the ones ending in t, are free of charge.

To motivate the agent to climb the layers, we add *shortcuts* to G that connect each node $v_{i,j}$ to t via an intermediate node $w_{i,j}$. The first edge $(v_{i,j}, w_{i,j})$ has a cost of 1, while the second edge $(w_{i,j}, t)$ is free of charge. Note that due to the cost of the first shortcut edge, a reward of $r(w_{i,j}) < 1/\beta$ can be placed at $w_{i,j}$ without the agent claiming it. Conversely, a reward of $r(w_{i,j}) \geq (1 - \varepsilon)/\beta$ is sufficient to make the incoming regular edges of $v_{i,j}$ motivating.

We finish the construction by drawing a path from each node $v_{i,j}$ to all nodes $w_{i',j'}$ for which $i' < i$ and $S_j \cap S_{j'} \neq \emptyset$. These so-called *backward paths* consist of two edges: the first one is free, but the second costs $(1 - \beta - k \cdot \varepsilon)/(\beta - \beta^2)$. The purpose of these paths is to enforce the disjointness constraint of $\mathcal{I}$. In the following paragraphs we address this idea in greater detail. But first note that the final task graph is acyclic and can be

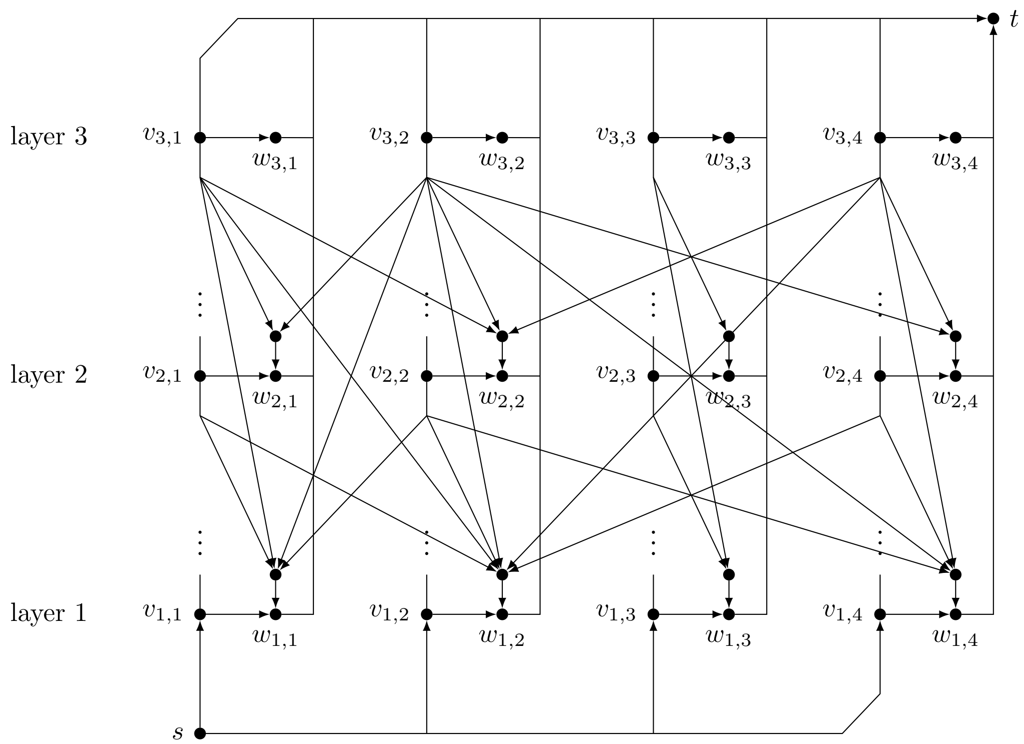

Figure 5.3: Reduction from the SP instance: $S_1 = \{a, b\}$, $S_2 = \{a, c\}$, $S_3 = \{d, e, f\}$, $S_4 = \{c, g\}$ and $k = 3$

constructed in polynomial time and space with respect to the SP instance $\mathcal{I}$. The latter observation holds true because the encoding length of the edge cost is independent of $\mathcal{I}$.

We proceed with the proof of statement (a), i.e., we show that $\mathcal{J}$ admits a reward configuration that is motivating for a budget of 0 whenever $\mathcal{I}$ contains a collection of k mutually disjoints sets. Assuming that such a collection exists, assign each of its sets S_j to a distinct layer i of G and set the reward $r(w_{i,j}) = (1 - \varepsilon)/\beta$. We refer to the corresponding node $v_{i,j}$ as *active*. To prove that r is motivating, it suffices to argue that r incentivizes the agent to climb from one active node to the next active node without ever taking a shortcut. As all rewards are placed within shortcuts, this also implies that the agent does not collect any reward and a budget of $b = 0$ is sufficient.

For a formal analysis assume that the agent is located at an active node $v_{i,j}$ with $i < k$. At this point, she has four options. The first one is to follow the shortcut to $w_{i,j}$. However, this has a positive perceived net cost of $\tilde{c}_r(v_{i,j}, w_{i,j}, \beta) = \varepsilon$ and therefore it is not motivating. Neither is the agent's second option, which is to take a backward path. The reason is that all backward paths have a positive cost, but by construction of G and

r no backward path that originates at $v_{i,j}$ leads to a reward. The agent's third option is to take the regular edge to the active node $v_{i+1,j'}$ of the next layer $i+1$. In this case the perceived net cost is at most

$$\tilde{c}_r(v_{i,j}, v_{i+1,j'}, \beta) \leq (1-\beta-\varepsilon) + \beta \cdot \left(1 - \frac{1-\varepsilon}{\beta}\right) = 0$$

if the agent plans to take the shortcut at $v_{i+1,j'}$. Note that this is motivating. Finally, the agent's fourth option is to plan along a path P whose initial edge is regular but leads to an inactive node. To prove that no such path P can be motivating, we make another case distinction, this time on P:

First, P may include a backward path. In this case, at most one reward can be located on P. As a result, the perceived net cost is at least

$$\tilde{c}_r(P, \beta) \geq (1-\beta-\varepsilon) + \beta \cdot \left(\frac{1-\beta-k\cdot\varepsilon}{\beta-\beta^2} - \frac{1-\varepsilon}{\beta}\right) = \frac{k}{1-\beta} \cdot \left(\frac{(1-\beta)^2}{k} - \varepsilon\right) > 0.$$

The inequality holds true by choice of ε and therefore P cannot be motivating. Secondly, P may take the shortcut at the next layer $i+1$. However, because this shortcut is not associated with an active node, no reward is placed on it and P cannot be motivating. Thirdly, P may take a shortcut on a layer $i' > i+1$. In this case, P contains at least two regular edges but at most one reward. The resulting net cost is

$$\begin{aligned}\tilde{c}_r(P, \beta) &\geq (1-\beta-\varepsilon) + \beta \cdot \left((1-\beta-\varepsilon) + 1 - \frac{1-\varepsilon}{\beta}\right) \\ &= \beta(1-\beta-\varepsilon) > \beta \cdot \left(\frac{(1-\beta)^2}{k} - \varepsilon\right) > 0.\end{aligned}$$

Similar to the first case of P the last inequality holds true by choice of ε and P is again not motivating. Finally, P may neither contain a backward path nor a shortcut. But in this case P does not contain a reward either and the positive cost of the first edge of P implies that P is not motivating.

To complete the proof of (a), we consider the corner cases in which the agent is located at s or at the active node $v_{k,j}$ of the top layer. At s her only option is to take a regular edge. As argued before, the only regular edge that is motivating is the one that ends at the active node of the bottom layer. Conversely, at $v_{k,j}$, the agent has only one regular edge to choose from. Because this edge is free of charge and ends at t, it is certainly motivating. Furthermore, we already know that none of the other outgoing edges of $v_{i,j}$, i.e., the shortcut and backward paths, are motivating. All in all, we conclude that the agent walks from s to t by climbing from one active node to the next.

Next, we consider statement (b). Assuming that $\mathcal{I}$ has a reward configuration r that is motivating for a budget of $b = 0$, our goal is to prove that $\mathcal{J}$ admits a collection of k mutually disjoint sets. For this purpose, consider an arbitrary walk of the agent through G_r. It is crucial to observe that this walk must not contain a shortcut or backward path. The reason is that the agent only enters shortcuts and backward paths if a positive reward is placed on them. However, in this case the agent also collects the reward unless she quits before. Both scenarios contradict the assumption that r is motivating for a budget of $b = 0$. As a result the agent visits exactly one node $v_{i,j}$ per layer i on her walk from s to t. Accordingly, we call any node $v_{i,j}$ that lies on one of the agent's paths an *active* node. Note that more than one active node per layer is possible.

In the following we prove that r needs to place a reward of $r(w_{i,j}) \geq (1-\varepsilon)/\beta$ on each shortcut of an active node $v_{i,j}$. As an auxiliary hypothesis we furthermore argue that the perceived net cost of $v_{i,j}$ is at least $(i-k)\cdot\varepsilon$. Assuming that $v_{i,j}$ is a node of the top layer, i.e., $i = k$, the auxiliary hypothesis is easy to see. After all, the agent must not take a shortcut or backward path at $v_{k,j}$. Her only remaining option is to cross the direct regular edge to t, which is free of charge. Moreover, r cannot place a reward on t without violating the budget constraint. Consequently, $v_{k,j}$ has a perceived net cost of $\tilde{c}_r(v_{k,j}, \beta) = 0 \geq (k-k)\cdot\varepsilon$.

To see that $r(w_{i,j}) \geq (1-\varepsilon)/\beta$ holds true for an arbitrary active node $v_{i,j}$, we consider an inductive argument. Our assumption is that $v_{i,j}$ satisfies the auxiliary hypothesis, i.e., $\tilde{c}_r(v_{i,j}, \beta) \geq (i-k)\cdot\varepsilon$. Furthermore, let $v_{i-1,j'}$ be an active node of the previous layer from which the agent plans to climb to $v_{i,j}$. In the special case of $i = 1$, assume that $v_{i-1,j'} = s$. By choice of $v_{i,j}$ and $v_{i-1,j'}$, there must exist a path $P = v_{i-1,j'}, v_{i,j}, \ldots, t$ that minimizes the perceived net cost of $v_{i-1,j'}$. To close in on the exact form of $P = v_{i-1,j'}, v_{i,j}, \ldots, t$, we consider a case distinction with respect to the edge type of the initial edge e of the subpath $P' = v_{i,j}, \ldots, t$.

In general, there are three choices for e. First, e could be a regular edge. If $i+1 = k$, this means that P' goes directly to t and r cannot place any reward onto P' without violating the budget constraint. Because the initial edge of P has a positive cost, it immediately follows that P is not motivating. But this contradicts the assumption that $v_{i-1,j'}$ is active. Conversely, if $i+1 < k$, there is a difference between the perceived net cost of P and P' of value

$$\begin{aligned}\tilde{c}_r(P,\beta) - \tilde{c}_r(P',\beta) &= c(v_{i-1,j'}, v_{i,j}) - r(v_{i,j}) - (1-\beta)\cdot c(e)\\ &= (1-\beta-\varepsilon) - 0 - (1-\beta)\cdot(1-\beta-\varepsilon) = \beta\cdot(1-\beta-\varepsilon).\end{aligned}$$

Note that $r(v_{i,j}) = 0$ due to the budget constraint. Together with the auxiliary hypothesis, which implies that the perceived net cost of P' is at least $(i-k)\cdot\varepsilon$, it follows that P is not motivating

$$\begin{aligned}\tilde{c}_r(P,\beta) &= \tilde{c}_r(P',\beta) + \beta\cdot(1-\beta-\varepsilon) \geq (i-k)\cdot\varepsilon + \beta\cdot(1-\beta-\varepsilon)\\ &= \big(k-i+\beta\big)\cdot\Big(\frac{\beta-\beta^2}{k-i+\beta}-\varepsilon\Big) > 0.\end{aligned}$$

The last inequality holds true by choice of ε. Again, we arrive at a contradiction to the assumption that $v_{i-1,j'}$ is active. Secondly, e could be the initial edge of a backward path. In this case, the difference between the perceived net cost of P and P' is

$$\tilde{c}_r(P,\beta) - \tilde{c}_r(P',\beta) = (1-\beta-\varepsilon) - 0 - (1-\beta)\cdot 0 = 1-\beta-\varepsilon.$$

The fact this difference is even greater than before contradicts the assumption that $v_{i-1,j'}$ is active. This leaves a shortcut edge as the only remaining choice for e. In particular, we may assume that P is of the form $P = v_{i-1,j'}, v_{i,j}, w_{i,j}, t$. As a result, r must place a reward greater or equal to $(1-\varepsilon)/\beta$ on $w_{i,j}$ to make P motivating.

To complete the inductive argument, it remains to show that the auxiliary hypothesis holds true for the active node $v_{i-1,j'}$. Similar to the previous paragraph, it is helpful to consider the difference in the perceived net cost of P and P'. Keeping in mind that the initial edge of P' is a shortcut edge, this difference is

$$\tilde{c}_r(P,\beta) - \tilde{c}_r(P',\beta) = (1-\beta-\varepsilon) - 0 - (1-\beta)\cdot 1 = -\varepsilon.$$

Together with the induction hypothesis, which implies that the perceived net cost of P' is at least $(i-k)\cdot\varepsilon$, we obtain

$$\tilde{c}_r(v_{i-1,j'},\beta) = \tilde{c}_r(P,\beta) = \tilde{c}_r(P',\beta) - \varepsilon \geq (i-k)\cdot\varepsilon - \varepsilon = \big((i-1)-k\big)\cdot\varepsilon.$$

This concludes the induction.

To summarize, recall that each active node $v_{i,j}$ has a reward of $r(w_{i,j}) \geq (1-\varepsilon)/\beta$ allocated to its shortcut and its perceived net cost is at most $(i-k)\cdot\varepsilon \geq (1-k)\cdot\varepsilon$. This means that there can be no backward path from an active node $v_{i,j}$ to another active node $v_{i',j'}$. Otherwise, the perceived net cost for following the path P from $v_{i,j}$ to t via $w_{i',j'}$ would be at most

$$\tilde{c}_r(P,\beta) \leq \beta\cdot\Big(\frac{1-\beta-k\cdot\varepsilon}{\beta-\beta^2} - \frac{1-\varepsilon}{\beta}\Big) = \frac{(1-k-\beta)\cdot\varepsilon}{\beta-\beta^2} < \frac{(1-k)\cdot\varepsilon}{\beta-\beta^2} < (1-k)\cdot\varepsilon,$$

which violates the bound on the perceived net cost of $v_{i,j}$. By construction of G, this means that the active nodes $v_{i,j}$ along any of the agent's walks correspond to a collection of k mutually disjoint sets. We conclude that $\mathcal{I}$ has a feasible solution.

To complete the proof, we sketch a generalized version of the reduction to prove NP-hardness for an arbitrary budget $b > 0$. For this purpose, consider a slightly modified version of G in which the target node is renamed to t' and a new target node t is inserted. Both nodes are connected via an edge (t', t) of cost $\beta \cdot b$. This means that a reward of b or more must be placed at t for the agent to cross (t', t) and complete G. Due to the placement of the reward, the agent is also certain to collect it whenever she completes G. Thus, each motivating reward configuration r that satisfies the budget constraint must set $r(t) = b$. Moreover, no further rewards must be collected by the agent.

To compensate for the extra reward at t, some of the edges in G become more expensive. The price of each regular edge that ends at t' as well as the price of the initial edge of each shortcut increases by an additional cost of $(1-\beta)\cdot\beta\cdot b$. Their new cost is $(1-\beta)\cdot\beta\cdot b$ and $1+(1-\beta)\cdot\beta\cdot b$ respectively. The price of the remaining regular edges rises by $(1-\beta)^2\cdot\beta\cdot b$ to $(1-\beta-\varepsilon)+(1-\beta)^2\cdot\beta\cdot b$. Finally, the price of the second edge of each backward path rises by $(1-\beta)\cdot b$ to $(1-\beta-k\cdot\varepsilon)/(\beta-\beta^2)+(1-\beta)\cdot b$. All other edges have the same cost as before.

