# Settling the open problems in Kleinberg-Oren time-inconsistent planning models

SHENKE XIAO, Tsinghua University
YICHONG XU, Tsinghua University
YIFENG TENG, Tsinghua University
ZIHE WANG, Tsinghua University
PINGZHONG TANG, Tsinghua University

This article shows an extended work of [Kleinberg and Oren 2014]. It gives more properties of the graph time-inconsistency model raised in [Kleinberg and Oren 2014] by solving three open problems proposed in [Kleinberg and Oren 2014]. Generally speaking, it gives a tight upper bound for the ratio of actual cost due to time-inconsistency to minimum theoretic cost, and shows the hardness of motivating agent. In the end of this article, it also shows some directions of further work based on this model or its extended version.

Categories and Subject Descriptors: J.4 [**Social and Behavioral Sciences**]: Economics

Additional Key Words and Phrases: behavior economics, time-inconsistency, NP-hardness

## 1. INTRODUCTION

This article is generally based on [Kleinberg and Oren 2014]. According to [Kleinberg and Oren 2014], there is a common fact in behavioral economics that people always prefer present profile to future profile, which is called time-inconsistency. [Kleinberg and Oren 2014] has formalized a graph-theoretic model to simulate this phenomenon. In this model, it regards the edges as steps to reach the final target, and each edge has a cost which represents the cost of this step. To keep the integrality of this article, we will restate the model in the next section. [Kleinberg and Oren 2014] has shown many interesting properties of this model, and proposes three open problems in the end. The main result of this article is to solve these problems.

Section 2 restates this basic graph model shown by Kleinberg and Oren formally. The next three sections solves the three problems respectively. Section 3 gives a tight bound of the cost ratio (i.e. the ratio between the ideal minimum cost and the actual cost due to time-inconsistency) for a graph with some constraints. Section 4 shows the hardness of motivating an agent by removing edges or nodes in the graph, while Section 5 shows the hardness of motivating an agent by dispersing the reward over nodes in the graph, by reducing them to well-known 3-SAT problem. At last in Section 6, we shows a conclusion and some further directions of relative works.

## 2. BASIC MODEL

In this section we restate the graph-theoretic model shown by [Kleinberg and Oren 2014] and fix some important definitions. As [Kleinberg and Oren 2014] defines, we have an acyclic direct graph $G$ with a start node $s$ and a target node $t$, where each edge $(u, v)$ has a non-negative cost $c(u, v)$. For any node $u, v$, denote by $d(u, v)$ the minimum total cost from $u$ to $v$, i.e.

$$d(u,v) = \min_{P\in\mathcal{P}(u,v)} \sum_{e\in P} c(e),$$

where $\mathcal{P}(u, v)$ is the set of all possible paths from $u$ to $v$. For simplifying, denote $d(v) = d(v, t)$ for any node $v$. We also have a present-bias parameter $\beta \in [0, 1]$. An agent starts at $s$ and goes along the edges towards $t$. In each step, the agent lying in node $u$ chooses the out-neighbor $v$ that minimizes $c(u, v) + \beta d(v)$ (if more than one node minimizes this value, the agent choose one arbitrarily).

We can see this model simulates the time-inconsistency phenomenon to some extent.

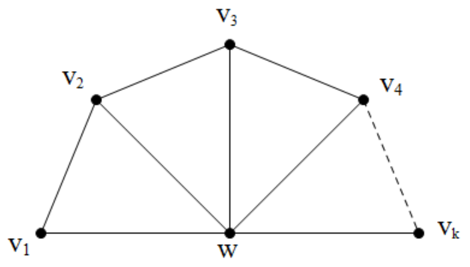

Fig. 1. a visible graph of $\mathcal{F}_k$

While making choices, the agent considers future cost by multiplying the cost with the coefficient $\beta$, which means that it cares more about present (the cost of edge it chooses presently). The smaller $\beta$ is, the less the agent cares about future. Suppose $P$ is the $s$-$t$ path the agent chooses, then the *cost ratio* $r$ is defined as $\sum_{e\in P} c(e)/d(s)$, i.e. the ratio of actual cost to theoretical minimum cost.

There are also some important concepts given in [Kleinberg and Oren 2014] and we would restate them as follows. "Given two undirected graph $H$ and $K$, we say that $H$ *contains a $K$-minor* if we can map each node $\kappa$ of $K$ to a connected subgraph $S_\kappa$ in $H$, with the properties that (i) $S_\kappa$ and $S_{\kappa'}$ are disjoint for every two nodes $\kappa$, $\kappa'$ of $K$, and (ii) if $(\kappa, \kappa')$ is an edge of $K$, then in $H$ there is some edge connecting a node in $S_\kappa$ with a node in $S_{\kappa'}$" ([Kleinberg and Oren 2014] 554). "Let $\sigma(G)$ denote the skeleton of $G$, the undirected graph obtained by removing the directions on the edges of $G$. Let $\mathcal{F}_k$ denote the graph with nodes $v_1, v_2, \ldots, v_k$, and $w$, and edges $(v_i, v_{i+1})$ for $i = 1, \ldots, k-1$, and $(v_i, w)$ for $i = 1, \ldots, k$" ([Kleinberg and Oren 2014] 555). See Figure 1. $\mathcal{F}_k$ is a very special structure in this setting. [Kleinberg and Oren 2014] has shown that a graph containing $\mathcal{F}_k$-minor may have exponential cost ratio, and a graph with asymptotic exponential cost ratio must contain an $\mathcal{F}_k$-minor. In fact, we will give a stronger result immediately in the next section.

## 3. MAXIMUM COST RATIO

The first open problem proposed by [Kleinberg and Oren 2014] is that what is the maximum cost ratio (or do not exist) if $\sigma(G)$ does not contain an $\mathcal{F}_k$-minor. Roughly speaking, the problem asks how much can the waste resulted by time-inconsistency reaches. Note an edge is exactly an $\mathcal{F}_1$, we will assume $k > 1$ in the following article. Next we will prove the following theorem:

THEOREM 3.1. *For any $k > 1$, if $\sigma(G)$ does not contain an $\mathcal{F}_k$-minor, the cost ratio is at most $\beta^{2-k}$.*

We prove some lemmas firstly.

### 3.1. Some Lemmas

LEMMA 3.2. *Fix real numbers $t_1, t_2, \ldots, t_n$ such that $t_i \geq i + 1$ for $i = 1, 2, \ldots, n$. Let $S_i = \{j|1 \leq j < i, t_j \geq i\}$ for $i = 1, 2, \ldots, n$. Then for any sequence $\{x_i, i = 1, 2, \ldots\}$ and for $m = 2, 3, \ldots, n$, we have*

$$x_{m-1} + \sum_{j\in S_{m-1}} x_j = \sum_{j:t_j=m-1} x_j + \sum_{j:m-1<t_j<m} x_j + \sum_{j\in S_m} x_j,$$

*and*

$$\sum_{j\in S_m} x_j = x_{m-1} + \sum_{j\in S_{m-1}:t_j\geq m} x_j.$$

PROOF. For $m = 2, 3, \ldots, n$, note $t_{m-1} \geq m$, i.e. $m-1 \in S_m$ and $m-1 \notin S_{m-1}$, we have

$$\begin{aligned} S_{m-1} \cup \{m-1\} &= \{m-1\} \cup \{j|1 \leq j < m-1, t_j \geq m-1\} \\ &= \{m-1\} \cup \{j|1 \leq j < m-1, t_j \geq m\} \cup \{j|m-1 \leq t_j < m-1\} \\ &= S_m \cup \{j|t_j = m-1\} \cup \{j|m-1 < t_j < m\}, \end{aligned}$$

where all sets in each side of the equality are disjoint. Thus

$$x_{m-1} + \sum_{j\in S_{m-1}} x_j = \sum_{j:t_j=m-1} x_j + \sum_{j:m-1<t_j<m} x_j + \sum_{j\in S_m} x_j.$$

Also note

$$\sum_{j:t_j=m-1} x_j + \sum_{j:m-1<t_j<m} x_j = \sum_{j:m-1\leq t_j<m} x_j = \sum_{j\in S_{m-1}:m-1\leq t_j<m} x_j,$$

we have

$$\sum_{j\in S_m} x_j = x_{m-1} + \sum_{j\in S_{m-1}} x_j - \sum_{j\in S_{m-1}:m-1\leq t_j<m} x_j = x_{m-1} + \sum_{j\in S_{m-1}:t_j\geq m} x_j.$$

□

LEMMA 3.3. *Fix real numbers $t_1, t_2, \ldots, t_n$ such that $t_i \geq i+1$ for $i = 1, 2, \ldots, n$. Let $S_i = \{j|1 \leq j < i, t_j \geq i\}$ for $i = 1, 2, \ldots, n$. Sequence $\{a_i, i = 1, 2, \ldots\}$ and $\{b_i, i = 1, 2, \ldots\}$ are constructed as follows:*

$$a_1 = 1, a_i = \beta b_{i-1} + \sum_{j:i-1<t_j<i} (1-\beta)b_j,$$

$$b_i = a_i + \sum_{j:t_j=i} (1-\beta)b_j.$$

*Then for $i = 1, 2, \ldots, n+1$, we have*

$$a_i \geq \beta^{|S_i|}.$$

PROOF. Fix $i$. Firstly we prove that for $m = 1, 2, \ldots, n$ we have

$$a_m + \sum_{j\in S_m} (1-\beta)b_j = 1.$$

This can be proved by induction on $m$: it holds for $m = 1$, and if it holds for $m-1$, then

$$\begin{aligned} & a_m + \sum_{j\in S_m} (1-\beta)b_j \\ = \ & b_{m-1} - (1-\beta)b_{m-1} + \sum_{j:m-1<t_j<m} (1-\beta)b_j + \sum_{j\in S_m} (1-\beta)b_j \\ = \ & a_{m-1} + \sum_{j:t_j=m-1} (1-\beta)b_j - (1-\beta)b_{m-1} + \sum_{j:m-1<t_j<m} (1-\beta)b_j + \sum_{j\in S_m} (1-\beta)b_j \\ = \ & a_{m-1} + \sum_{j\in S_{m-1}} (1-\beta)b_j \qquad (1) \end{aligned}$$

$$= 1,$$

where equality (1) is held by Lemma 3.2. Let $S_i = \{j_1, j_2, \ldots, j_{|S_i|}\}$ where $j_1 < j_2 < \cdots < j_{|S_i|}$. Then for $l = 1, 2, \ldots, |S_i|$ we have $t_{j_l} \geq i > j_l$, thus for all $1 \leq p \leq l \leq |S_i|$, we have $t_{j_p} \geq i \geq j_l + 1 > j_p$, i.e. $j_p \in S_{j_l+1}$. Now we can obtain for $l = 1, 2, \ldots, |S_i|$,

$$b_{j_l} + \sum_{p=1}^{l-1}(1-\beta)b_{j_p} = \beta b_{j_l} + \sum_{p=1}^{l}(1-\beta)b_{j_p} \leq a_{j_l+1} + \sum_{j \in S_{j_l+1}}(1-\beta)b_j = 1.$$

Note $a_i + \sum_{p=1}^{|S_i|}(1-\beta)b_{j_p} = 1$, to prove $a_i \geq \beta^{|S_i|}$, we only need to prove $\sum_{p=1}^{|S_i|}(1-\beta)b_{j_p} \leq 1 - \beta^{|S_i|}$, or more strongly

$$\sum_{p=1}^{l}(1-\beta)b_{j_p} \leq 1 - \beta^l$$

for $l = 1, 2, \ldots, |S_i|$. We prove it by induction on $l$: for $l = 1$ it holds trivially, and if it holds for $l - 1$, then

$$\begin{aligned}
\sum_{p=1}^{l}(1-\beta)b_{j_p} &= \sum_{p=1}^{l-1}(1-\beta)b_{j_p} + (1-\beta)b_{j_l} \\
&\leq \sum_{p=1}^{l-1}(1-\beta)b_{j_p} + (1-\beta)\left(1 - \sum_{p=1}^{l-1}(1-\beta)b_{j_p}\right) \\
&= \beta\sum_{p=1}^{l-1}(1-\beta)b_{j_p} + 1 - \beta \\
&\leq \beta\left(1 - \beta^{l-1}\right) + 1 - \beta \\
&= 1 - \beta^l.
\end{aligned}$$