Assuming that the encoding length of b is independent of $\mathcal{I}$, it is easy to see that the resulting $\mathcal{J}$ can be constructed in polynomial time and space with respect to $\mathcal{I}$. Moreover, the same line of reasoning presented for $b = 0$ implies that the agent is motivated to travel from s to t' if and only if the sets $S_1, \ldots, S_\ell$ admit a collection of k mutually disjoint sets. To verify this, it is helpful to observe that the reward at t and the additional edge costs cancel each other in the perceived net cost of paths originating at active nodes. □

A particularly noteworthy detail about Theorem 5.4.2 is that it applies to MRC instances with a budget of 0. As a result, it is also NP-hard to approximate the minimum motivating reward $\mu^*(G, \beta)$ within any finite factor of its actual value. Unfortunately, this implies that almost optimal reward configurations are not only difficult to compute but also unlikely to admit any polynomial time approximation within a finite factor.

5.5 Mixed Incentives

To overcome the computational difficulties inherent to the design of reward based incentives, we conclude this chapter with the brief study of a more general incentive design setting that permits the use of rewards, penalty fees and prohibitions at the same time.

From a conceptual perspective this combination is much more powerful than the mere sum of its parts. The reason is that a simultaneous placement of extra cost and intermediate rewards permits the construction of arbitrarily exploitative rewards. For instance, let ε be a small but positive quantity and consider an edge (v, u) with a reward of $r(u) = (c(v, u) + h(v, u))/\beta - \varepsilon$ placed at u. Assuming that the agent is located at v, it is easy to see that she has no incentive to cross (v, u) if no additional reward is offered to her. However, whenever the agent considers (v, u) from a distance, she perceives a net reward of

$$\begin{aligned}\beta \cdot \big(r(u) - c(v, u) - h(v, u)\big) &= \beta \cdot \Big(\frac{c(v, u) + h(v, u)}{\beta} - \varepsilon - c(v, u) - h(v, u)\Big) \\ &= (1 - \beta) \cdot \big(c(v, u) + h(v, u)\big) - \beta \cdot \varepsilon.\end{aligned}$$

Considering that the extra cost $h(v, u)$ can be chosen by the designer, the above equation implies that the excess reward of (v, u) may also become arbitrarily high if $\beta < 1$. Consequently, we can use (v, u) to draw the agent towards v without the need to pay out the reward placed u. We call such an edge an *attractor*.

Based on the idea of attractors the following straight forward approach to designing cost efficient incentives suggests itself. Assume that G contains a node v with at least two outgoing edges (v, w) and (v, u). In this case, one of the two edges, let us say (v, u), can be used as an attractor that keeps the agent motivated while traveling from s to v. Once the agent is located at v she can then be motivated to cross the other edge, i.e., (v, w), by placing a sufficient reward at t. This way only the reward at t is paid out.

To minimize the reward at t, it is crucial that the maximum perceived cost of the edges of the path $P_{v,w} = v, w, \ldots, t$ that the agent follows from v to t is as low as possible. Without loss of generality, we may assume that all nodes $v' \in P_{v,w} \setminus \{v\}$ only have successors nodes w' that are themselves contained in $P_{v,w}$. Otherwise, we could reduce cost by constructing the attractor at an edge (v', u') with $u' \notin P_{v,w}$. Since the internal nodes of $P_{v,w}$ have no outgoing edges leading out of $P_{v,w}$, we refer to $P_{v,w}$ as a *home stretch*. More formally, a home stretch $P_{v,w}$ is defined by the following two properties:

(a) If a node $v' \in P_{v,w} \setminus \{v\}$ has an outgoing edge (v', w'), then $w' \in P_{v,w}$.

(b) The node v either has an outgoing edge (v, u) such that $u \notin P_{v,w}$ or $v = s$.

Note that the path of a home stretch $P_{v,w}$ is distinctly determined by its first edge (v, w). To see this, assume (v, w) admits two different paths that satisfy the home stretch properties. In this case the paths need to fork at some node $v' \neq v$ and reunite at some other node w'. The later observation holds true because both paths end at t.

However, since G does not admit parallel edges, at least one of the two paths violates property (a). This contradicts the assumption that both paths are home stretches.

Using standard graph search algorithms, it is not too difficult to construct the home stretch associated with a particular edge of G in polynomial time. In particular it is possible to enumerate all home stretches of G and select one that minimizes the maximum perceived cost of its edges in polynomial time. Let $P_{v,w}$ be such a homestretch. In the special case of $v = s$ a motivating reward configuration is easily obtained by placing a sufficiently large reward onto t. Otherwise, if $v \neq s$, it is possible to construct a motivating combination of reward, penalty and prohibition based incentives as follows: First, compute a sufficiently large reward that motivates the agent to traverse the homestretch. Secondly, compute a path $R = v, u, \ldots, t$ and place an attractor at (v, u). For this purpose, R should satisfy $u \notin P_{v,w}$. Finally, compute a path $Q = s, \ldots, t$ that leads the agent from s to v. To make sure that the agent does not leave Q too early, remove all edges of G that are neither part of Q, R, $P_{v,w}$ or a parallel edge of $P_{v,w}$, i.e., an edge that starts and ends at a node of $P_{v,w}$. According to these algorithmic ideas we obtain the following polynomial time algorithm, which we call GreedyAttractor.

Algorithm 5: GreedyAttractor

Input: task graph G, present-bias β
Output: edge set F, cost configuration h, reward configuration r

foreach $v \in V$ **do** $r(v) = 0$;
foreach $e \in E$ **do** $h(e) = 0$;
$P_{v,w} \leftarrow$ home stretch that minimizes the maximum perceived cost of its edges;
$r(t) \leftarrow \max\{\tilde{c}(e, \beta) \mid e \in P_{v,w}\}/\beta$;
if $v = s$ **then** $F \leftarrow \emptyset$;
else
  $Q \leftarrow$ path from s to v;
  $R \leftarrow$ path from v to t whose initial edge (v, u) is different from (v, w);
  $F \leftarrow E \setminus (Q \cup R \cup P_{v,w} \cup \{(v', w') \mid v' \in P_{v,w} \setminus \{v\}\})$;
  $\alpha \leftarrow \sum_{e \in Q \cup R} c(e)$; $h(v, u) \leftarrow \alpha/(1 - \beta) + 1 + r(t)$; $r(u) \leftarrow \alpha/(\beta \cdot (1 - \beta)) + 1$;
return F, h, r;

An illustration of the paths Q, R and $P_{v,w}$ constructed by GreedyAttractor is depicted in Figure 5.4. To verify that the algorithm indeed yields motivating incentives, it is instructive to trace the agent's walk through G. By construction of F, the first part of the walk follows the path Q. When located at a node $v' \in Q \setminus \{v\}$, the agent perceives a cost of at most $\alpha + \beta \cdot h(v, u)$ for taking the paths Q and R to t. Conversely, the perceived reward of this path is $\beta \cdot (r(u) + r(t))$. By definition of $h(v, u)$ and $r(u)$,

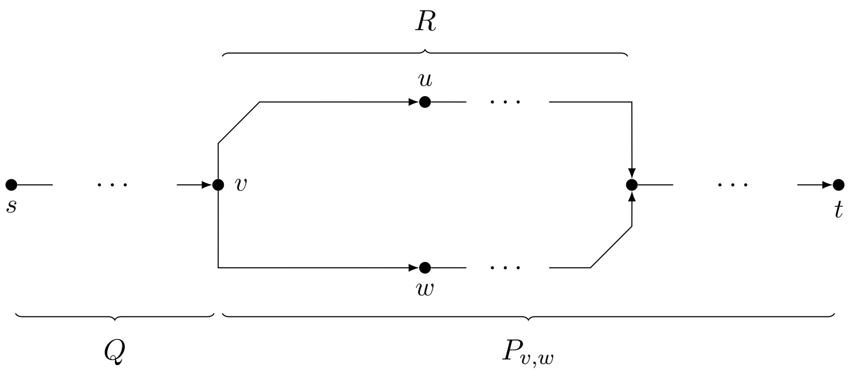

Figure 5.4: Structure of $P_{v,w}$, Q and R

this suffices to cover the cost

$$\begin{aligned}\alpha + \beta \cdot h(v,u) &= \alpha + \beta \cdot \Big(\frac{\alpha}{1-\beta} + 1 + r(t)\Big) \\ &= \beta \cdot \Big(\frac{\alpha}{\beta \cdot (1-\beta)} + 1 + r(t)\Big) = \beta \cdot \big(r(u) + r(t)\big).\end{aligned}$$

As a result, the agent is motivated to move along Q until she reaches v. At this point the perceived cost of the edge (v,u) is at least $h(v,u)$. However, the perceived reward remains $\beta \cdot (r(u) + r(t))$. Clearly, this reward is not sufficient to motivate the agent to enter R

$$\begin{aligned}h(v,u) &= \frac{\alpha}{(1-\beta)} + 1 + r(t) > \frac{\alpha}{(1-\beta)} + \beta \cdot \big(1 + r(t)\big) \\ &= \beta \cdot \Big(\frac{\alpha}{\beta \cdot (1-\beta)} + 1 + r(t)\Big) = \beta \cdot \big(r(u) + r(t)\big).\end{aligned}$$

The only other choice available to the agent is to enter the home stretch $P_{v,w}$. By definition of $P_{v,w}$ and $r(t)$, we know that the edge (v,w) as well as all other edges of $P_{v,w}$ are motivating. As a result, the agent travels along the home stretch or its parallel edges until she eventually reaches t.

All that remains is to investigate how well GreedyAttractor performs with respect to an optimal solution. As it turns out, the collected reward that is induced by the incentives of GreedyAttractor actually matches the collected reward of an optimal solution. This result is all the more surprising since neither reward nor penalty nor prohibition based incentives are likely to admit optimal and at the same time computationally tractable designs when considered on their own.

Theorem 5.5.1. *Given a task graph G and present-bias $\beta \in (0,1)$,* GreedyAttractor *constructs a reward configuration r, a cost configuration h and set of prohibited edges F whose collected reward $q(G_{r,h,F},\beta)$ is less or equal to the collected reward $q(G_{r',h',F'},\beta)$ of any other motivating combination of incentives r', h' and F'*

Proof. Recall that the collected reward $q(G_{r,h,F},\beta)$ corresponds to the reward $r(t)$ that GreedyAttractor places at t. By definition of $r(t)$, this means that $q(G_{r,h,F},\beta)$ is equal to the maximum perceived edge cost of the home stretch $P_{v,w}$ divided by β.

Now consider the path P' that an agent with present-bias β takes through an arbitrary motivating task graph $G_{r',h',F'}$. Clearly, the tail section of P' contains a home stretch $P_{v',w'}$. This follows directly from the definition of a home stretch. Furthermore, the same definition implies the impossibility of motivating the agent with exploitative rewards once she enters $P_{v',w'}$. In other words, we know that the agent collects all rewards that keep her motivated while traveling along $P_{v',w'}$. To make sure that the agent does not quit, we conclude that the collected reward $q(G_{r',h',F'},\beta)$ must be at least the maximum perceived edge cost of $P_{v',w'}$ divided by β. By choice of the homestretch $P_{v,w}$, we know that this value is at most $q(G_{r,h,F},\beta)$. □

6 Heterogeneous Agents

So far we have focused our attention on incentives that target a single person with a specific present-bias. However, in many real-life scenarios it is not possible, or at least not practical, to design customized incentives for each individual of a given population. Instead, it is often necessary to design a single incentive that motivates each member of a population equally well. The difficulty is that the population may be *heterogeneous* in the sense that the present-biases of its members differ from one person to the next. In the following chapter we investigate the formal implications of this generalization of the graphical model. In particular, we are interested in the conceptual loss of efficiency arising due to the need to address multiple individuals at the same time. Using penalty fees as our incentive of choice, we are able to bound the loss of efficiency by a constant factor of 2. Furthermore, we present a polynomial time algorithm to construct a matching assignment of penalty fees.

6.1 Preparing for the Big Race

Designing a single incentive that motivates a heterogeneous group of people is a complex task. One of the main challenges is that the graphical model tends to be extremely sensitive to the particular value of the present-bias β. Sometimes even the slightest change in β may result in very different and sometimes counter-intuitive behavioral patterns. Consider for instance the following scenario:

Assume that Alice wants to participate in a running event and asks her good friend Bob to join her. In the two week leading up to the event, the two runners can prepare for the race by choosing between a light workout plan A and a more demanding workout plan B. Workout plan A has a cost of 1 each week, whereas plan B has a cost of 3 in the first and 9 in the second week. However, workout plan B offers a more thorough preparation than A. As a result, the two runners incur a cost of 13 in the final race if they consistently choose A, but only a cost of 1 if they consistently choose B. Since the plans A and B are incompatible with each other, it is furthermore the case that a switch after one week results in an effort of 16 in the final race. Figure 6.1 models this setting

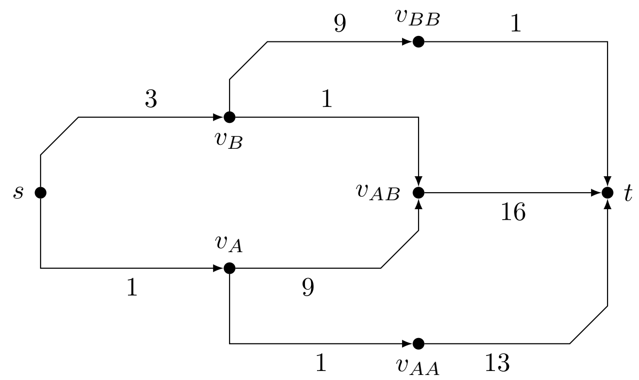

Figure 6.1: Task graph of the workout plans A and B

as a task graph. The nodes v_X and v_{XY} represent an agent's level of preparation after completing the workouts $X, Y \in \{A, B\}$. To transition between nodes, the agent must complete the workout associated with the corresponding edge or, if the edge happens to end at t, run the final race.

Recall that we set out to construct an instance of the graphical model that is ill-conditioned with respect to the present-bias parameter. For this purpose assume that Alice and Bob have an almost identical bias of $a = 1/2 - \varepsilon$ and $b = 1/2 + \varepsilon$ respectively. If their present-biases were exactly $1/2$, it is not difficult to see that they would consider the two workout plans to be equal options at the beginning of the first week. The reason is that the edges (s, v_A) and (s, v_B) would have identical perceived cost

$$\tilde{c}\Big(s, v_A, \frac{1}{2}\Big) = 1 + \frac{1}{2} \cdot (1 + 13) = 8 = 3 + \frac{1}{2} \cdot (9 + 1) = \tilde{c}\Big(s, v_B, \frac{1}{2}\Big).$$

However, since Alice is slightly more present-biased than Bob, she has a strict preference for workout plan A. More formally, it holds true that

$$\begin{aligned}\tilde{c}(s, v_A, a) &= 1 + a \cdot (1 + 13) = 8 - 14 \cdot \varepsilon \\ &\leq 8 - 10 \cdot \varepsilon = 3 + a \cdot (9 + 1) = \tilde{c}(s, v_B, a).\end{aligned}$$

Conversely, Bob has a strict preference for workout plan B considering that

$$\begin{aligned}\tilde{c}(s, v_A, b) &= 1 + b \cdot (1 + 13) = 8 + 14 \cdot \varepsilon \\ &\leq 8 + 10 \cdot \varepsilon = 3 + b \cdot (9 + 1) = \tilde{c}(s, v_B, b).\end{aligned}$$

Assuming that a sufficiently large reward is paid out upon completion of the race, it

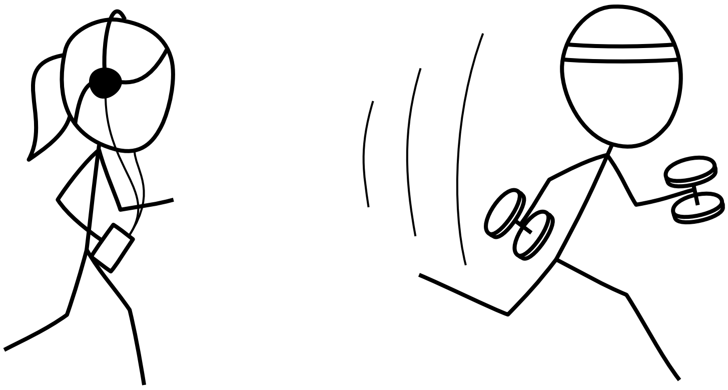

Figure 6.2: Alice and Bob in preparation for the race

follows that Alice moves to the node v_A, whereas Bob moves to v_B. Figure 6.2 illustrates the runners completing their respective workouts.