□

LEMMA 3.4. *Let $P$ be a path of $G$, and $u_1, u_2, \ldots, u_k$ are nodes in order in $P$. If for $i = 1, 2, \ldots, k-1$, there exists a path $P_i$ such that (i) it starts at $u_i$, and (ii) the second crossing point in $P_i$ with $P$ (say $u_i'$) exists and lies behind $u_k$ (not including $u_k$) in $P$, then $\sigma(G)$ contains an $\mathcal{F}_k$-minor.*

PROOF. Let $v_1, v_2, \ldots, v_k, w'$ name the nodes of an $\mathcal{F}_k$ such that there are exactly edges $(v_i, v_{i+1})$ for $i = 1, 2, \ldots, k-1$ and edges $(v_i, w')$ for $i = 1, 2, \ldots, k$. Let $U_1, U_2, \ldots, U_k, W$ be subgraphs of $\sigma(G)$ which are constructed by their nodes completely: there is an edge between two nodes in one subgraph if and only if this edge exists in $\sigma(G)$. The nodes of these subgraph are collected as follows: for $i = 1, 2, \ldots, k-1$, nodes of $U_i$ are nodes between $u_i$ and $u_{i+1}$ in $P$ (including $u_i$ and not including $u_{i+1}$); $U_k$ is exactly $u_k$; and nodes $W$ are made up of $k$ parts $V_1, V_2, \ldots, V_{k-1}, W'$ where $V_i$ are nodes between $u_i$ and $u_i'$ in $P_i'$ (including $u_i'$ and not including $u_i$) for $i = 1, 2, \ldots, k-1$ and $W'$ are all nodes after $u_k$ in $P$ (not including $u_k$). See Figure 2.

For each $1 \leq i \leq k$, $U_i$ is connected, as its nodes are connected by $P$. Then we show that $W$ is also connected. Note that the nodes in $V_i$ are connected through $P_i$ for $i = 1, 2, \ldots, k-1$ while the nodes in $W'$ are connected through $P$, and $V_i$ and $W'$ are also joint by $u_i'$ for $i = 1, 2, \ldots, k-1$, we can conclude that $W$ is connected.

Trivial to see that $U_1, U_2, \ldots, U_k, W$ are disjoint. For $i = 1, 2, \ldots, k-1$, $U_i$ and $U_{i+1}$ are connected by the edge lying before $u_{i+1}$ in $P$, and $U_i$ and $W$ are connected by the edge

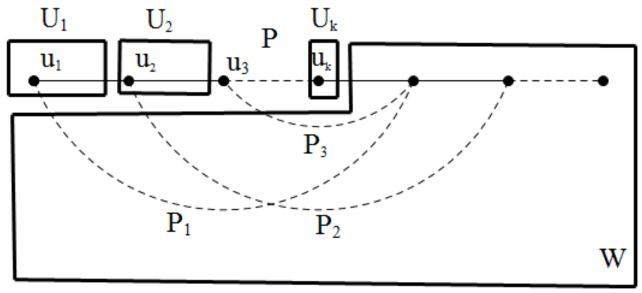

Fig. 2. the graph contains an $\mathcal{F}_k$-minor.

lying after $u_i$ in $P_i'$. $U_k$ and $W$ are connected by the edge lying after $u_k$ in $P$. Now we can establish a map $f$ from $\{v_1, v_2, \ldots, v_k, w'\}$ to $\{U_1, U_2, \ldots, U_k, W\}$ where $f(v_i) = U_i$ for $i = 1, 2, \ldots, k$ and $f(w') = W$. By definition, we can conclude that $\sigma(G)$ contains an $\mathcal{F}_k$-minor. □

## 3.2. Main Proof

Now we would prove Theorem 3.1.

PROOF. We firstly fix some notations: Let $P$ be the path that the agent actually travels. For any node $u$ in $P$, denote by $u'$ the first node after $u$ in $P$. For any pair of nodes $u, v$ in $P$, denote by $c(u, v)$ the total cost of edges between $u$ and $v$ in $P$. A node $u$ in $P$ is a *shortcut node* if the second point of the path from $u$ to $t$ with minimum total cost (note the first node is just $u$) is not $u'$, and suppose that there are $n$ shortcut nodes: $u_1, u_2, \ldots, u_n$ in order. For $i = 1, 2, \ldots, n$, denote by $w_i$ the second crossing point of $P$ and the $u_i$-$t$ path $P_i$ with the minimum total cost (note the first crossing point is $u_i$, and if more than one path has the minimum total cost, arbitrarily choose one), and denote by $v_i$ the first node after $u_i$ on $P_i$.

Note the agent at $u_i$ does not choose $P_i'$, we have

$$d(u_i, w_i) + \beta d(w_i) \geq c(u_i, v_i) + \beta d(v_i, w_i) + \beta d(w_i) \geq c(u_i, u_i') + \beta d(u_i').$$

Adding both sides by $(1 - \beta) d(w_i)$ we have

$$d(u_i) \geq c(u_i, u_i') + \beta d(u_i') + (1 - \beta) d(w_i). \tag{2}$$

Now for $i = 1, 2, \ldots, n$, define $t_i$ as follows: if there exists $j$ such that $w_i = u_j$, then $t_i = j$; otherwise if there exists $j$ such that $w_i = u_j'$, then $t_i = j + 0.5$; otherwise $t_i$ is the smallest index such that $u_{t_i}$ lies after $w_i$ in $P$ (if no such index, $t_i = n + 1$). We have the following lemma.

LEMMA 3.5. *For $i = 1, 2, \ldots, n$, $t_i$ has the following properties:*

(*1*) $t_i \geq i + 1$;
(*2*) *if $j < t_i < j + 1$ for some positive integer $j$, then $w_i = u_j'$;*
(*3*) *if $t_i$ is an integer, $d(w_i) \geq d(u_{t_i})$;*
(*4*) *if $t_i \geq j$ for some positive integer $j > 1$, then $w_i$ lies after $u_{j-1}$ in $P$.*

PROOF. (1) We know $w_i$ must lie after $u_i$ in $P$. If $w_i = u_i'$, note $u_i$ is a shortcut node, we have

$$c(u_i, u_i') + d(u_i') > d(u_i) \geq c(u_i, u_i') + \beta d(u_i') + (1 - \beta) d(w_i) = c(u_i, u_i') + d(u_i'),$$

which is a contradiction! Hence $w_i$ must lie after $u_i'$ in $P$. By the definition of $t_i$, if there exists $j$ such that $w_i = u_j$, then $u_j$ lies after $u_i'$ in $P$, i.e. $t_i = j \geq i + 1$. Otherwise if there exists $j$ such that $w_i = u'_j$, then $u'_j$ lies after $u'_i$ in $P$, i.e. $t_i = j + 0.5 > j \geq i + 1$; if such $j$ does not exist, $u_{t_i}$ lies after $w_i$, or $u'_i$ in $P$, i.e. $t_i \geq i + 1$.

(2) In this case $t_i = j + 0.5$, then $w_i = u'_j$ by the definition of $t_i$ directly.

(3) If there exists $j$ such that $w_i = u_j$, then $t_i = j$ so that $d(w_i) = d(u_j) = d(u_{t_i})$. Otherwise if there exists $j$ such that $w_i = u'_j$, $t_i$ is not an integer, and it makes a contradiction; if such $j$ does not exist, $u_{t_i}$ lies after $w_i$ in $P$ and there is no shortcut node between them (otherwise we will choose this node as $u_{t_i}$), hence $d(w_i) = d(u_{t_i}) + c(w_i, u_{t_i}) \geq d(u_{t_i})$.

(4) If there exists $l$ such that $w_i = u_l$, then $l = t_i \geq j$, hence $u_l$, or $w_i$ lies after $u_{j-1}$ in $P$. Otherwise if there exists $l$ such that $w_i = u'_l$, then $l + 0.5 = t_i \geq j$, or $l \geq j$, hence $u'_l$, or $w_i$ lies after $u_{j=1}$ in $P$; if such $l$ does not exist, $t_i$ is the smallest index such that $u_{t_i}$ lies after $w_i$ in $P$, also note there does not exist $l$ such that $w_i = u_i$, we have $u_{t_i-1}$, or $u_{j-1}$ lies before $w_i$ in $P$. □

By Lemma 3.5 (1), we can use these $t_i$ as the same parameters in Lemma 3.3 to construct $S_1, S_2, \ldots, S_n$ and sequences $\{a_i, i = 1, 2, \ldots\}, \{b_i, i = 1, 2, \ldots\}$. Generally speaking, $S_i$ represents those shortcut nodes whose shortcut "comes back" after or at $u_i$. Next we are trying to use inequality (2) iteratively to get an inequality with the following form:

$$d(s) \geq \sum_{e \in P} \alpha(e) c(e),$$

for some coefficient $\alpha$. If the inequality with this form is obtained, the bound with cost ratio can be obtained easily. In fact we have the following inequality that for $i = 1, 2, \ldots, n + 1$,

$$d(s) \geq \sum_{j=1}^{i} \left( a_j c(u'_{j-1}, u_j) + b_j c(u_j, u'_j) \right) + a_{i+1} d(u'_i) + \sum_{j \in S_{i+1}} (1 - \beta) b_j d(w_j),$$

where $u'_0$ and $u_{n+1}$ are defined as $s$ and $t$ respectively. We prove the inequality above by induction on $i$: it holds for $i = 1$, and if it holds for $i - 1$, then

$$\begin{aligned}
d(s) \geq\ & \sum_{j=1}^{i-1} \left( a_j c(u'_{j-1}, u_j) + b_j c(u_j, u'_j) \right) + a_i d(u'_{i-1}) + \sum_{j \in S_i} (1 - \beta) b_j d(w_j) \\
=\ & \sum_{j=1}^{i-1} \left( a_j c(u'_{j-1}, u_j) + b_j c(u_j, u'_j) \right) + a_i \left( c(u'_{i-1}, u_i) + d(u_i) \right) + \sum_{j : t_j = i} (1 - \beta) b_j d(w_j) \\
& + \sum_{j : i < t_j < i+1} (1 - \beta) b_j d(w_j) + \sum_{j \in S_i : t_j \geq i+1} (1 - \beta) b_j d(w_j) \\
\geq\ & \sum_{j=1}^{i-1} \left( a_j c(u'_{j-1}, u_j) + b_j c(u_j, u'_j) \right) + a_i c(u'_{i-1}, u_i) + \left( a_i + \sum_{j : t_j = i} (1 - \beta) b_j \right) d(u_i) \\
& + \sum_{j : i < t_j < i+1} (1 - \beta) b_j d(u'_i) + \sum_{j \in S_i : t_j \geq i+1} (1 - \beta) b_j d(w_j) \qquad (3) \\
\geq\ & \sum_{j=1}^{i-1} \left( a_j c(u'_{j-1}, u_j) + b_j c(u_j, u'_j) \right) + a_i c(u'_{i-1}, u_i) \\
& + b_i \left( c(u_i, u'_i) + \beta d(u'_i) + (1 - \beta) d(w_i) \right)
\end{aligned}$$

$$
\begin{aligned}
&+ \sum_{j:i<t_j<i+1} (1-\beta) b_j d\left(u_i'\right) + \sum_{j\in S_i : t_j \ge i+1} (1-\beta) b_j d\left(w_j\right) \\
&\ge \sum_{j=1}^{i} \left(a_j c(u_{j-1}', u_j) + b_j c(u_j, u_j')\right) + \left(\beta a_i + \sum_{j:i<t_j<i+1} (1-\beta) b_j\right) d(u_i') \\
&\quad + (1-\beta) b_i d\left(w_i\right) + \sum_{j\in S_i : t_j \ge i+1} (1-\beta) b_j d\left(w_j\right) \\
&= \sum_{j=1}^{i} \left(a_j c(u_{j-1}', u_j) + b_j c(u_j, u_j')\right) + a_{i+1} d(u_i') + \sum_{j\in S_{i+1}} (1-\beta) b_j d\left(w_j\right), \qquad (4)
\end{aligned}
$$

where inequality (3) is held by Lemma 3.5 (2) and (3), and the last step (4) is held by Lemma 3.2. Now choose $i$ to be $n$, we can get

$$
\begin{aligned}
d(s) &\ge \sum_{j=1}^{n} \left(a_j c\left(u_{j-1}', u_j\right) + b_j c\left(u_j, u_j'\right)\right) + a_{n+1} d(u_n') \\
&\ge \sum_{j=1}^{n} \left(\beta^{|S_j|} c\left(u_{j-1}', u_j\right) + \beta^{|S_j|} c\left(u_j, u_j'\right)\right) + \beta^{|S_{n+1}|} c\left(u_n', t\right). \qquad (5)
\end{aligned}
$$

where the last step (5) holds by Lemma 3.3.