After the first week, it is time to reevaluate Alice and Bob's plans. In Bob's case, the perceived cost of his two new choices evaluates to $\tilde{c}(v_B, v_{AB}, b) = 9 + 16 \cdot \varepsilon$ and $\tilde{c}(v_B, v_{BB}, b) = 19/2 + \varepsilon$. Assuming that ε is small enough, this implies that Bob is inclined to switch to plan A; a strategy that results in a cost of 16 during the final race. Conversely, it is easy to see that Alice has no reason to change from plan A to plan B. As a result, her cost for running the final race is 13. It is astonishing that Alice ends up paying much less than Bob during the training and the race despite the fact that their biases are only marginally different. This observation may appear even more surprising considering that Bob is less biased than Alice.

The intuitive reason for this phenomenon is simple: First, observe that even a slight difference in the present-bias may cause two otherwise identical agents to take different outgoing edges at a given node of the task graph. In Alice and Bob's scenario, this happens at the initial node s. Secondly, once the agents' paths separate, they may develop independently of one another. Of course, the latter observation depends on the topology of the task graph. However, it is not hard to imagine scenarios in which the difference between the cost experienced by the two agents becomes arbitrarily large. In fact, it suffices to let one of the agents, once their paths have split, traverse a procrastination structure like the one described in the proof of Proposition 3.1.1. All in all, this goes to show how intricate the relation between the present-bias and the topology of the task graph may be; an observation that should be kept in mind when designing incentives for a heterogeneous population of agents.

6.2 Modeling Heterogeneous Populations

An elegant property of the graphical model is that the all the information needed to predict an agent's behavior is condensed in the single present-bias parameter β. In order to model a population of agents, it therefore suffices to collect their individual present-biases in a *present-bias set* $B \subset (0, 1]$. Whenever we have perfect information about each individual of the population, we may assume B to be a finite set. However, in many scenarios such detailed information is difficult, if not impossible, to obtain. In this case, it is often more convenient to think of B as an infinite set. More precisely, we define B to be a collection of closed intervals from the set $(0, 1]$ that confine the actual present-bias values of the population. This way the graphical model becomes much more robust to minor imprecision in the specification of the population; a property that is all the more desirable considering the observations of the previous section.

For technical reasons, we need to impose two additional assumptions on the structure of B: First, we assume that B is composed of finitely many closed intervals. This way the intersection of B with some other closed interval $I \in (0, 1]$ is either empty or contains a minimum and a maximum element. In particular, the set B itself must contain a minimum element $\min B$ and a maximum element $\max B$. To quantify the heterogeneity of B, we define the *range* $\tau = \max B / \min B$ of B to be the ration between these two values. Secondly, we assume that the intersection of B and I can be computed in polynomial time with respect to the encoding lengths of B and I.

Based on the notion of B, we consider a given task graph $G = (V, E, c, r)$ to be motivating if no agent with a present-bias from the set B quits on one of her walks through G. Clearly, this definition of motivating task graphs raises computational concern. After all, we only know how to verify whether a particular task graph is motivating for a single agent, a task that can be performed efficiently according to Proposition 2.3.2. However, since B may contain infinitely many present-biases, it is generally not possible to explicitly verify whether G is motivating for every single $\beta \in B$. Fortunately, we do not have to do this. According to the following Theorem, which is taken from our work on incentive design for uncertain present-bias values [4], it suffices to consider a finite subset $B' \subseteq B$ of size $\mathcal{O}(n^2)$. Note that the proof of the Theorem is partly inspired by a result of Kleinberg and Oren that bounds the number of paths that an agent may take through a task graph by $\mathcal{O}(n^2)$ [16].

Theorem 6.2.1. *For any task graph $G = (V, E, c, r)$ and present-bias set B a finite subset $B' \subseteq B$ of size $\mathcal{O}(n^2)$ exists such that G is motivating for all $\beta \in B$ if it is motivating for all $\beta \in B'$.*

Proof. The proof consists of two steps. First, we determine which present-bias values incline agents to traverse a given edge (v, w). More formally, let the set $B_{v,w}$ contain all values $\beta \in [0, 1]$ such that $\tilde{c}(v, w, \beta) = \tilde{c}(v, \beta)$. Our goal is to show that $B_{v,w}$ is a closed, possibly empty, subinterval of $[0, 1]$. In the second step, we collect the minimum values of all non-empty intersection of a set B_e and B in the set $B' = \{\min(B_e \cap B) \mid B_e \cap B \neq \emptyset\}$. Note that by our assumption on B and B_e the minimum of $B_e \cap B$ is guaranteed to exist as long as $B_e \cap B \neq \emptyset$. Furthermore, B' contains at most $|E|$ elements and therefore has a size of $\mathcal{O}(n^2)$. Using a proof by contradiction, we argue that G is motivating for all $\beta \in B$ if it is motivating for all $\beta \in B'$. This completes the proof.

To see that $B_{v,w}$ is a closed interval, it is helpful to observe how the perceived cost of (v, w) changes with respect to β. For this purpose, let $\ell_{v,w}(\beta) = c(v, w) + \beta d(w)$ denote the perceived cost $\tilde{c}(v, w, \beta)$ with respect to β. Clearly, the function $\ell_{v,w}$ is linear and so are the functions $\ell_{v,w'}$ associated with the incident edges of (v, w). Together, these functions yield the line arrangement $L_v = \{\ell_{v,w'} \mid (v, w') \in E\}$. Recall that the agent is only motivated to cross (v, w) if $\tilde{c}(v, w, \beta) \leq \tilde{c}(v, w', \beta)$ for all $(v, w') \in E$. The values of $\beta \in [0, 1]$ that satisfy this property are exactly those for which the line $\ell_{v,w}$ is on the lower envelope of the line arrangement L_v. From the basic structure of line arrangements we can conclude that $B_{v,w}$ must be a closed interval, see e.g. [10].

To conclude the proof, it remains to show that G is motivating for each $\beta \in B$ if it is motivating for each $\beta \in B'$. For the sake of contradiction assume that G is motivating for each $\beta \in B'$ but not for some $b \in B$. As a result there must exist a path P from s to t along which an agent with present-bias b may walk and which contains an edge (v, w) such that $\tilde{c}(v, w, b)/b > r$. Let $B_P = \bigcap_{e \in P} B_e$ be the set of all present-biases for which an agent may follow the path P. From this definition it is immediately apparent that b is contained in B_P and therefore $B_P \cap B \neq \emptyset$. We now consider the structure of B_P. Since each set B_e associated with an $e \in P$ is a closed interval, so is B_P. In particular, as $B_P \cap B$ is not empty, one of these sets B_e must satisfy $\min(B_e \cap B) = \min(B_P \cap B)$. Let $a = \min(B_e \cap B)$ denote the corresponding present-bias. By definition of a it holds true that $a \in B'$ and $a \leq b$. Moreover, considering that $a \in B_P$, we know that an agent with present-bias a may follow the path P. However, once this agent reaches (v, w), her perceived cost exceeds her perceived reward as

$$\frac{\tilde{c}(v, w, a)}{a} = \frac{c(v, w)}{a} + d(w) \geq \frac{c(v, w)}{b} + d(w) = \frac{\tilde{c}(v, w, b)}{b} > r.$$

Consequently, G is not motivating for a. But this contradicts the assumption that G is motivating for all $\beta \in B'$. □

It is interesting to observe that the proof of Theorem 6.2.1 does not only establish the existence of the set B', but also implies an algorithm to compute B' in polynomial time. As a result, it can be decided in polynomial time whether a given task graph G is motivating with respect to the present-bias set B. It should also be noted that Theorem 6.2.1 provides an efficient way to compute the minimal motivating reward $\mu(G, B)$, i.e., the minimum reward that must be placed at t to ensure that G is motivating for all agents whose present-bias is contained in B. For this purpose, it suffices to set $\mu(G, B) = \max\{\mu(G, \beta) \mid \beta \in B'\}$.

6.3 The Price of Heterogeneity

Having established a suitable framework for analyzing heterogeneous populations, we now focus on the design of cost efficient incentives. The difference to the previous chapters is that a single incentive needs to motivate each agent of a given population equally. We are particularly interested in the loss of efficiency that emerges when comparing such universal incentives to incentives that only address a single individual of the population. We call this hypothetical loss the *price of heterogeneity*. To quantify the price of heterogeneity for a given task graph G and preset-bias set B, let the minimum motivating reward $\mu^*(G, B)$ be redefined as

Definition 6.3.1 (Minimum Motivating Reward)**.** The minimum motivating reward $\mu^*(G, B)$ is the infimum over all rewards for which the task graph G admits a certain incentive that is motivating for all agents whose present-bias is contained in $B \subset (0, 1]$.

Due to its favorable conceptual and computational properties, our incentive of choice throughout this chapter is penalty fees. However, note that the price of heterogeneity could also be used for other incentives. In the case of reward based incentives, it may be sensible to consider the collected rewards rather than all the rewards that are laid out.

Having specified the minimum motivating reward $\mu^*(G, B)$, we can compare this quantity to the minimum motivating rewards $\mu^*(G, \beta)$ for specific values of $\beta \in B$. As a general rule of thumb one may expect to invest more if β is a small and less if β is large. The reason is that an agent who is strongly present-biased discounts future rewards more than her less present-biased counter parts and therefore she requires more reward to compensate for current cost. Keeping this observation in mind, it becomes clear that a direct comparison between $\mu^*(G, B)$ and $\mu^*(G, \beta)$ is not reasonable. After all, $\mu^*(G, B)$ applies to all values of B, which may include small values, whereas $\mu^*(G, \beta)$ applies to just one value of B, which may be comparatively large. Instead, it seems more sensible to compare $\mu^*(G, B)$ to the largest minimum motivating reward $\mu^*(G, \beta)$ of any present-bias $\beta \in B$. Consequently, we define the price of heterogeneity as follows:

Definition 6.3.2 (Price of Heterogeneity)**.** Given a task graph G and a present-bias set B, the price of heterogeneity is defined as

$$\frac{\mu^*(G, B)}{\sup\{\mu^*(G, \beta) \mid \beta \in B\}}.$$

Let us illustrate the price of heterogeneity by going back to Alice and Bob's scenario and assume that $B = \{a, b\}$ with $a = 1/2 - \varepsilon$ and $1/2 + \varepsilon$. It is easy to see that both agents minimize their maximum perceived cost along the path $P = s, v_B, v_{BB}, t$. This cost, which is either $\tilde{c}(v_B, v_{BB}, a) = 19/2 - \varepsilon$ or $\tilde{c}(v_B, v_{BB}, b) = 19/2 + \varepsilon$, provides two lower bounds for the reward that is required to motivate the agents. More formally, it holds true that $\mu^*(G, a) \geq (19/2 - \varepsilon)/(1/2 - \varepsilon)$ and $\mu^*(G, b) \geq (19/2 + \varepsilon)/(1/2 + \varepsilon)$. However, as we have seen in Section 6.1, neither Alice nor Bob are willing to follow P without external incentives. To discourage them from leaving P, we may assign an extra cost of $h(s, v_A) = 5 \cdot \varepsilon$ to (s, v_A) and $h(v_B, v_{AB}) = 1/2 + 16 \cdot \varepsilon$ to (v_B, v_{AB}). Observe that the cost configuration h does not affect the agents' maximum perceived cost along P, which they still experience at (v_B, v_{BB}). As a result, our bounds for $\mu^*(G, a)$ and $\mu^*(G, b)$ are tight and we get $\sup\{\mu^*(G, \beta) \mid \beta \in B\} = \mu^*(G, a)$. Moreover, h guides both agents along the same path. Consequently, $\mu^*(G, B) = \sup\{\mu^*(G, \{\beta\}) \mid \beta \in B\}$ holds true, which implies that the price of heterogeneity is 1.

As Alice and Bob's scenario demonstrates, cost configurations designed for a heterogeneous population are not necessarily less efficient than those designed for a specific individual of the population. In other words, we have seen a scenario in which the same cost configuration motivates each individual of a given population efficiently. But this raises the question whether all scenarios admit such a cost configuration or whether a real loss of efficiency is bound to occur in at least some scenarios when transitioning from individual to universal cost configurations. In other words, the question is whether the price of heterogeneity can become greater than 1. According to the following proposition, which is taken from [4], the answer to this question is yes.

Proposition 6.3.1. *There exists a family of task graphs and present-bias sets for which the price of heterogeneity converges to* 1.1*.*

Proof. Let $0 < a \leq 3/8$ be some present-bias such that $4/a$ is integral and consider a task graph G consisting of a directed path $v_0, v_1, \ldots, v_{12+4/a}$. We call this path the *regular path* and charge a cost of 2 on its first edge. All other edges of the regular path

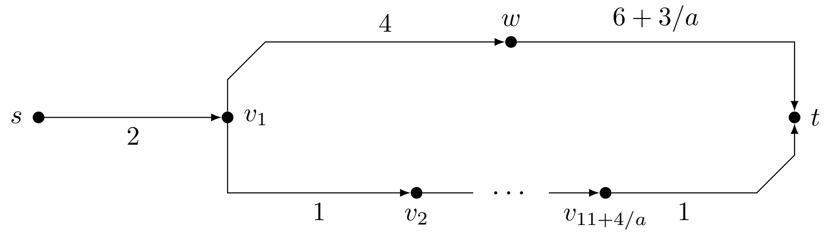

Figure 6.3: Task graph admitting a price of heterogeneity of $(9 + 11/(2 \cdot a))/(10 + 5/a)$

have a cost of 1. In addition to the regular path, we introduce a *shortcut* from v_1 to $v_{12+4/a}$ along a middle node w. The cost of the edges (v_1, w) and $(w, v_{12+4/a})$ is 4 and $6 + 3/a$ respectively. For the sake of convenience let $s = v_0$ and $t = v_{12+4/a}$. Figure 6.3 shows a sketch of G.

Now, assume that G is traversed by agents whose present-bias is either a or $b = 1/2$, i.e., $B = \{a, 1/2\}$. Our goal is to construct two cost configurations h_a and h_b that are motivating for a reward of $10 + 5/a$ when applied to agents with the respective present-bias. Consequently, it holds true that $\sup\{\mu^*(G, a), \mu^*(G, b)\} \leq 10 + 5/a$. We then continue to argue that a reward less than $9 + 11/(2a)$ is not sufficient to motivate both types of agents simultaneously, i.e., $\mu^*(G, B) \geq 9+11/(2\cdot a)$. As $(9+11/(2a))/(10+5/a)$ converges to $11/10$ for $a \to 0$, this establishes the proposition.

We begin with h_a. For this purpose let h_a assign no extra cost at all, i.e., $h_a(e) = 0$ for all edges e. Furthermore, assume the agent has a present-bias of a and a reward of $10+5/a$ is placed at t. When located at s, the agent's only choice is (s, v_1). If she plans to take the shortcut next, her perceived cost of (s, v_1) is at most $\tilde{c}_{h_a}(s, v_1, a) \leq 10\cdot a+5$. As this matches her perceived reward, she proceeds to v_1 where she faces two options: The first one is to take the shortcut for a perceived cost of $\tilde{c}_{h_a}(v_1, w, a) = 6\cdot a+7$. Considering that we have chosen a to satisfy $a < 1/2$, it follows that $\tilde{c}_{h_a}(v_1, w, a) > 10 \cdot a + 5$ and therefore the shortcut is not motivating. The second option is to take the regular path along the $11+4/a$ edges of cost 1, resulting in a perceived cost of $\tilde{c}_{h_a}(v_1, v_2, a) = 10\cdot a+5$. Similar to the situation at s, this cost matches her perceived reward and she proceeds to v_2. Since all remaining edges (v_i, v_{i+1}) have a perceived cost less than $\tilde{c}_{h_a}(v_1, v_2, a)$, the agent eventually reaches t and we conclude that h_a is motivating for the given reward and present-bias.

We continue to construct h_b by setting $h_b(v_1, w) = 1/(2 \cdot a)$ and $h_b(e) = 0$ for all other edges. In contrast to the previous scenario, assume the agent now has a present-bias of $b = 1/2$. The reward is still $10 + 5/a$, but its perceived value has changed to

$5 + 5/(2 \cdot a)$. When located at the initial node s, the agent's perceived cost is at most $\tilde{c}_{h_b}(s, v_1, b) \leq 7 + 7/(4 \cdot a)$ if she plans to take the shortcut afterwards. By choice of a it holds true that $a < 3/8$ and we get $\tilde{c}_{h_b}(s, v_1, b) \leq 5 + 5/(2 \cdot a)$. Consequently, the agent is motivated to proceed to v_1. At this point, she has to choose between the shortcut and the regular path. Her perceived cost of the former is $\tilde{c}_{h_b}(v_1, w, b) = 7+2/a$, whereas the latter has a perceived cost of $\tilde{c}_{h_b}(v_1, v_2, b) = 6 + 2/a$. Clearly, the regular path is her preferred choice and since $a < 1/2$, it is also a motivating one. Because all remaining edges (v_i, v_{i+1}) have a perceived cost less than $\tilde{c}_{h_b}(v_1, v_2, b)$, it follows that h_b is motivating for the given reward and present-bias as well.

It remains to show that no cost configuration can be motivating for both agents at the same time if the reward is less than $9 + 11/(2 \cdot a)$. For the sake of contradiction assume such a cost configuration h exists. Note that an agent with present-bias b must not enter the shortcut as her perceived cost $\tilde{c}_h(w, t, b) = 6 + 3/a$ exceeds her perceived reward of $9/2 + 11/(4 \cdot a)$ for any $a > 0$. However, if we do not assign extra cost to G, such an agent prefers the shortcut to the regular path when located at v_1. The difference in perceived cost is $\tilde{c}(v_1, v_2, b) - \tilde{c}(v_1, w, b) = 1/(2 \cdot a) - 1$. Consequently, h must assign an extra cost greater than $1/(2 \cdot a) - 1$ to the shortcut. Next consider an agent with a present-bias of a located at s. At this point her perceived cost for taking the shortcut is greater than $9 \cdot a + 11/2$, due to the extra cost assigned by h. Note that this cost exceeds her perceived reward. Her other option is to plan along the regular path. In this case her perceived cost is $11 \cdot a + 6$. Clearly, this is even more expensive and therefore contradicts the assumption that h is motivating for both agents simultaneously. □

6.4 Revisiting the Copied Cost Approximation

Considering that a heterogeneous population may impair the cost efficiency of penalty based incentives, i.e., the price of heterogeneity can become strictly greater than 1, the question of an upper bound on the price of heterogeneity arises. Ideally, we would like to design a cost configuration h whose minimal motivating reward $\mu(G_h, B)$ is within a constant factor ϱ of the minimum motivating reward $\mu^*(G, \min B)$ for any conceivable task graph G and any present-bias set B. Note that the existence of h would imply a universal bound of ϱ on the price of heterogeneity. In particular, the bound would be independent of the range of B, i.e., the degree of heterogeneity exhibited by a certain population. Surprisingly, we are able to construct a suitable cost configuration h for a relatively small bound of $\varrho = 2$. In fact, all we have to do is to slightly adapt the CopiedCostApprox algorithm of Chapter 4 in the following way:

Algorithm 6: CopiedCostApprox

Input: Task graph G, present-bias set B

Output: Cost configuration h

$P \leftarrow$ minmax path from s to t with respect to the cost perceived by an agent with present-bias $\min B$;
$\alpha \leftarrow \max\{\tilde{c}(e, \min B) \mid e \in P\}$;
foreach $v \in V \setminus \{t\}$ **do**
  $\sigma(v) \leftarrow$ successor node of v on a cheapest path from v to t;
$T = \{(v, \sigma(v)) \mid v \in V \setminus \{t\}\}$;
foreach $(v, w) \in E$ **do**
  if $(v, w) \notin P \cup T$ **then**
    $h(e) \leftarrow 2 \cdot \alpha / \min B + 1$;
  else if $v \in P$ **and** $w \notin P$ **then**
    $P' \leftarrow v, \sigma(v), \sigma(\sigma(v)), \ldots, t$;
    $u \leftarrow$ first node of P' different from v that is also a node of P;
    $h(v, w) \leftarrow$ cost of a most expensive edge of P' between v and u;
  else $h(e) \leftarrow 0$;
return h;

As in the original version of the algorithm, the above implementation of CopiedCostApprox consists of two simple steps: First, it computes a value α such that $\alpha / \min B$ is a lower bound on the reward necessary to motivate agents with a present-bias of $\min B$, i.e., $\mu^*(G, \min B) \geq \alpha / \min B$. In particular, this bound implies

$$\sup\{\mu^*(G, \beta) \mid \beta \in B\} \geq \frac{\alpha}{\min B}.$$

Secondly CopiedCostApprox constructs a cost configuration h such that a reward of $2 \cdot \alpha / \min B$ is sufficiently motivating for all $\beta \in B$, i.e., $\mu^*(G, B) \leq 2 \cdot \alpha / \min B$. As a result the price of heterogeneity can be at most 2. To see that h indeed satisfies the promised properties, it is instructive to brush up on some of its main ideas.

Recall that h assigns extra cost in such a way that any agent with a present-bias $\beta \in B$ traverses only two kinds of edges: The first kind is edges of P. Note that each such edge $(v, w) \in P$ is motivating for a reward of $\alpha / \min B$ if $\beta \geq \min B$. The reason is that

$$\begin{aligned} \tilde{c}(v, w, \beta) &= \beta \cdot \Big(\frac{c(v, w)}{\beta} + d(w)\Big) \leq \beta \cdot \Big(\frac{c(v, w)}{\min B} + d(w)\Big) \\ &= \beta \cdot \frac{\tilde{c}(v, w, \min B)}{\min B} \leq \beta \cdot \frac{\alpha}{\min B}. \end{aligned}$$

In particular, P is motivating for each $\beta \in B$. The second kind of edge corresponds to the edges of the cheapest path tree T induced by the successor relation σ. Recall that a path of the form $P' = v, \sigma(v), \sigma(\sigma(v)), \ldots, t$ is a cheapest path by definition of σ.

To keep agents on the edges of P and T, CopiedCostApprox assigns an extra cost of $h(e) = 2 \cdot \alpha / \min B + 1$ to all other edges. This raises their perceived cost to a value greater of equal to $2 \cdot \alpha / \min B + 1$; a price no agent is willing to pay for a perceived reward of $\beta \cdot 2 \cdot \alpha / \min B$. However, since no extra cost has been assigned to T so far, the perceived cost of the edges in P and T is unaffected by h at this point. In particular, all edges of P are still motivating for a reward of $\alpha / \min B$ and any present-bias $\beta \in B$. To keep agents from entering a costly σ-path $P' = v, \sigma(v), \sigma(\sigma(v)), \ldots, t$, CopiedCostApprox assigns an extra cost to the outgoing edges $(v, \sigma(v))$ of P, i.e., $v \in P$ but $\sigma(v) \notin P$. The extra cost $h(v, \sigma(v))$ is chosen to match the cost of a most expensive edge on P' between v and the next intersection of P' and P. This way, the resulting cost configuration h can at most double the perceived cost of any edge in P; see the proof of Theorem 4.5.1 for a precise argument. Furthermore, the perceived cost of any outgoing edge $(v, \sigma(v))$ of P is either high enough to keep agents on P or they do not encounter edges exceeding the perceived cost of $(v, \sigma(v))$ until they reenter P. Again, a precise argument is given in the proof of Theorem 4.5.1. We conclude that a reward of $2 \cdot \alpha / \min B$ is sufficiently motivating and therefore the price of heterogeneity cannot become greater than 2.

Theorem 6.4.1. *The price of heterogeneity is at most* 2.

Proof. The correctness of the theorem is a direct result of the above considerations and the arguments laid out in the proof of Theorem 4.5.1. □

In addition to the conceptual implications that CopiedCostApprox has on the price of heterogeneity, it is also interesting to consider its computational implications on the design of cost efficient cost configurations for a heterogeneous population. Clearly, this is a difficult problem in general. After all Theorem 4.6.1 indicates that even in the special case of a homogenous population, i.e., $|B| = 1$, no polynomial time approximation of the problem is possible within a constant factor of 1.08192 unless P = NP. However, considering the fact that CopiedCostApprox can be executed in polynomial time with respect to the encoding length of G and B, we are able to construct a cost configuration h that approximates the performance of any other cost configuration within a factor of at most 2.

Proposition 6.4.2. *Given a task graph G and present-bias set $B \subset (0,1]$,* CopiedCostApprox *yields a cost configuration h whose minimal motivating reward $\mu(G_h, B)$ is less than $2 \cdot \mu(G_{h'}, B)$ for any other cost configuration h'.*

Proof. Recall that CopiedCostApprox constructs a cost configuration that is motivating for a reward of $2 \cdot \alpha / \min B$ independent of the particular present-bias chosen from the set B. Furthermore, $\alpha / \min B$ is a lower bound on the reward required by any conceivable cost configuration h' to motivate an agent with a present-bias of $\min B$. Considering that $\mu(G_{h'}, \min B) \leq \mu(G_{h'}, B)$, we conclude that

$$\mu(G_h, B) \leq 2 \cdot \frac{\alpha}{\min B} \leq 2 \cdot \mu(G_{h'}, \min B) \leq 2 \cdot \mu(G_{h'}, B). \qquad \square$$

7 A Variable Present-Bias

The previous chapter addresses heterogeneous populations. A benefit of incentives that are designed in such a setting is that they are robust with respect to imperfect knowledge of a single person's present-bias as long as it is expressed by at least one member of the population. Our ambition in this chapter is to develop incentives that are robust with respect to an even greater degree of uncertainty. More precisely, we consider a setting in which a person's present-bias is not only drawn from a range of different values, but also changes within that range over time. The idea of this approach, which is inspired by work of Gravin et al. [13], is to compensate for behavioral phenomena that are not accounted for in Kleinberg and Oren's original version of the graphical model [16]. Similar to the previous chapter our goal is to quantify the conceptual loss of efficiency that arises due to the unpredictable variability of the present-bias. Furthermore, we are interested in the approximability of optimal incentives. In the case of penalty based incentives, both of these issues can be addressed with a single polynomial time algorithm. The resulting bounds are proportional to the range over which the present-bias varies. Although this result is in stark contrast with the constant bounds presented in the previous chapter, evidence that these results cannot be much improved exists.

7.1 Modeling Variability

To capture the notion of variability in the graphical model, it is convenient to think of the agent's present-bias not as a fixed parameter $\beta \in B$, but as a *present-bias configuration*, i.e., an assignment of present-bias values $\beta(v) \in B$ to the nodes $v \in V$ of a given task graph $G = (V, E, c, r)$. Consequently, whenever the agent reaches a node v, she acts according to her current present-bias value $\beta(v)$. More precisely, she crosses an arbitrary edge (v, w) that minimizes the perceived cost, i.e., $\tilde{c}(v, w, \beta(v)) = \tilde{c}(v, \beta(v))$, or quits if the perceived cost of v exceeds the perceived reward, i.e., $\tilde{c}(v, \beta(v)) > \beta(v) \cdot r$. We say that G is motivating with respect to a certain present-bias configuration $\beta \in B^V$ if and only if the agent does not quit on any of her walks from s to t. Note that B^V denotes the set of all present-bias configurations that map the nodes of G to the values of B.

To illustrate this setting, we revisit Alice and Bob's scenario of the previous chapter depicted in Figure 6.1. Recall that the agent of the original scenario has a fixed present-bias that is either $a = 1/2 - \varepsilon$ if she happens to be Alice or $b = 1/2 + \varepsilon$ if she is Bob. Thus, the present-bias set is given by $B = \{a, b\}$. However, instead of committing herself to either a or b, the new setting allows the agent to change between the two values depending on her current state. For instance, her present-bias configuration could be

$$\beta(v) = \begin{cases} a & \text{if } v \in \{v_A, v_{AA}, v_{AB}, v_{BB}, t\} \\ b & \text{if } v \in \{s, v_B\} \end{cases}.$$

Assuming that the reward r is sufficiently large, any agent with the above present-bias configuration walks along the same path that Bob would take, i.e., s, v_B, v_{AB}, t. However, there is a subtle difference in the agent's perception. At v_{AB} her present-bias becomes that of Alice and the perceived value of the reward drops from $b \cdot r$ to $a \cdot r$. As a result, the agent now needs more reward than Bob to stay motivated. In fact, it is not too hard to see that the minimal motivating reward with respect to β evaluates to

$$\mu(G, \beta) = \frac{\tilde{c}(v_{AB}, t, \beta(v_{AB}))}{\beta(v_{AB})} = \frac{16}{1/2 - \varepsilon};$$

a value that exceeds the minimal motivating reward of any fixed present-bias considering that $\mu(G, a) = 13/(1/2 - \varepsilon)$ and $\mu(G, b) = 16/(1/2 + \varepsilon)$.

Whenever precise information about the present-bias configuration β of an agent is available, it is easy to simulate the agent's walk through the task graph G and determine the corresponding minimal motivating reward $\mu(G, \beta)$. However, perfect knowledge of β might be very difficult if not impossible to achieve. It is therefore often more feasible to assume that the agent's present-bias varies arbitrarily over time. Whenever this is the case, we are interested in the minimal motivating reward $\mu(G, B^V)$, i.e., the smallest reward that must be placed at t to ensure that G is motivating for all present-bias configurations contained in B^V. The following proposition presents a straight forward approach to compute $\mu(G, B^V)$ in polynomial time.

Proposition 7.1.1. *The minimal motivating reward $\mu(G, B^V)$ can be computed in polynomial time with respect to the encoding length of G and B.*

Proof. Consider the following procedure, which consists of three simple steps:

(a) Construct a set F containing all edges (v, w) that minimize the agent's perceived cost at a node v for some present-bias $\beta(v) \in B$.

(b) Compute the set W containing all nodes reachable from s via edges of F.

(c) Set $\mu(G, B^V)$ to be $\max\{\tilde{c}(v, \min B)/\min B \mid v \in W\}$.

In the following proof we argue that each step requires only polynomial time and that the final result computed in step (c) is indeed the minimal motivating reward.

We begin with (a). Observe that F only contains edges (v, w) for which a present-bias $\beta(v) \in B$ exists such that $\tilde{c}(v, w, \beta(v)) = \tilde{c}(v, \beta(v))$. We already know from the proof of Theorem 6.2.1 that the collection of all values $\beta' \in B$ satisfying $\tilde{c}(v, w, \beta') = \tilde{c}(v, \beta')$ forms a closed subinterval $B_{v,w} \subseteq [0, 1]$. Moreover, the end points of this interval can be computed in polynomial time. Together with the structural assumptions on B from the previous chapter, we may assume that the statement $B_{v,w} \cap B \neq \emptyset$ can be determined efficiently for any edge (v, w). Consequently, F can be constructed in polynomial time.