At last, we are going to prove $|S_i| \le k-2$ for $i = 1, 2, \ldots, n+1$. If $|S_i| \ge k-1$ for some $i$, choose $k-1$ elements from $S_i$, say $j_1, j_2, \ldots, j_{k-1}$ where $j_1 < j_2 < \cdots < j_{k-1} < i$. For $l = 1, 2, \ldots, k-1$, note $t_{j_l} \ge i$, by Lemma 3.5 (4) $w_{j_l}$ must lie after $u_{i-1}$, or $u_{j_{k-1}}$ in $P$, and also $w_{j_l} \ne u_{j_{k-1}}'$ (otherwise $t_{j_l} = j_{k-1} + 0.5 < i$), $w_{j_l}$ must lie after $u_{j_{k-1}}'$ in $P$. Now consider $u_{j_1}, u_{j_2}, \ldots, u_{j_{k-1}}, u_{j_{k-1}}'$. They lie in $P$ in order, and for $l = 1, 2, \ldots, k-1$, $P_{j_l}$ starts at $u_{j_l}$, and the next crossing point of $P_{j_l}$ and $P$ lies after $u_{j_{k-1}}'$ in $P$, thus by Lemma 3.4, $\sigma(G)$ contains an $\mathcal{F}_k$-minor, which forms a contradiction. Now we have proved $|S_i| \le k-2$.

Hence

$$
d(s) \ge \sum_{j=1}^{n} \left(\beta^{|S_j|} c\left(u_{j-1}', u_j\right) + \beta^{|S_j|} c\left(u_j, u_j'\right)\right) + \beta^{|S_{n+1}|} c\left(u_n', t\right) \ge \beta^{k-2} c(s,t),
$$

or

$$
\frac{c(s,t)}{d(s)} \le \beta^{2-k}. \qquad \square
$$

## 3.3. Tightness of the Bound

So far we have achieved an upper bound for $r$, now we would like to provide an example to show that the bound is achievable. We would simply use the example mentioned in [Kleinberg and Oren 2014] to show a graph may have exponential cost ratio, i.e. the graph obtained from $\mathcal{F}_{k-1}$ by adding directions (see Figure 3). In fact, this example is a real example of shipping packages that was proposed firstly in [Akerlof 1991]. According to the analysis in [Kleinberg and Oren 2014], the cost ratio of the graph is exactly $\beta^{2-k}$, which proves the tightness of our upper bound.

## 4. HARDNESS OF FINDING MOTIVATING SUBGRAPHS

[Kleinberg and Oren 2014] also extends the model to allow the agent to give up: it sets a reward $r$ at the target node, and suppose the agent is in $u$ now, if $\min_v c(u,v) + \beta d(v) >$

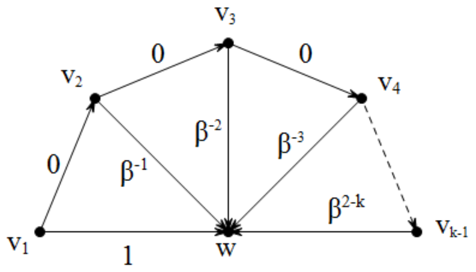

Fig. 3. Akerlof's example

$\beta r$, i.e. the weighted cost is less than the reward, the agent will stop and give up. [Kleinberg and Oren 2014] also gave the definition of *motivating graph*: a graph is motivating if the agent will arrive at the target node finally. [Kleinberg and Oren 2014] stated that for a non-motivating graph, we can delete some nodes or edges to make it motivating. The second problem is whether there is a polynomial-time algorithm that "takes an instance $G$ (including a reward $r$) and determines whether $G$ contains a motivating subgraph with reward $r$?" ([Kleinberg and Oren 2014] 563), i.e. to find a motivating subgraph of the original graph. In this section, we are going to show the following theorem:

THEOREM 4.1. *Define problem* ***ms*** *as follows: for an acyclic graph $G$ with $n$ nodes, given reward $r$ on target node and present bias $\beta$, find a motivating subgraph of $G$. Then* ***ms*** *is NP-hard.*

[Kleinberg and Oren 2014] introduced a concept named *minimal motivating subgraph* to denote a motivating subgraph of the original graph, such that every subgraph of it is not motivating. Before proving Theorem 4.1, we first consider a easier problem related to minimal motivating subgraphs.

### 4.1. Hardness of finding minimal motivating subgraph

In this section, we would show that finding a minimal motivating subgraph is hard. Define problem **mms** as follows: *for an acyclic graph $G$ with $n$ nodes, given reward $r$ on target node and present bias $\beta$, find a minimal motivating subgraph of $G$.* Then we have

THEOREM 4.2. ***mms*** *is NP-hard.*

PROOF. We show that finding a valid assignment to a 3-CNF can be polynomial-time reduced to an instance of **mms**. Consider a 3-CNF with $n$ variables $x_1, x_2, ..., x_n$ and $m$ clauses $C_1, C_2, ..., C_m$. Construct a weighted acyclic graph $G$ with $4m + 3n + 7$ vertices as follows:

(1) Each clause $C_i$ corresponds to 4 nodes $u_i, u_{i,1}, u_{i,2}, u_{i,3}$, $1 \leq i \leq m$;
(2) Each variable $x_i$ corresponds to 3 nodes $v_i, v_{i,1}, v_{i,2}$, $1 \leq i \leq n$;
(3) The other 7 nodes are start node $s$, target node $t$ and interior points $w_1, w_2, w_3, w_4$ and $w_5$;
(4) For any clause $C_i = y_{i,1} \vee y_{i,2} \vee y_{i,3}$: for $j = 1, 2, 3$, if $y_{i,j}$ is $x_k$, then there is a link $u_i \rightarrow u_{i,j}$ with weight 16, and a link $u_{i,j} \rightarrow v_k$ with weight 11. We call such two links an "expensive path" from $u_i$ to $v_k$; if $y_{i,j}$ is $\neg x_k$, then there is a link $u_i \rightarrow u_{i,j}$ with weight 9, and a link $u_{i,j} \rightarrow v_k$ with weight 11. We call such two links a "cheap path" from $u_i$ to $v_k$.