We continue with (b). Given the set F the construction of W in polynomial time is trivial. Furthermore, W contains exactly those nodes an agent with variable present-bias can reach if the reward is sufficiently large. To demonstrate this, let P be a path from s to v using only edges of F. If we go through the edges (v', w') of P one by one and choose $\beta(v')$ as an element of the non-empty intersection $B_{v',w'} \cap B$, we obtain a valid present-bias configuration that may lead the agent to v for a sufficiently large reward. Conversely, no such present-bias configuration exists for nodes $v \notin W$. The reason is that all paths P to v must contain at least one edge $(v, w) \notin F$. But by definition of F the agent does not traverse (v, w) for any $\beta(v) \in B$.

We conclude with (c). As a result of (b) we know that the agent can never reach nodes outside of W. However, she can reach each $v \in W$ unless she quits at some other node of W before she gets to v. Therefore we know that G is motivating for all $\beta \in B^V$ if and only if she never quits at a node $v \in W$. Since the agent's present-bias may become $\min B$ at v, it is clear that a reward of $\max\{\tilde{c}(v, \min B)/\min B \mid v \in W\}$ or more is necessary to keep her motivated. To convince oneself that this reward also suffices for any other present-bias $\beta(v) \in B$, it is instructive to consider the following inequality

$$\begin{aligned} \frac{\tilde{c}(v, \beta(v))}{\beta(v)} &= \min\Big\{ \frac{c(v, w)}{\beta(v)} + d(w) \;\Big|\; (v, w) \in E \Big\} \leq \min\Big\{ \frac{c(v, w)}{\min B} + d(w) \;\Big|\; (v, w) \in E \Big\} \\ &= \frac{\tilde{c}(v, \min B)}{\min B} \leq \max\Big\{ \frac{\tilde{c}(v, \min B)}{\min B} \;\Big|\; v \in W \Big\}. \end{aligned}$$

We conclude that $\max\{\tilde{c}(v, \min B)/\min B \mid v \in W\}$ corresponds to the minimal motivating reward $\mu(G, B^V)$. Furthermore, the value of $\max\{\tilde{c}(v, \min B)/\min B \mid v \in W\}$ can be computed in polynomial time. This completes the proof. □

7.2 The Price of Variability

The computational considerations laid out in the previous section allow us to decide whether a given task graph G is motivating for an agent whose present-bias varies unpredictably over the present-bias set B. But of course, we do not limit ourselves to assessing a given scenario. Instead, we try to improve the agent's performance by constructing suitable incentives. Similar to the previous chapter, we are particularly interested in the conceptual loss of efficiency that emerges when comparing incentives designed to be robust with respect to variability to incentives designed for a fixed present-bias which is known a priori. We call this loss the *price of variability.*

Definition 7.2.1 (Price of Variability)**.** Given a task graph $G = (V, E, c, r)$ and a present-bias set B, the price of variability is defined as

$$\frac{\mu^*(G, B^V)}{\sup\{\mu^*(G, \beta) \mid \beta \in B\}}$$

where the minimal motivating reward $\mu^*(G, B^V)$ denotes the infimum over all rewards for which G admits an incentive of a certain type that is motivating for an agent whose present-bias varies over B.

Note that the price of variability may be used to study different types of incentives. Nevertheless, due to the favorable properties of penalty based incentives laid out throughout this work, we make them our incentive of choice once again.

Taking a closer look at the price of variability, it becomes clear that its definition bares a certain resemblance to that of the price of heterogeneity. Consequently, one may wonder whether other similarities between the two concepts exist. For instance, recall that the CopiedCostApprox algorithm can be used to establish an upper bound of 2 on the price of heterogeneity; an upper bound that is independent of the particular task graph G and present-bias set B. Clearly, this raises the question whether the same technique also yield a similar upper bound on the price of variability. The answer is yes and no. On the one hand, a slightly altered version of CopiedCostApprox can indeed be used to derive an upper bound on the price of variability. On the other hand, the resulting bound is not constant. Instead, it depends on the particular range $\tau = \max B / \min B$ of the present-bias set B.

The reason for this unsatisfying result is simple. The idea of CopiedCostApprox is to create a cost configuration h that guides the agent along some favorable minmax path P, but lets her take an occasional shortcut via a cheapest path if the maximum perceived cost along the shortcut does not become too expensive. To ensure that the shortcut does not become too expensive, h assigns a copy of the most expensive edge cost of the shortcut to its initial edge. This way the perceived cost of any edge within the shortcut is not greater than the perceived cost for entering the shortcut as long as the present-bias stays the same. However, if the present-bias is subject to change, the agent may become more biased after she has entered the shortcut. In this case she may require a higher reward to stay motivated. A straight forward way to fix this problem is to scale the assigned extra cost by the range of B. As a result we obtain the following algorithm, which we call CopiedAndScaledCostApprox:

Algorithm 7: CopiedAndScaledCostApprox

Input: Task graph G, present-bias set B
Output: Cost configuration h

$P \leftarrow$ minmax path from s to t with respect to the cost perceived by an agent with present-bias $\min B$;
$\alpha \leftarrow \max\{\tilde{c}(e, \min B) \mid e \in P\}$;
foreach $v \in V \setminus \{t\}$ **do**
  $\sigma(v) \leftarrow$ successor node of v on a cheapest path from v to t;
$T = \{(v, \sigma(v)) \mid v \in V \setminus \{t\}\}$;
foreach $(v, w) \in E$ **do**
  if $(v, w) \notin P \cup T$ **then**
    $h(e) \leftarrow (\tau + 1) \cdot \alpha / \min B + 1$;
  else if $v \in P$ **and** $w \notin P$ **then**
    $P' \leftarrow v, \sigma(v), \sigma(\sigma(v)), \ldots, t$;
    $u \leftarrow$ first node of P' different from v that is also a node of P;
    $h(v, w) \leftarrow$
      τ times the cost of a most expensive edge of P' between v and u;
  else $h(e) \leftarrow 0$;
return h;

Note that the above implementation of CopiedAndScaledCostApprox is closely related to the implementation of CopiedCostApprox presented in Section 6.4. This is particularly convenient as it allows us to extend many of the arguments originally intended to establish a bound on the price of heterogeneity to the price of variability. More precisely, it enables us to bound the price of variability by $\tau + 1$. Note that this result is taken from our work on incentive design for uncertain present-biases [4].

Theorem 7.2.1. *The price of variability is at most $\tau + 1$.*

Proof. From the analysis of CopiedCostApprox in Section 4.5 it is clear that $\alpha / \min B$ is a lower bound on $\mu^*(G, \min B)$ and thus also on $\sup\{\mu^*(G, \beta) \mid \beta \in B\}$. To establish the theorem, it therefore suffices to show that $(\tau+1)\cdot\alpha/\min B$ is an upper bound on $\mu^*(G, B^V)$. In particular, it must be demonstrated that CopiedAndScaledCostApprox returns a cost configuration h that is motivating for all $\beta \in B^V$ if the reward is set to $(\tau+1)\cdot\alpha/\min B$.

Using the same reasoning as in the proof of Theorem 4.5.1 we may assume that the cost of a cheapest path from any node v to t is at most $\tau+1$ times more expensive in G_h than in G. Consequently, the perceived cost of each edge $(v,w)\in P$, and therefore also that of each node $v \in P$, is covered by a reward of $(\tau+1)\cdot\alpha/\min B$ independent of the actual value of $\beta(v)$. The reason is that

$$\begin{aligned}\tilde{c}_h(v,w,\beta(v)) &= \beta(v)\cdot\Big(\frac{c(v,w)}{\beta(v)}+d_h(w)\Big) \leq \beta(v)\cdot\Big(\frac{c(v,w)}{\min B}+(\tau+1)\cdot d(w)\Big)\\ &\leq \beta(v)\cdot(\tau+1)\cdot\Big(\frac{c(v,w)}{\min B}+d(w)\Big) = \beta(v)\cdot(\tau+1)\cdot\frac{\tilde{c}(v,w,\min B)}{\min B}\\ &\leq \beta(v)\cdot(\tau+1)\cdot\frac{\alpha}{\min B}.\end{aligned}$$

All that remains is to show that the same reward is also sufficient if the agent is located at a node $v \notin P$. For this purpose, let v' be the last node of P that the agent visits before reaching v. Since all edges that are neither part of P nor a σ-path have a perceived cost of at least $(\tau+1)\cdot\alpha/\min B+1$, we can be sure that the agent travels from v' to v via σ-path edges only. In particular, this implies that the agent must cross the edge $(v',\sigma(v'))$ to reach v. For the same reasons that are presented in the proof of Theorem 4.5.1, we may assume that a cheapest path from $\sigma(v')$ to t with respect to h is more expensive than a cheapest path from $\sigma(v)$ to t, i.e., $d_h(\sigma(v)) \leq d_h(\sigma(v'))$. Furthermore, the inequalities $c(v,\sigma(v)) \leq h(v',\sigma(v'))/\tau$ and $h(v,\sigma(v)) = 0$ hold true by construction of h. Finally the definition of $\tau = \max B/\min B$ implies that

$$\beta(v) \geq \min B \geq \min B \cdot \frac{\beta(v')}{\max B} = \frac{\beta(v')}{\tau}.$$

Combining these observations yields

$$\begin{aligned}\tilde{c}_h(v,\sigma(v),\beta(v)) &= \beta(v)\cdot\Big(\frac{c(v,\sigma(v))}{\beta(v)}+d_h(\sigma(v))\Big) \leq \beta(v)\cdot\Big(\frac{c(v,\sigma(v))}{\beta(v')/\tau}+d_h(\sigma(v))\Big)\\ &\leq \beta(v)\Big(\frac{h(v',\sigma(v'))+c(v',\sigma(v'))}{\beta(v')}+d_h(\sigma(v'))\Big)\\ &= \frac{\beta(v)}{\beta(v')}\cdot\tilde{c}_h(v',\sigma(v'),\beta(v')).\end{aligned}$$

However, the agent has already crossed $(v',\sigma(v'))$, which means that the perceived cost

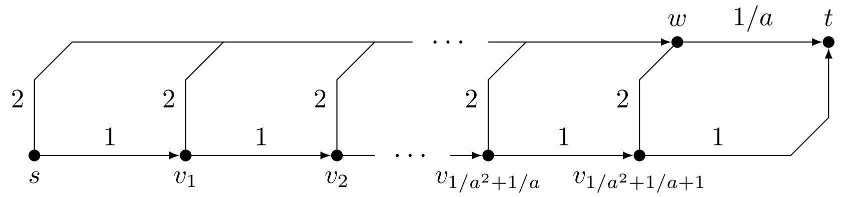

Figure 7.1: Task graph admitting a price of variability of $(1/a^2)/(2/a+2)$

of the edge $(v', \sigma(v'))$ can be at most $\tilde{c}_h(v', \sigma(v'), \beta(v')) \leq \beta(v') \cdot (\tau+1) \cdot \alpha/\min B$. Consequently, the perceived cost of the edge $(v, \sigma(v))$ is bounded by

$$\tilde{c}_h(v, \sigma(v), \beta(v)) \leq \frac{\beta(v)}{\beta(v')} \cdot \beta(v') \cdot (\tau+1) \cdot \frac{\alpha}{\min B} = \beta(v) \cdot (\tau+1) \cdot \frac{\alpha}{\min B}.$$

We conclude that a reward of $(\tau+1) \cdot \alpha/\min B$ sufficiently covers the perceived cost of v. This completes the proof. □

The fact that the just established bound on the price of variability is proportional to τ may at first seem like a weak result, especially since τ can be an arbitrarily large number. One may therefore wonder whether a different approach yields a better bound. However, it is not difficult to construct scenarios in which the price of variability is close to $\tau/2$ for arbitrary choices of τ, see e.g. the proof of the following proposition which is taken from [4]. Consequently, Theorem 7.2.1 is in fact tight up to a constant factor of 2. This result is particularly surprising as it stands in stark contrast with the price of heterogeneity.

Proposition 7.2.2. *There exists a family of task graphs and present-bias sets for which the price of variability converges to $\tau/2$ as τ goes to infinity.*

Proof. To obtain a price of variability close to $\tau/2$ we consider an agent whose present-bias varies over the set $B = \{a, 1\}$ for some $0 < a < 1/2$ such that $1/a$ is integral. Furthermore, we construct a task graph G consisting of a directed path $v_0, v_1, \ldots, v_{1/a^2+1/a+2}$ whose edges are all of cost 1. We call this the *regular path*. In addition to the regular path we introduce $1/a^2 + 1/a + 2$ *shortcuts* via a common node w. Each shortcut i with $0 \leq i \leq 1/a^2 + 1/a + 1$ consists of two edges: The first edge goes from v_i to w for a cost of 2 while the second edge goes from w to t for a cost of $1/a$. As source and target node, we choose $s = v_0$ and $t = v_{1/a^2+1/a+2}$. Figure 7.1 shows a sketch of G.

The remainder of the proof has a similar structure to that of Proposition 6.3.1. We first argue that a reward of $2/a + 2$ is sufficiently motivating for any agent with a fixed present-bias of either a or 1, implying $\sup\{\mu^*(G,a), \mu^*(G,1)\} \leq 2/a+2$. We then show that no cost configuration h can motivate an agent with a variable present-bias if the reward is less than $1/a^2$, i.e., $\mu^*(G,\{a,1\}^V) \geq 1/a^2$. As a result the price of variability must be at least $(1/a^2)/(2/a+2)$. Note that this term approaches $1/(2 \cdot a) = \tau/2$ as a goes to 0, which establishes the theorem.

To see that a reward of $2/a+2$ suffices to motivate an agent with a fixed present-bias of a, let the agent be located at an arbitrary node v_i with $i \leq 1/a^2 + 1/a + 1$. The perceived cost of (v_i, v_{i+1}) is at most $\tilde{c}(v_i, v_{i+1}, a) \leq 1 + a \cdot (2 + 1/a) = 2 + 2 \cdot a$ if she plans to take the next shortcut at v_{i+1}. In the special case of $i = 1/a^2 + 1/a + 1$ the perceived cost is exactly 1 as she can reach t directly via $(v_{1/a^2+1/a+1}, t)$. Either way a reward of $1/a \cdot (2 + 2a) = 2/a + 2$ covers the perceived cost for staying on the regular path. In contrast, taking the immediate shortcut at v_i has a perceived cost of $\tilde{c}(v_i, w, a) = 2 + a \cdot (1/a) = 3$. As we assume $a < 1/2$, the agent clearly perceives direct shortcuts to be more expensive than the regular path. Consequently, she follows the regular path from s to t for a reward of $2/a + 2$.

Next consider an unbiased agent, i.e., an agent with a fixed present-bias of 1. Clearly, this agent strictly follows a cheapest path P from s to t. Furthermore, her perceived cost along P never exceeds the total cost of P. Taking the first shortcut at s, we can bound the cost of a cheapest path from s to t by $2 + 1/a < 2/a + 2$. This implies that the unbiased agent successfully reaches t for a reward of $2/a + 2$.