(5) For any $1 \leq i \leq n$: there is a link $v_i \to v_{i,1}$ with weight 19, and a link $v_{i,1} \to t$ with weight 0. We call such two links an "expensive path" from $v_i$ to $t$; also, there is a link $v_i \to v_{i,2}$ with weight 12, and a link $v_{i,2} \to t$ with weight 0. We call such two links a "cheap path" from $v_i$ to $t$.

(6) $s \to u_1$ forms a link with weight 8. For each $1 \leq i \leq m-1$, $u_i \to u_{i+1}$ forms a link with weight $8$. $u_m \to w_1$, $w_1 \to w_2$, $w_2 \to w_3$, $w_3 \to v_1$ are arcs with weight 8, 8, 9, 10 respectively. For each $1 \leq i \leq n-1$, $v_i \to v_{i+1}$ forms a link with weight $10$. $v_n \to w_4 \to t$ are two links with weight 10 respectively. The path from $s$ to $t$ mentioned above is called "bus". There are final two links: $w_1 \to w_5$ with weight 39, and $w_5 \to t$ with weight 0.

As an example, figure 4 is a graph constructed from 3-CNF $(x_1 \vee \neg x_2 \vee x_3) \wedge (x_2 \vee \neg x_3 \vee x_4)$ following the instructions above. All expensive paths lie above the bus, while all cheap paths are drawn below the bus.

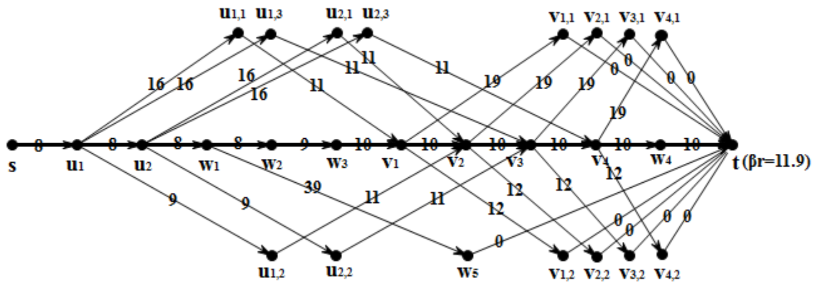

Fig. 4. Corresponding graph of $(x_1 \vee \neg x_2 \vee x_3) \wedge (x_2 \vee \neg x_3 \vee x_4)$.

To form an instance of **mms**, define the reward at $t$ to be $r = 119$, and let the present bias $\beta$ be assigned 0.1. It can be easily seen that such instance can be constructed from the 3-CNF in polynomial time. Now we have the following three lemmas about the structure of minimal motivating subgraph of $G$.

LEMMA 4.3. *Let $G'$ be a minimal motivating subgraph of $G$, then the bus of $G$ is fully included in $G'$.*

PROOF. Assume by way of contradiction some nodes on the bus are not included in $G'$. Consider the path traveled by the agent in $G'$, and let $v$ be the first vertex not on the bus. Let $u \to v$ be the edge that the agent passes. Notice that the cost of the edge should not be greater than 11.9, otherwise the agent would rather not follow the path. From the construction we know that $v$ must be $u_{i,j}$ for some $i, j$, and let $v_k$ be the node that can be directly reached from $u_{i,j}$. But notice that at $u_{i,j}$ the agent would rather give up, which forms a contradiction: the cost from $v_k$ to $t$ is at least 12, thus the evaluated cost from $u_{i,j}$ to $t$ is no less than $11 + 12\beta = 12.2 > \beta r$. Thus the bus of $G$ is fully included in $G'$. □

LEMMA 4.4. *Let $G'$ be a minimal motivating subgraph of $G$, then for any $1 \leq i \leq n$, exactly one path between the expensive path and the cheap path from $v_i$ to $t$ is preserved.*

PROOF. Fix $i$. Theorem 5.1 in [Kleinberg and Oren 2014] stated that every node in a minimal motivating subgraph has no more than two outgoing edges, thus at most one of the two nodes can be included, as $v_i$ has already got one out-degree in the bus from Lemma 4.3. If both paths are eliminated, the agent would not travel to $v_i$ as its evaluated cost from the former node of $v_i$ to $t$ would be at least $10 + (10 + 12)\beta = 12.2 > \beta r$. Thus exactly one path remains. □

LEMMA 4.5. *Let $G'$ be a minimal motivating subgraph of $G$, then for any $1 \leq i \leq m$, exactly one node among $u_{i,1}, u_{i,2}, u_{i,3}$ is included in $G'$.*

PROOF. Fix $i$. Theorem 5.1 in [Kleinberg and Oren 2014] stated that every node in a minimal motivating subgraph has no more than two outgoing edges, thus at most one of the three nodes can be included, as $u_i$ has already got one out-degree in the bus from Lemma 4.3. If all three nodes are eliminated, the agent would not travel to $u_i$ as its evaluated cost from the former node of $v_i$ to $t$ would be at least $8+(8+39)\beta = 12.7 > \beta r$. Hence exactly one of the three nodes remains in $G'$. □

Now we would show how to transform a valid assignment of a 3-CNF to a corresponding minimal motivating subgraph of $G$.

LEMMA 4.6. *Given a valid assignment to a 3-CNF, there exists a polynomial-time algorithm finding a minimal motivating subgraph of graph $G$ constructed from the formula.*

PROOF. Construct a subgraph $G'$ of the original graph $G$ as follows. For $1 \leq i \leq m$, let $C_i = y_{i,1} \vee y_{i,2} \vee y_{i,3}$, assume $y_{i,j}$ is assigned true without loss of generality, then preserve only $u_{i,j}$ among $\{u_{i,1}, u_{i,2}, u_{i,3}\}$, eliminate the other two nodes from $G$; for each $1 \leq i \leq n$, if $x_i$ is assigned true, delete $v_{i,1}$ (the more expensive one), otherwise delete $v_{i,2}$. Observe that for any $1 \leq i \leq m$, there is no path from node $u_i$ to $t$ formed by two consecutive cheap paths or expensive paths. To show that the remaining graph is a motivating graph, it suffices to show that at any time the agent would travel through the bus and does not halt on his way.

(1) The agent is at $u_i$, $0 \leq i \leq m-1$ (let $u_0 = s$): Notice that the distance from $u_{i+1}$ to $t$ is $16 + 11 + 12 = 9 + 11 + 19 = 39$, thus the evaluated cost would be $8 + 3.9 = 11.9$ if the agent wants to go to $u_{i+1}$; the cost to follow the expensive path would be greater than 16, while the cost to travel to follow the cheap path would be $9 + (11 + 19)\beta = 12 > \beta r$. Thus the agent would go straight to $u_{i+1}$.
(2) The agent is at $u_m, w_1, w_2, w_3$ or $w_4$: the only reasonable move for the agent should be following the bus. At $u_m, w_1, w_2, w_3, w_4$, the respective evaluated cost would be 11.9, 11.8, 11.9, 11.9, 10 respectively, which are all no larger than $\beta r$.
(3) The agent is at $v_i$, $1 \leq i \leq n$: the only reasonable move for the agent should be following the bus, and the largest possible evaluated cost would be 11.9, which is no larger than $\beta r$.

Finally, we remove $w_5$ if possible, and it can be verified that $w_5$ can be removed if and only if $x_1$ is assigned true. The minimality of the graph follows from Lemma 4.3, 4.4 and 4.5, and the entire procedure can be definitely done in polynomial time. □

To see how Lemma 4.6 works, consider $(x_1 \vee \neg x_2 \vee x_3) \wedge (x_2 \vee \neg x_3 \vee x_4)$ as an example. A valid assignment would be $x_1 = \neg x_2 = \neg x_3 = x_4 = 1$, and the corresponding minimal motivating subgraph would be shown in Figure 5.

Finally, we would like to see how a minimal motivating subgraph of $G$ can be transformed to a valid assignment to the 3-CNF.

LEMMA 4.7. *Let $G'$ be a minimal motivating subgraph of $G$. For any $1 \leq i \leq n$, if the expensive path from $v_i$ to $t$ is remained, assign $x_i = 0$, otherwise assign $x_i = 1$, then the assignment is a valid assignment to the original 3-CNF.*

PROOF. To show that the assignment is valid, we only need to show that for any $1 \leq i \leq m$, in $G'$ there is a path from $u_i$ to $t$ that is a cheap path from $u_i$ to some $v_k$,

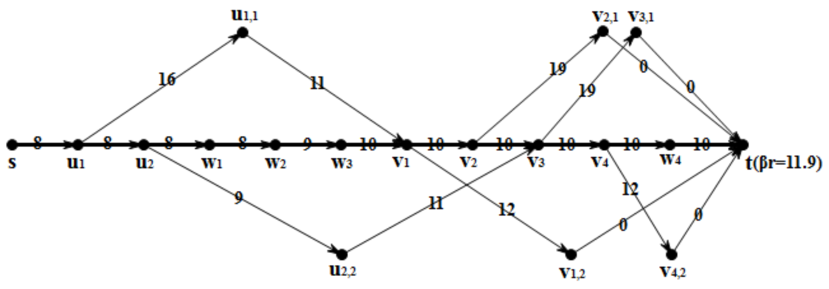

Fig. 5. Corresponding minimal motivating subgraph of $(x_1 \vee \neg x_2 \vee x_3) \wedge (x_2 \vee \neg x_3 \vee x_4)$.

then an expensive path from $v_k$ to $t$, meaning that $x_k$ is assigned false, $C_i$ is satisfied as it includes $\neg x_k$; or there is a path from $u_i$ to $t$ that is an expensive path from $u_i$ to some $v_k$, then a cheap path from $v_k$ to $t$, meaning that $x_k$ is assigned true, $C_i$ is satisfied as it includes $x_k$. From Lemma 4.3, 4.4 and 4.5 we only need to show that there is no path from node $u_i$ to $t$ formed by two consecutive cheap paths or two consecutive expensive paths. Consider the following two cases.

(1) Assume by way of contradiction there is a cheap path from $u_i$ to some $v_k$, then a cheap path from $v_k$ to $t$, here $i$ is maximum. Then at $u_i$, choosing the cheap path leads to an evaluated cost $9 + (11 + 12)\beta = 11.3$, while choosing the bus leads to an evaluated cost at least 11.9. Then the agent would go through the cheap path, while at the next vertex he would evaluate the cost to be at least $11 + 12\beta = 12.2 > \beta r$, then he would halt here, which contradicts the fact that $G'$ is a motivating graph.
(2) Assume by way of contradiction there is an expensive path from $u_i$ to some $v_k$, then an expensive path from $v_k$ to $t$. This is impossible since when the agent is at $u_{i-1}$, he would evaluate traveling through the bus with cost $8+(16+11+19)\beta = 12.6 > \beta r$, which means that he would rather not go through $u_{i-1} \to u_i$.

Directly from the two cases above we show the correctness of the lemma. □

From Lemma 4.6 and Lemma 4.7, we know that if we can find a minimal motivating subgraph for any graph in polynomial time, we may find a polynomial time algorithm finding a valid assignment for any 3-CNF, which finishes the proof of Theorem 4.2. □

### 4.2. Main Proof

Now we are going to prove Theorem 4.1, which implies not only finding a minimal motivating subgraph is hard, it is also difficult to find a not minimal one.

PROOF. Assume by way of contradiction we have a poly-time algorithm $\mathcal{A}$ to solve **ms**, consider the following algorithm solving **mms**.