It remains to show that no cost configuration h is motivating for all present-bias configurations $\beta \in \{1, a\}^V$ if the reward is less than $1/a^2$. For the sake of contradiction, assume such a cost configuration h exists. Note that h must keep the agent from visiting w. The reason is that a reward less than $1/a^2$ cannot motivate the agent to cross (w, t) should her present-bias become $\beta(w) = a$. However, to prevent the agent from taking a shortcut, h must assign a cost greater than $1/a^2 - i$ to all shortcuts i for $0 \leq i \leq 1/a^2$. To see this, consider the following induction on i:

We start the induction with $i = 1/a^2$. At v_{1/a^2} exactly $2 + 1/a$ edges remain on the regular path. Ignoring extra cost, there are two cheapest paths to t, one along the regular path and one along the current shortcut. Consequently, if the agent is momentarily unbiased, i.e., $\beta(v_{1/a^2}) = 1$, and therefore indifferent between the two cheapest path from v_{1/a^2} to t, it becomes clear that h must assign an extra cost greater than $0 = 1/a^2 - i$ to the current shortcut. This is to prevent the agent from moving to w.

For the induction step, let $i < 1/a^2$ and assume that each shortcut j with $i < j \leq 1/a^2$ has an extra cost greater than $1/a^2 - j$ assigned to it. Our goal is to argue that the extra cost of shortcut i must be greater than $1/a^2 - i$ as well. When located at v_i, exactly

$1/a^2 - i + 1/a + 2$ edges of the regular path remain until t. If the agent is currently unbiased, she perceives a cost of at least $1/a^2 - i + 1/a + 2$ for taking the regular path. Clearly she cannot reduce this cost by planning to take a shortcut j' with $j' > 1/a^2$. Should she consider a shortcut j with $i < j \leq 1/a^2$ instead, she must first traverse $(j-i)$ edges of the regular path. Together with the induction hypothesis her total perceived cost for such a plan is at least $(j-i) + 2 + 1/a + (1/a^2 - j) = 1/a^2 - i + 1/a + 2$. Consequently, her perceived cost for staying on the regular path is greater or equal to $1/a^2 - i + 1/a + 2$. Since this exceeds the cost of shortcut i by $1/a^2 - i$ or more, we know that an extra cost greater than $1/a^2 - i$ must be assigned to the current shortcut to prevent the agent from walking onto w. This concludes the induction.

We now know that all shortcuts i with $0 \leq i \leq 1/a^2$ have an extra cost greater than $1/a^2 - i$. Using the same argument as in the inductive step, it should be clear that each path from s to t has a cost of at least $1/a^2 + 1/a + 2$. Therefore, if the agent is unbiased at s, we need a reward of $1/a^2 + 1/a + 2 > 1/a$ to motivate her. However, this contradicts our initial assumption on h. □

7.3 The Approximability of Cost Configurations for a Variable Present-Bias

Apart from studying the price of variability, CopiedAndScaledCostApprox can of course also be used to construct actual cost configurations for agents with a variable present-bias. Based on the arguments presented in the previous section, it is not hard to see that CopiedAndScaledCostApprox yields a $(\tau + 1)$ approximation of the given design problem.

Proposition 7.3.1. *Given a task graph G and present-bias set $B \subset (0, 1]$,* CopiedAndScaledCostApprox *yields a cost configuration h whose minimal motivating reward $\mu(G_h, B^V)$ is less than $(\tau + 1) \cdot \mu(G_{h'}, B^V)$ for any other cost configuration h'.*

Proof. The proof of Theorem 7.2.1 argues that CopiedAndScaledCostApprox returns a cost configuration h that requires a reward of at most $(\tau + 1) \cdot \alpha / \min B$ to be motivating for any present-bias configuration B^V, i.e., $\mu(G_h, B^V) \leq (\tau + 1) \cdot \alpha / \min B$. However, by construction of the parameter α we furthermore know that any agent whose present-bias is fixed to be $\min B$ needs a reward of at least $\alpha / \min B$ to stay motivated. This holds true independent of the actual cost configuration h' assigned to G, i.e., $\mu(G_{h'}, \min B) \geq \alpha / \min B$. Consequently, we conclude that

$$\mu(G_h, B^V) \leq (\tau + 1) \cdot \frac{\alpha}{\min B} \leq (\tau + 1) \cdot \mu(G_{h'}, \min B) \leq (\tau + 1) \cdot \mu(G_{h'}, B^V). \quad \square$$

The approximation ratio of $(\tau+1)$ is of course far from the constant ration of 2 that can be achieved for fixed present-bias values. Moreover, our best hardness result so far, i.e., the result of Theorem 4.6.1, only implies that an approximation within a constant factor of 1.08192 or less is NP-hard. Clearly, there is a large gap between the approximation ratio that we can currently achieve in polynomial time and the approximation ratio which we know to be hard. In an attempt to narrow this gap at least a little bit, we present a reduction from VECTOR SCHEDULING (VS) in [4]. The goal of the reduction is to prove that any constant factor approximation of an almost optimal cost configuration that is robust with respect to variability is unlikely to be achievable in polynomial time.

Before we take a closer look at the reduction, it is of course sensible to specify an exact notion of the computational problem for which we want to prove hardness. In reminiscence of the original optimization problem MCC-OPT, and also in lack of a better name, we call the problem at hand MCCV-OPT, where the "V" stands for "variability".

Definition 7.3.1 (MCCV-OPT)**.** The problem of computing $\mu^*(G, B^V)$ for a given task graph $G = (V, E, c, r)$ and present-bias set $B \subset (0, 1]$.

Furthermore, we need to define the optimization version of the VS problem that the reduction is based on.

Definition 7.3.2 (VS-OPT)**.** The problem of finding the smallest makespan with respect to all schedules that assign ℓ jobs $q_1, \dots, q_\ell \in \mathbb{R}^d_{\geq 0}$ to m machines $M_1, \dots, M_m$, i.e., minimize $\max\{\| \sum_{q \in M_i} q\|_\infty \mid 1 \leq i \leq m\}$ over all partitions of the vectors $q_1, \dots, q_\ell$ into the sets $M_1, \dots, M_m$.

According to the work of Chekuri and Khanna [8], VS-OPT is highly unlikely to admit algorithms that achieve a constant approximation in polynomial time. More precisely, they prove that no polynomial time algorithm approximates VS-OPT within a constant factor of $\gamma > 1$, unless NP = ZPP. This holds true even if all vectors q_k are 0-1 vectors. Based on Chekuri and Khanna's result, we can deduce that MCCV-OPT does not permit a constant approximation in polynomial time either unless NP = ZPP.

Theorem 7.3.2. *There is no polynomial time algorithm that approximates MCCV-OPT within a constant factor $\gamma > 1$ unless NP = ZPP.*

Proof. Let $\mathcal{I}$ be an arbitrary instance of VS-OPT with a set of m machines and $\ell \geq 2$ jobs $q_1, \ldots, q_\ell \in \{0,1\}^d$. Our goal is to construct a MCCV-OPT instance $\mathcal{J}$ that is of polynomial size with respect to $\mathcal{I}$ and satisfies the following two properties:

(a) If $\mathcal{I}$ has a schedule with a makespan of κ, then $\mathcal{J}$ has a cost configuration that is motivating for a reward $r = \kappa \cdot \ell + \ell + 1$ independent of the particular present-bias configuration $\beta \in B^v$.

(b) If $\mathcal{J}$ has a cost configuration that is motivating for a reward r independent of the present-bias configuration $\beta \in B^v$, then $\mathcal{I}$ has a schedule with a makespan of at most $\kappa = 2 \cdot r/\ell$.

Consequently, any polynomial time approximation algorithm for MCCV-OPT that has a constant approximation ratio ϱ can be applied to $\mathcal{J}$ to obtain a bound of

$$2 \cdot \frac{\varrho \cdot (\kappa \cdot \ell + \ell + 1)}{\ell} = 2 \cdot \varrho \cdot \kappa + 2 \cdot \varrho + 2 \cdot \frac{\varrho}{\ell}$$

on the makespan. However, this solution approximates $\mathcal{I}$ within a factor of $2 \cdot \varrho + \mathcal{O}(1)$. According to Chekuri and Khanna [8], this is not possible unless NP = ZPP.

As our first step we construct the MCCV-OPT instance $\mathcal{J}$. For this purpose, we need to specify the present-bias set B and a task graph G. We begin with B. Proposition 7.3.1 shows that MCCV-OPT has a $\tau + 1$ approximation algorithm. Consequently, the range τ of B must depend on the size of $\mathcal{I}$. To keep things simple, we set $B = \{1/\ell^2, 1/2\}$.

We continue with the construction of G. For each machine i and dimension j, let G contain a subcomponent $H_{i,j}$. The structure of these components, which we call *columns*, is illustrated in Figure 7.2. Observe that each column $H_{i,j}$ consists of ℓ *layers*. Moreover, each layer k consists of a so-called *regular path* leading from the node $v_{i,j,k}$ to the node $w_{i,j,k}$ via ℓ^4 edges of cost $1/\ell^2$. The total cost of this path therefore sums up to ℓ^2. In addition to the regular path, each node of the layer k has a *shortcut* to the next layer $k+1$, or to t if $k = \ell$. For a convenient notation let $t = v_{i,j,\ell+1}$. We distinguish between two types of shortcuts: First, there is a shortcut from each node $v_{i,j,k}$ to $v_{i,j,k+1}$ via an intermediate node $v'_{i,j,k}$. The initial edge of this shortcut is free of charge while the second has a cost of 1. Secondly, all remaining nodes of the layer k have a direct shortcut to $v_{i,j,k+1}$ for a cost of ℓ.

To combine the individual columns into the final task graph G, we introduce $\ell - 1$ additional nodes $u_1, \ldots, u_{\ell-1}$ and set $s = u_0$ and $t = u_\ell$. As Figure 7.3 illustrates for a small sample instance of $\mathcal{I}$, each pair of consecutive nodes u_{k-1} and u_k is connected via a so-called *extended regular paths*, one such path $P_{i,k}$ for each machine i. The path

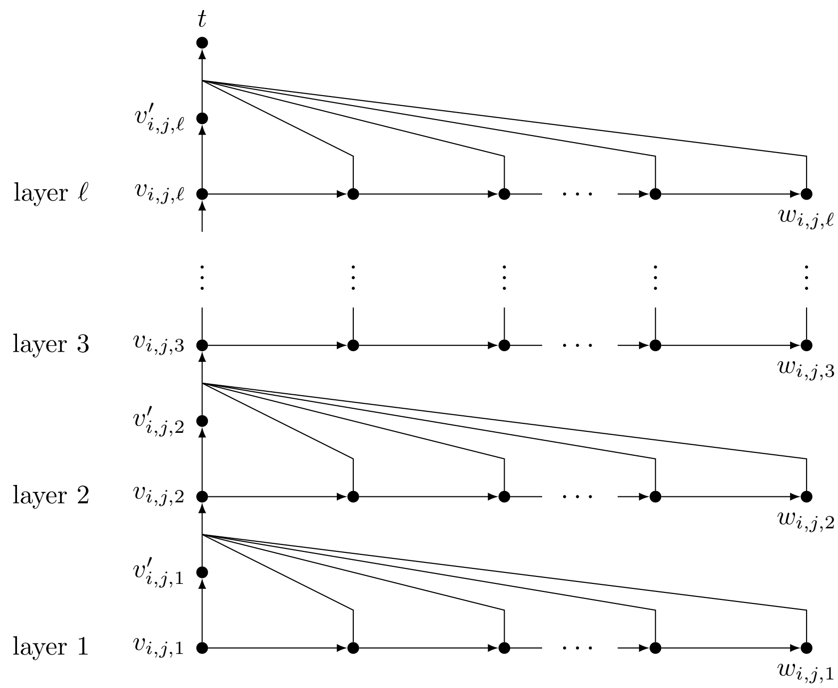

Figure 7.2: A column $H_{i,j}$ of the task graph G

$P_{i,k}$ crosses the k-th layer of all the columns $H_{i,j}$ for which the job q_k has a cost of 1 in dimension j, i.e., $(q_k)_j = 1$. More formally, $P_{i,k}$ has the following structure: Let j be an arbitrary dimension for which $(q_k)_j = 1$. Without loss of generality, we may assume that at least one such dimension exists. Otherwise, q_k could be assigned to any machine without affecting the makespan and therefore it would be irrelevant to the schedule. If j is the dimension of the lowest index satisfying $(q_k)_j = 1$, an edge of cost $1/\ell^2$ is drawn from u_{k-1} to $v_{i,j,k}$. Similarly, an edge of cost $1/\ell^2$ is drawn from $w_{i,j,k}$ to u_k if j is the dimension of highest index for which $(q_k)_j = 1$. For all intermediate dimensions j satisfying $(q_k)_j = 1$ an edge of cost $1/\ell^2$ is drawn from $w_{i,j,k}$ to a distinct intermediate node $u_{i,j,k}$ and another edge of the same cost is drawn from $u_{i,j,k}$ to $v_{i,j',k}$, where j' is the dimension of the next higher index satisfying $(q_k)_{j'} = 1$. Note that the endpoints of the extended regular path appear twice in Figure 7.3. Keeping this in mind, it is easy to see that the agent can travel from s to t along a concatenation of extended paths and that each such walk corresponds to an assignment of jobs to machines.

To complete G, we introduce one more shortcut type that connects all the nodes $u_k \neq t$ and $u_{i,j,k}$ to t via a single edge of cost ℓ. Note that these shortcuts are not depicted in Figure 7.3 for the sake of a clear representation. Note that the resulting task graph G is

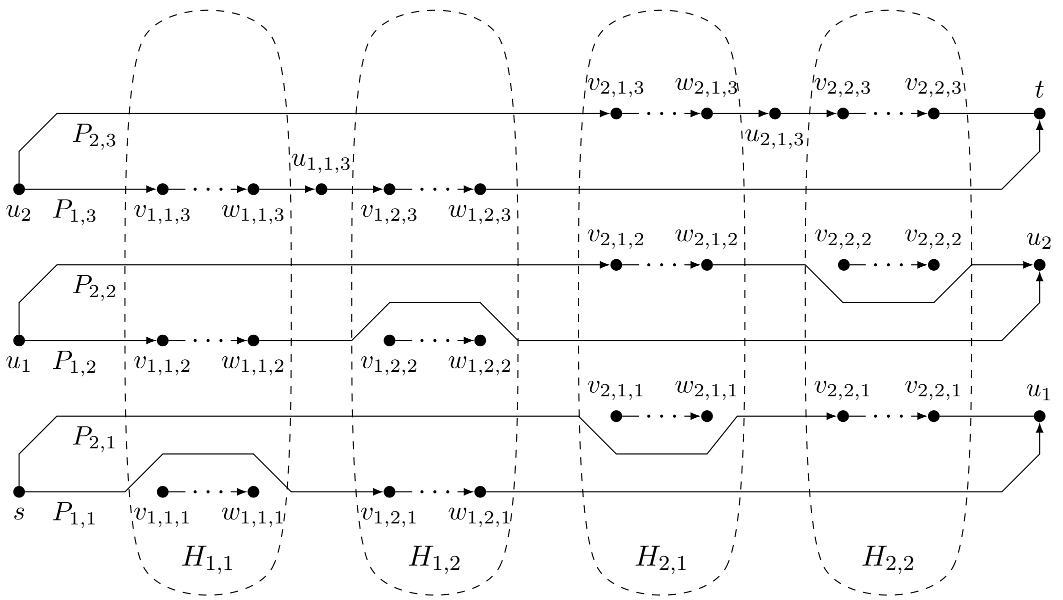

Figure 7.3: Reduction from a VS-OPT instance with $m = 2$ machines and $\ell = 3$ jobs: $q_1 = (0, 1)^T, q_2 = (1, 0)^T, q_3 = (1, 1)^T$

acyclic and can be constructed in polynomial time with respect to $\mathcal{I}$. It should also be mentioned that some columns of G might contain nodes which are not reachable from s. For instance, there is no path from s to the first layer of $H_{1,1}$ in the task graph sketched in Figure 7.3. However, because these nodes do not affect the agent's behavior, we may ignore them in our analysis.

We proceed with the proof of statement (a). For this purpose, assume that $\mathcal{I}$ has a schedule $M_1, \ldots, M_m$ with makespan κ. Our goal is to show that $\mathcal{J}$ admits a cost configuration h that is motivating for a reward of $\kappa \cdot \ell + \ell + 1$ for any present-bias configuration $\beta \in B^V$. A simple instance of h satisfying this property can be constructed as follows: Given an arbitrary job q_k, let i be the machine it is scheduled on, i.e., $q_k \in M_i$. In this case, h assigns an extra cost of $\kappa \cdot \ell + \ell + 2$ to the initial edge of all extended regular paths $P_{i',k}$ with $i' \neq i$. Furthermore, h assigns a cost of ℓ to the initial edge of all shortcuts that start at a node $v_{i,j,k}$ for which q_k has a cost of 1 in dimension j, i.e., $(q_k)_j = 1$. The idea behind this construction is to keep the agent from visiting extended regular paths $P_{i',k}$ that do not match the given schedule, i.e., $q_k \notin M_{i'}$.

It remains to give a formal argument for why h is motivating. To see this, assume that the agent is located at a node $v \neq t$ of an extended regular path $P_{i,k}$ with $q_k \in M_i$. Furthermore, let $\beta(v) \in B$ be the current present-bias. In the following we use a case distinction on the type of v to show that the agent has a strict preference for staying on $P_{i,k}$ or taking a single edge shortcut to t. In both cases, the perceived cost $\tilde{c}(v, \beta(v))$ does not exceed the perceived reward $\beta(v) \cdot (\kappa \cdot \ell + \ell + 1)$. Thus, applying this argument repeatedly until the agent reaches t yields the desired result.

We begin with $v = u_{k-1}$. At any such node the agent has three options: The first one is to enter an extended regular path $P_{i',k}$ for which $i' \neq i$. However, because h assigns an extra cost of $\kappa \cdot \ell + \ell + 2$ to the initial edge of $P_{i',k}$, this clearly cannot be motivating. The agent's second option is to enter $P_{i,k}$. In this case, let $(u_{k-1}, v_{i,j,k})$ be the first edge of $P_{i,k}$ and define $P' = u_{k-1}, v_{i,j,k}, v'_{i,j,k}, v_{i,j,k+1}, v'_{i,j,k+1}, \dots, t$ as the path from u_{k-1} to t that climbs the layers of $H_{i,j}$ along the double edge shortcuts. Clearly, P' consists of at most ℓ shortcuts, each of which has an edge of cost 1. Furthermore, at most κ shortcuts of P' charge an extra cost of ℓ. This holds true by the assumption on the makespan of the schedule. Together with the fact that $c(u_{k-1}, v_{i,j,k}) = 1/\ell^2$ and $\beta(u_{k-1}) \leq 1/\ell^2$, we conclude that

$$\begin{aligned}\tilde{c}_h\big(P', \beta(u_{k-1})\big) &\leq \frac{1}{\ell^2} + \beta(u_{k-1}) \cdot (\ell + \kappa \cdot \ell) \leq \beta(u_{k-1}) + \beta(u_{k-1}) \cdot (\ell + \kappa \cdot \ell) \\ &= \beta(u_{k-1}) \cdot (\kappa \cdot \ell + \ell + 1).\end{aligned}$$

This means that the agent is motivated to enter $P_{i,k}$. Moreover, if she does not enter $P_{i,k}$, she must have chosen her only remaining option, which is to take the direct shortcut from u_{k-1} to t. Either way our hypothesis holds true.

We continue with $v = u_{i,j,k}$. In contrast to before, the agent has only two options at such a node: either she stays on $P_{i,k}$ or she takes the direct shortcut to t. As long as one of these options is motivating, it does not matter which one she chooses. However, the same argument presented in the previous paragraph to show that the agent is motivated to enter $P_{i,k}$ at u_{k-1} also proves that she is motivated to stay on $P_{i,k}$ at $u_{i,j,k}$.

Next consider the case that v is a node on the k-th layer of a column $H_{i,j}$ such that v is different from $w_{i,j,k}$. When located at v, the agent has two options: either she takes the immediate shortcut at v or she crosses a regular edge and stays on $P_{i,k}$. The perceived cost of the first option is at least $\ell + \beta(v) \cdot d_h(v_{i,j,k+1})$. Conversely, the perceived cost of the second option is at most $1/\ell^2 + \beta(v) \cdot (\ell + d_h(v_{i,j,k+1}))$ if she plans to take a shortcut immediately after crossing the regular edge. Consequently, she prefers to stay on $P_{i,k}$ whenever $\ell > 1/\ell^2 + \beta(v) \cdot \ell$. Rearranging this term to $1 - 1/\ell^3 > \beta(v)$ and recalling that $\ell \geq 2$ and $\beta(v) \leq 1/2$ immediately show that the inequality is satisfied. But is the agent also motivated to stay on $P_{i,k}$? Reusing the argument from the previous two paragraphs immediately shows that she is motivated.

Finally, we consider the case $v = w_{i,j,k}$. The agent's options at such a node are twofold:

either she takes the shortcut at $w_{i,j,k}$ or she moves to some node u across a regular edge of $P_{i,k}$. Considering that the shortcut at $w_{i,j,k}$ is of cost ℓ, it must have a perceived cost of at least ℓ. In contrast, the perceived cost of moving to u is at most $1/\ell^2 + \beta(w_{i,j,k}) \cdot \ell$ if the agent plans to take the shortcut to t immediately after. As a result, she prefers to stay on $P_{i,k}$ whenever $\ell > 1/\ell^2 + \beta(w_{i,j,k}) \cdot \ell$. But from the previous paragraph we already know that this inequality is satisfied. Furthermore, $1/\ell^2 \leq \beta(w_{i,j,k})$ implies that staying on $P_{i,k}$ is also motivating because

$$\begin{aligned}\tilde{c}_h\big(w_{i,j,k}, u, \beta(w_{i,j,k})\big) &\leq \frac{1}{\ell^2} + \beta(w_{i,j,k}) \cdot \ell \leq \beta(w_{i,j,k}) + \beta(w_{i,j,k}) \cdot \ell \\ &\leq \beta(w_{i,j,k}) \cdot (\kappa \cdot \ell + \ell + 1).\end{aligned}$$

As the last part of the proof, we consider statement (b). For this purpose assume that $\mathcal{J}$ has a cost configuration h that is motivating for a reward of r for any present-bias $\beta(v) \in B^V$. Our goal is to schedule the jobs of $\mathcal{I}$ in such a way that the makespan is at most $2 \cdot r/\ell$. Since any schedule has a makespan less or equal to ℓ, we focus on the case $r < \ell^2/2$. In particular, this means that the perceived reward becomes strictly less than $1/2$ for a present-bias of $1/\ell^2$. However, because the intermediate node $v'_{i,j,k}$ of a double edge shortcut has a perceived cost of 1, this implies that the agent must never take any such shortcut. Otherwise, she would lose motivation if her present-bias becomes $1/\ell^2$ at $v'_{i,j,k}$. Clearly, the agent is also not interested in taking a single edge shortcut for a present-bias of $1/\ell^2$. Consequently, if the present-bias is fixed to $\beta(v) = 1/\ell^2$ at all the nodes v of G_h, the agent constructs a path P from s to t containing no shortcuts. As a result, P can be partitioned into a sequence of extended regular paths $P_{i,k}$. Assigning job q_k to machine i if $P_{i,k}$ is contained in P therefor yields a feasible schedule $M_1, \ldots, M_m$.

We proceed to argue that the makespan of $M_1, \ldots, M_m$ is at most $2 \cdot r/\ell$. As a first step, it is helpful to observe that h needs to assign an extra cost of $\ell/2 - 1$ or more to all shortcuts starting at a node $v_{i,j,k}$ of P. To verify this, assume that the agent is located at $v_{i,j,k}$. This scenario may occur whenever $\beta(v) = 1/\ell^2$ for all nodes v of P that come before $v_{i,j,k}$. Furthermore, assume that $\beta(v_{i,j,k}) = 1/2$. As argued in the previous paragraph, the agent must not take the shortcut at $v_{i,j,k}$. Instead she needs to stay on $P_{i,k}$. Let P' denote her planned path at $v_{i,j,k}$. We distinguish between two possible scenarios. First, P' might follow $P_{i,k}$ until it exits the current column $H_{i,j}$. In this case, the first $\ell^4 + 1$ edges of P' are all of cost $1/\ell^2$ and the perceived cost of P' is at least

$$\tilde{c}_h\big(P', \beta(v_{i,j,k})\big) \geq \frac{1}{\ell^2} + \beta(v_{i,j,k}) \cdot \left(\ell^4 \cdot \frac{1}{\ell^2}\right) = \frac{1}{\ell^2} + \frac{\ell^2}{2}.$$

Keeping in mind that $r < \ell^2/2$, this cannot be motivating. Secondly, P' might contain a shortcut to the next layer of $H_{i,j}$. Recall that this shortcut cannot be the shortcut at $v_{i,j,k}$. But even if we assume that the agent takes the very next shortcut, the perceived cost of P' is at least

$$\tilde{c}_h\big(P', \beta(v_{i,j,k})\big) \geq \frac{1}{\ell^2} + \beta(v_{i,j,k}) \cdot \big(\ell + d_h(v_{i,j,k+1})\big).$$

In contrast, the perceived cost of the shortcut at $v_{i,j,k}$ is only $\beta(v_{i,j,k}) \cdot (1 + d_h(v_{i,j,k+1}))$ plus the extra cost assigned to $(v_{i,j,k}, v'_{i,j,k})$ and $(v'_{i,j,k}, v_{i,j,k+1})$. Because the agent must not enter the shortcut at $v_{i,j,k}$, it follows that this extra cost is greater than

$$1/\ell^2 + \beta(v_{i,j,k}) \cdot (\ell - 1) > \beta(v_{i,j,k}) \cdot (\ell - 1) = \frac{\ell - 1}{2} > \frac{\ell}{2} - 1.$$

Next, consider the workload $\kappa_{i,j}$ on an arbitrary machine i in an arbitrary dimension j. Assuming that $\kappa_{i,j} > 0$, let q_k be the job of the lowest index scheduled on i such that $(q_k)_i = 1$. Furthermore, assume that the agent is located at $v_{i,j,k}$ and that her current present-bias is $\beta(v_{i,j,k}) = 1/\ell^2$. Because $v_{i,j,k}$ is located on P, this assumption is justified. We continue with a case distinction on the path P' that the agent plans to take when located at $v_{i,j,k}$. In general, P' can either exit column $H_{i,j}$ via a regular edge or it is completely contained in $H_{i,j}$ and climbs to t via internal shortcuts. If P' exits $H_{i,j}$, it must contain at least $\ell^4 + 1$ regular edges, each of cost $1/\ell^2$. Clearly, the perceived cost of such a P' is at least

$$\tilde{c}_h\big(P', \beta(v_{i,j,k})\big) \geq \frac{1}{\ell^2} + \beta(v_{i,j,k}) \cdot \left(\ell^4 \cdot \frac{1}{\ell^2}\right) > \beta(v_{i,j,k}) \cdot \left(\ell^4 \cdot \frac{1}{\ell^2}\right) = \frac{1}{\ell^2} \cdot \left(\ell^4 \cdot \frac{1}{\ell^2}\right) = 1$$

and cannot be motivating for the perceived reward, which is less than $1/2$. However, this means that P' climbs to the top of column $H_{i,j}$ via shortcuts. According to the observation from the previous paragraph, this means that P' consists of $\kappa_{i,j}$ or more shortcuts that have a cost of at least $\ell/2$ with respect to h. As a result, the perceived cost of P' is at least

$$\tilde{c}_h\big(P', \beta(v_{i,j,k})\big) \geq \beta(v_{i,j,k}) \cdot \kappa_{i,j} \cdot \frac{\ell}{2} = \frac{\kappa_{i,j}}{2 \cdot \ell}.$$

To make sure that this cost does not exceed the perceived reward r/ℓ^2, makespan κ can be at most $2 \cdot r/\ell$. □

7.4 Occasionally Unbiased Agents

Although MCCV-OPT most likely does not admit an algorithm that achieves a constant approximation factor in polynomial time, we would like to end this chapter with an interesting special case that can actually be solved optimally in polynomial time. The only constraint we need to impose on MCCV-OPT to obtain this positive result is that the present-bias set B contains the value 1. The interpretation of this setting in real-life is that the agent may become temporarily unbiased at any given point in time. For this reason we call such an agent *occasionally unbiased*.

The intuitive reason why incentives are easier to construct for an occasionally unbiased agent than for a general agent is that the former may base her decisions on objectively optimal plans from time to time, i.e., there is always a chance that she follows a cheapest path to t. This observation is crucial as it allows us to reduce the decision version of MCCV-OPT, which we call MCCV, to a simple structural analysis of the task graph G.

Definition 7.4.1 (MCCV)**.** The problem of deciding if a task graph $G = (V, E, c, r)$ admits a cost configuration h that is motivating for any present-bias configuration $\beta \in B^V$.

More precisely, it turns out that MCCV is feasible if and only if the task graph G admits a node set W that contains s and satisfies the following two properties:

(a) Each node $v \in W$ can be associated with a path P to t that only uses nodes of W.

(b) The perceived cost of P with respect to the present-bias value $\min B$ is covered by the promised reward, i.e. $\tilde{c}(P, \min B) \leq \min B \cdot r$.

Because of property (b) we also refer to W as a *threshold set*.

Proposition 7.4.1. *Assuming that* $1 \in B$*, MCCV has a solution if and only if* G *admits a threshold set* W *such that* $s \in W$*.*

Proof. To prove the proposition, we first assume that G admits a threshold set W that contains s. Our goal is to construct a cost configuration h that is motivating for all $\beta \in B^V$. For this purpose, let h assign an extra cost of $h(v, w) = r + 1$ to all edges (v, w) that leave W, i.e., all edges for which $v \in W$ but $w \notin W$. All other edges e of G are free of charge, i.e., $h(e) = 0$. Note that the resulting cost configuration h discourages the agent from ever leaving W. The reason is that the perceived extra cost for crossing an edge (v, w) that exits W alone is already $r + 1$. Clearly, this is not covered by the perceived reward of $\beta(v) \cdot r$. To conclude the first part of the proof, it remains to show that no matter at which node $v \in W \setminus \{t\}$ the agent is located, she always finds a successor node $w \in W$ that she is motivated to visit next.

In order to prove the above statement, let v be an arbitrary node of W different from t. Because W is a threshold set, a path $P = v, w, \dots, t$ must exist that consists exclusively of nodes from W. Furthermore, the perceived cost of P with respect to a present-bias of $\min B$ can be at most $\min B \cdot r$. The perceived cost of P must therefore be less than $\beta(v) \cdot r$ for any other present-bias $\beta(v) \in B$ too since

$$\begin{aligned}\tilde{c}(P, \beta(v)) &\leq c(v, w) + \beta(v) \cdot \sum_{e \in P \setminus \{v\}} c(e) \leq \frac{\beta(v)}{\min B} \cdot \Big(c(v, w) + \min B \cdot \sum_{e \in P \setminus \{v\}} c(e) \Big) \\ &= \frac{\beta(v)}{\min B} \cdot \tilde{c}(P, \min B) \leq \beta(v) \cdot r.\end{aligned}$$

Considering that $\tilde{c}(P, \beta(v))$ is an upper bound on $\tilde{c}_h(v, w, \beta(v))$, we know that the agent is motivated to cross the edge (v, w).

Next, assume that G admits a cost configuration h that is motivating for all present-bias configurations $\beta \in B^V$. Moreover, let W' be the set of all nodes that the agent might visit on her walk from s to t. To complete the proof, we argue that W' is a critical node set, i.e., W' satisfies the following two properties:

(a) Each node $v \in W'$ can be associated with a path P to t that only uses nodes of W'.

(b) The perceived cost of P with respect to the present-bias value $\min B$ is at most $\tilde{c}(P, \min B) \leq \min B \cdot r$.

For this purpose, let v be some node of W' and choose w as the immediate successor of v that the agent visits should her present-bias be $\beta(v) = \min B$. Furthermore, let P' be a cheapest path from w to t with respect to h. By adding (v, w) to the initial node of P', we obtain a suitable path P that satisfies (a) and (b).

We first show that P satisfies (b). Recall that by definition of w the agent is motivated to cross (v, w) if her present-bias is $\min B$, i.e., $\tilde{c}_h(v, w, \min B) \leq \min B \cdot r$. The fact that P' is a cheapest path from w to t with respect to h immediately implies that the perceived cost of P with respect to $\min B$ is at most

$$\tilde{c}(P, \min B) \leq \tilde{c}_h(P, \min B) = \tilde{c}_h(v, w, \min B) \leq \min B \cdot r.$$

We continue with (a). Assuming the agent is located at v, consider a present-bias configuration β that assigns a value of $\beta(v) = \min B$ to v and $\beta(v') = 1$ to all nodes $v' \neq v$ located on paths from v to t. Because the value 1 is contained in B, it is possible to construct such a present-bias configuration. By choice of (v, w) we also know that the agent may actually cross (v, w) and end up at w. If she does, she then proceeds to follow a cheapest path from w to t with respect to h. Since P' is such a cheapest path, we conclude that the agent may visit any node of P. But this implies that all nodes of P are contained in W'. □

As a result of the above theorem we can easily reduce the problem of constructing a cost configuration that is motivating for an occasionally unbiased agent to the problem finding a threshold set W in G. After all, the proof of the theorem suggests that W can be converted to a motivating cost configuration by assigning an extra cost of $h(e) = r + 1$ to all edges e that exit the set W. Alternatively, it is also possible to simply delete e. The latter observation is quite remarkable as it implies that penalty fees and prohibitions are equally powerful in the special case of occasionally unbiased agents.

The only missing piece in our attempt to construct an optimal cost configuration for occasionally unbiased agents is an algorithm to compute a suitable threshold set W. Clearly, it is not too difficult to come up with an implementation that solves this problem in polynomial time. To give just one example, consider the following algorithm, which we call ThresholdSet:

Algorithm 8: ThresholdSet

Input: Task graph G, present-bias set B
Output: Threshold set W

$\delta(t) \leftarrow 0$;
foreach $v \in V \setminus \{t\}$ **in reverse topological order do**
  $U \leftarrow \{w \mid c(v, w) + \min B \cdot \delta(w) \leq \min B \cdot r\}$;
  if $U = \emptyset$ **then**
    $\delta(v) \leftarrow \infty$;
  else
    $\delta(v) \leftarrow \min\{c(v, w) + \delta(w) \mid w \in U\}$;
return $\{v \mid \delta(v) < \infty\}$

It is easy to see that ThresholdSet runs in polynomial time with respect to the encoding length of G and B. Furthermore, it should be clear that the node set W returned by ThresholdSet satisfies properties (a) and (b) of a threshold set. Together with the observation that the algorithm adds nodes greedily and in a reverse topological order to W, we may conclude that W is not only a threshold set, but a maximum threshold set in the sense that no other threshold set contains nodes which are not also contained in W. In particular this implies that W contains s whenever there exists some threshold set that contains s. As a result of Proposition 7.4.1, ThresholdSet therefore allows us to construct a motivating reward configuration for occasionally unbiased agents in polynomial time whenever such an incentive is possible. In particular, this implies that MCCV can be decided in polynomial time for occasionally unbiased agents.

Proposition 7.4.2. *MCCV can be solved in polynomial time whenever* $1 \in B$.

Proof. According to Proposition 7.4.1 and our analysis of ThresholdSet, it suffices to run ThresholdSet on the given instance of G and B and check whether the returned node set W contains s. This requires polynomial time at most. □

8 Conclusions

Drawing on Kleinberg and Oren's graphical model [16] we have designed and analyzed behavioral economic incentives from an algorithmic perspective. The common idea behind these incentives was to help present-biased people reach predefined long-term goals in a way that is cost efficient for the designer of the incentive. In the following we present a brief summary of the main results before we conclude with some suggestions for future research that is particularly interesting in our opinion.

8.1 Summary of Results

The first part of this work focused on three different types of frequently used incentives that are based on prohibition, penalty fees and intermediate rewards. The reason why we were interested in these incentives was twofold. Our first goal was to quantify and compare the conceptual power of different types of incentives. For this purpose we considered the minimum motivating rewards $\mu^*_{\text{prb}}(G,\beta)$, $\mu^*_{\text{pnl}}(G,\beta)$ and $\mu^*_{\text{rwd}}(G,\beta)$ with respect to the three incentives as well as the minimal motivating reward $\mu(G,\beta)$ that must be invested if no incentives are admissible. Based on these values, we showed that

$$1 \leq \frac{\mu^*(G,\beta)}{\mu^*_{\text{prb}}(G,\beta)} \leq \beta^{-n+2}, 1 \leq \frac{\mu^*_{\text{prb}}(G,\beta)}{\mu^*_{\text{pnl}}(G,\beta)} \leq \frac{1}{\beta} \text{ and } 0 \leq \frac{\mu^*_{\text{rwd}}(G,\beta)}{\mu^*_{\text{pnl}}(G,\beta)} \leq \beta^{-n+2}$$

for arbitrary choices of β and G. These results are tight. Furthermore, they imply that penalty fees are more efficient than prohibition and prohibition is more efficient than no incentive at all. Only intermediate rewards resist a classification in this hierarchy as they are sometimes more and sometimes less powerful than prohibition based incentives.

Our second goal was to design actual instances of these incentives whose minimal motivating reward is as close to the theoretical optimum as possible. For all three types of these incentives, we showed that this computational problem is NP-hard to solve optimally. Furthermore, we showed that the design of an optimal prohibition based incentive remains NP-hard even if the path of the agent with respect to an optimal solution is known in advance. This result is contrasted by penalty based incentives, for which we proved that similar information makes the design of almost optimal penalty fees feasible in polynomial time. Since optimal designs are generally NP-hard for all three types of incentives, we continued with a study of approximation techniques. In the case of prohibition based incentives, we devised two simple algorithms with approximation ratios of $1/\beta$ and $1 + \beta \cdot n$ respectively. Combining these two algorithms yields a straight forward $(1 + \sqrt{n})$-approximation. To complement this result, we argued that any approximation better than $\sqrt{n}/3$ is NP-hard. Next, we turned our attention to penalty based incentives and presented a 2-approximation algorithm. Moreover, we showed that any approximation within a constant factor of 1.08192 is NP-hard. Finally, we argued that reward based incentives on their own do not admit a polynomial time approximation within any bounded ratio unless P = NP. However, when combined with prohibition and penalty fees it is in fact possible to construct optimal incentives in polynomial time.

The second part of this work was concerned with the effects of heterogeneous and variable present-biases on the design of efficient incentives. We first focused on a setting in which incentives have to be designed in a way that motivates each member of a population equally. The challenge was that people do not necessarily share the same present-bias. Instead, each person has her own present-bias, which is drawn from a larger set $B \subset (0, 1]$. The goal was to design an incentive that is motivating for each present-bias of the set B. Based on these underlying assumptions, we considered the conceptual cost of addressing the entire population with the same incentive rather than individualized incentives. We called this cost the price of heterogeneity and defined it as the ratio between the minimum motivating reward $\mu^*(G, B)$ required to motivate each person with the same incentive and the minimum motivating reward $\max\{\mu^*(G, \beta) \mid \beta \in B\}$ required to motivate the most costly person with an individualized incentive. Using penalty fees as our incentive of choice, we constructed a family of scenarios in which the price of heterogeneity converges to 1.1. This proves that a true loss of efficiency is sometimes unavoidable. However, we were also able to prove that the loss of efficiency cannot be much greater than this. More precisely, we argued that the price of heterogeneity is at most 2 for penalty based incentives. To obtain this result, we adjusted the 2-approximation algorithm from the first part of this work and, in the process, devised a 2-approximation algorithm for the heterogeneous setting.

Inspired by the work of Gravin et al. [13] we then proceeded to a slightly modified setting in which the present-bias of a person is drawn from a set B, but varies arbitrarily over time. In this case, it is important that incentives are motivating independent of the changes of the present-bias. As in the case of heterogeneity, we quantified the conceptual loss of efficiency arising in such a setting in terms of the ratio between the minimum motivating reward $\mu^*(G, B^V)$ required if the present-bias is variable and the minimum motivating reward $\max\{\mu^*(G, \beta) \mid \beta \in B\}$ required if the present-bias is fixed to a worst case instance. We called this ratio the price of variability. In contrast to the price of heterogeneity, which in the case of penalty fees is bounded by a constant, we were able to construct a family of scenarios for which the price of variability converges to $\tau/2$ as the range $\tau = \max B / \min B$ of B goes to infinity. This implies that the price of variability for penalty fees generally depends on τ. To complement this result, we adjusted the 2-approximation of the first part of this work once more and obtained an upper bound of $\tau + 1$ on the price of variability as well as a matching $(\tau + 1)$-approximation algorithm. The gap between this approximation ratio and the NP-hard 1.08192-approximation ratio following from our previous work is of course significant. To narrow this gap, we argued that a constant approximation is impossible in polynomial time unless NP = ZPP. Nevertheless, we identified a curious special case in which the construction of optimal penalty fees is in fact feasible in polynomial time. The only requirement is that the value 1 is contained in B. In other words, it must be assumed that people occasionally lose their present-bias.

8.2 Future Work

Kleinberg and Oren's graphical model [16] has opened a fascinating research area at the intersection between computer science and behavioral economics with many directions for future work. In the light of our results, we would like to point out three specific directions: First, there is the problem of improving the approximability results. So far we have devised polynomial time algorithms for designing optimal prohibition based and penalty based incentives with approximation ratios of $1 + \sqrt{n}$ and 2 respectively. However, the corresponding NP-hardness results only apply to approximation ratios of $\sqrt{n}/3$ and 1.08192. An even wider gap still remains between the $(\tau + 1)$-approximation of optimal penalty fees in a setting with a variable present-bias and the impossibility of a constant factor approximation unless NP = ZPP. It would be interesting to see these upper and lower bounds tightened, either by devising new approximation algorithms or by establishing stronger hardness results.

Secondly, a straight forward extension of our work is the design and analysis of other types of incentives in the graphical model. Our choice of prohibitions, penalty fees and intermediate rewards was primarily motivated by the generality of these three types of incentives. However, more specific applications may justify the study of other types of incentives in the graphical model. For instance, Kleinberg and Oren briefly consider a setting in which a single task can be stretched over a period of time by dividing it into a sequence of smaller tasks [16].

Finally, it would be worth the effort to consider more general versions of the graphical model. Recall that at the beginning of this work, we made some simplifying assumptions. The first assumption was that people have a discount rate of $\rho = 0$ at any point that is $t > 0$ time units in the future. However, the standard notion of quasi-hyperbolic discounting permits arbitrary non-negative discount rates. Thus, the question arises how a discount rate of $\rho > 0$ affects the graphical model and the design of incentives. Another assumption was that people behave naively in the sense that they are unaware of their present-biased perception. Kleinberg, Oren and Raghavan address this issue by extending the graphical model to agents who are fully or at least partially aware of their present-bias [17]. Part of their work is focused on conceptual properties of prohibition based incentives in this new model. It would be interesting to extend their work to algorithmic consideration and introduce other incentives into this setting. The last assumption was that no other cognitive bias except for the present-bias affects a person's behavior. However, designing robust incentives may sometimes require to consider other aspects of human behavior as well. In another extension of the graphical model, Kleinberg, Oren and Raghavan examine agents who suffer from different cognitive biases [18]. More precisely, they look at agents whose decisions are simultaneously influenced by a present-bias and a sunk-cost-bias. Designing incentives for this setting or other settings with multiple cognitive biases may be a promising research direction.

Bibliography

[1] George A Akerlof. Procrastination and obedience. *The American Economic Review*, 81(2):1–19, 1991.

[2] Susanne Albers and Dennis Kraft. Motivating time-inconsistent agents: A computational approach. In *Proceedings of the 12th International Conference on Web and Internet Economics*, pages 309–323. Springer, 2016.

[3] Susanne Albers and Dennis Kraft. On the value of penalties in time-inconsistent planning. In *Proceedings of the 44th International Colloquium on Automata, Languages and Programming*, pages 10:1–10:12. Schloss Dagstuhl Leibniz-Zentrum für Informatik, 2017.

[4] Susanne Albers and Dennis Kraft. The price of uncertainty in present-biased planning. In *Proceedings of the 13th International Conference on Web and Internet Economics*, pages 325–339. Springer, 2017.

[5] Dan Ariely and Klaus Wertenbroch. Procrastination, deadlines, and performance: Self-control by precommitment. *Psychological science*, 13(3):219–224, 2002.

[6] Robert J Aumann. Rationality and bounded rationality. *Games and Economic Behavior*, 21(1-2):2–14, 1997.

[7] Gharad Bryan, Dean Karlan, and Scott Nelson. Commitment devices. *Annual Review of Economics*, 2(1):671–698, 2010.

[8] Chandra Chekuri and Sanjeev Khanna. On multidimensional packing problems. *SIAM Journal on Computing*, 33(4):837–851, 2004.

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

[10] Herbert Edelsbrunner. *Algorithms in Combinatorial Geometry*. Springer, 1987.

[11] Shane Frederick, George Loewenstein, and Ted O'donoghue. Time discounting and time preference: A critical review. *Journal of Economic Literature*, 40(2):351–401, 2002.

[12] Michael R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., 1990.

[13] Nick Gravin, Nicole Immorlica, Brendan Lucier, and Emmanouil Pountourakis. Procrastination with variable present bias. In *Proceedings of the 17th ACM Conference on Economics and Computation*, pages 361–361. ACM, 2016.

[14] Leonard Green, Nathanael Fristoe, and Joel Myerson. Temporal discounting and preference reversals in choice between delayed outcomes. *Psychonomic Bulletin Review*, 1(3):383–389, 1994.

[15] Kris N. Kirby and Nino N. Maraković. Modeling myopic decisions: Evidence for hyperbolic delay-discounting within subjects and amounts. *Organizational Behavior and Human Decision Processes*, 64(1):22–30, 1995.

[16] Jon Kleinberg and Sigal Oren. Time-inconsistent planning: A computational problem in behavioral economics. In *Proceedings of the 15th ACM Conference on Economics and Computation*, pages 547–564. ACM, 2014.

[17] Jon Kleinberg, Sigal Oren, and Manish Raghavan. Planning problems for sophisticated agents with present bias. In *Proceedings of the 17th ACM Conference on Economics and Computation*, pages 343–360. ACM, 2016.

[18] Jon Kleinberg, Sigal Oren, and Manish Raghavan. Planning with multiple biases. In *Proceedings of the 18th ACM Conference on Economics and Computation*, pages 567–584. ACM, 2017.

[19] David Laibson. Golden eggs and hyperbolic discounting. *The Quarterly Journal of Economics*, 112(2):443–478, 1997.

[20] James F. Lynch. The equivalence of theorem proving and the interconnection problem. *ACM SIGDA Newsletter*, 5(3):31–36, 1975.

[21] Ted O'Donoghue and Matthew Rabin. Doing it now or later. *American Economic Review*, 89(1):103–124, 1999.

[22] Ted O'Donoghue and Matthew Rabin. Choice and procrastination. *The Quarterly Journal of Economics*, 116(1):121–160, 2001.

[23] Pingzhong Tang, Yifeng Teng, Zihe Wang, Shenke Xiao, and Yichong Xu. Computational issues in time-inconsistent planning. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence*, pages 3665–3671. AAAI, 2017.

[24] Richard Thaler. Some empirical evidence on dynamic inconsistency. *Economics Letters*, 8(3):201–207, 1981.