The correctness of the algorithm is straightforward, while the running time of the algorithm is polynomial of the size of the graph. From Theorem 4.2 we know that **ms** is NP-hard. □

## 5. HARDNESS OF MOTIVATING AGENTS TO REACH THE GOAL

[Kleinberg and Oren 2014] also considers the case where we can put rewards on all nodes in the graph, denoted by $r(v)$ for each node; the agent at node $u$ will continue to

**ALGORITHM 1:** Solve **mms** by Applying Algorithm Solving **ms**.

```
Input: An acyclic graph $G$, reward $r$ on target node and present bias $\beta$.
Output: Reject the input or output a minimal motivating subgraph.
Use $\mathcal{A}$ to check whether $G$ has an motivating subgraph;
if no then
    reject the input;
else
    while $G$ contains some edges do
        for each edge $e$ in $G$ do
            Use $\mathcal{A}$ to check whether $G$ has a motivating subgraph after deleting $e$;
            if yes then
                record $e$;
            end
        end
        if $G$ has no motivating subgraph after deleting $e$ for each edge $e$ then
            output $G$;
        else
            delete an edge recorded from $G$;
        end
    end
end
```

move if and only if there exists a path $P$ such that

$$c'(P) = c(u, v_0) + \beta \sum_{v \in P} (c(v, v') - r(v)) \leq 0,$$

where $v_0$ denotes the first node in $P$, and $v'$ denotes the next node of $v$ on $P$. Let $c(t, t') = 0$. And the agent chooses the path that minimize $c'(P)$. We aim to find the configuration that puts the smallest total sum of rewards that motivates the agent to finish at $t$.

### 5.1. Problem statement

We considers multiple versions of this problem, as suggested by [Kleinberg and Oren 2014]. In the first version, only positive rewards are allowed, and we can put rewards on every node of the graph. But due to the trivialness of this problem, in this we might put rewards that is never claimed - the agent simply use the reward as on its shortest path but then deviates from its original computation. In the third version, we consider the possible use of negative rewards; the negative reward might be used to get the agent rid of paths that appears to be costless but then becomes very costly(for example, a path start with 0 and then an edge with a very large cost). In this version, we define the sum of rewards to be the sum of absolute values of all rewards put on nodes; we do this because if we define the sum as the direct sum, we can put a rewards of $-\infty$ at some node that there still exists a path from $s$ to $t$ if deleted.

We formally restate this problem:

*Finding the Minimum Total Reward for an agent with present bias $\beta$ (**fmtr**$_\beta$)*

— Input: a weighted acyclic graph $G$, node $s, t$, and a possible total possible reward $R$.

— Output: Whether there exists a reward setting $r(v), \forall v \in G$, such that the agent with a present bias $\beta$ follows a path $P$ from $s$ to $t$ using the strategy stated above, and that

$$\sum_{v \in G} |r(v)| \leq R.$$

The following three editions gives different constraints on the reward setting:
**fmtr**$_\beta$ I: $r(v) \geq 0, \forall v$.
**fmtr**$_\beta$ II: $r(v) \geq 0, \forall v \in P; r(v) = 0, \forall v \notin P$.
**fmtr**$_\beta$ III: No constraints: $r(v) \in \mathbb{R}$.

## 5.2. Main Theorem

Unfortunately, we find that this problem, along with each form of it, are all NP-hard:

THEOREM 5.1. ***fmtr**$_\beta$ I,II,III are NP-hard for all $\beta < 1$.*

Generally, we prove the NP-hardness by reducing 3-SAT to **fmtr**$_\beta$.

## 5.3. Graph construction

For any 3-SAT instance, suppose it has $m$ clauses and $n$ variables.
Let $x = \beta^{-1} - 1, y = \frac{1}{2}x$, and

$$\begin{aligned} g_k &= 6(2n-k), \\ h_k &= 6(2n-k) + 6 + y, \\ r_t &= 12n - 6 + 6\beta^{-1}, \\ l &= \left\lceil \frac{nx + 12n + 6\beta^{-1} - 6}{\beta x} \right\rceil + 1. \end{aligned}$$

We choose $n$ large enough such that

$$nx = n(\beta^{-1} - 1) > 2\beta^{-1}.$$

We construct graph $G$ in the following way: We construct $l$ nodes for each clause $i$, namely $a_{i,1}, a_{i,2}, \cdots, a_{i,l}, b_i$; and 4 nodes $u_i, u'_i, v_i, v'_i$ for each variable $i$. We also have node $c^*$, and $s, t$. We construct edges with weights in the following configuration: (each triple is the start node, end node and the weight of an edge respectively)

(1) $(s, a_{1,1}, 0); (a_{i,1}, b_i, 0); (a_{i,j}, a_{i,j+1}, \beta x); (a_{i,l}, a_{i+1,1}, 0); (1 \leq i \leq m-1, 1 \leq j \leq l-1)$
(2) $(a_{m,l}, c_1, 0), (c_i, c_{i+1}, 6), (c_n, u_n, 6), (c_n, u'_n, 6); (1 \leq i \leq n-1)$
(3) $(u_i, v_i, x); (u'_i, v'_i, x); (1 \leq i \leq n)$
(4) $(v_i, u_{i-1}, 6); (v_i, u'_{i-1}, 6); (v'_i, u_{i-1}, 6); (v'_i, u'_{i-1}, 6); (1 \leq i \leq n)$
(5) $(u_0, t, 0); (u'_0, t, 0)$.

For a clause $i$ that contains variable $k$, we construct $(b_i, u_{k-1}, h_k), (b_i, u'_{k-1}, h_k)$, If it contains a positive subclause of $k$, we construct $(a_{i,j}, v_k, g_k), 2 \leq j \leq l$; otherwise if it contains "not variable $k$", we construct $(a_{i,j}, v'_k, g_k), 2 \leq j \leq l$. An illustration of the graph is depicted below. For simplicity, direct values of $h_k$ and $g_k$ are drawn. The graph shows an example of a clause with $v_n$ and $\neg v_k$. The feasibility of this graph is easily seen from the fact that $\beta < 1$ (so $x > 0$). We now prove the following main proposition:

PROPOSITION 5.2. *The 3-CNF is satisfiable if and only if $G$ has a minimum total reward of at most $nx + r_t$.*

## 5.4. Two Lemmas

To start with the proof, we first give two lemmas:

LEMMA 5.3. *We have*

$$h_k\beta^{-1} + 6(k-1) > nx + r_t, \tag{6}$$

$$g_k\beta^{-1} + 6k > nx + r_t, \tag{7}$$

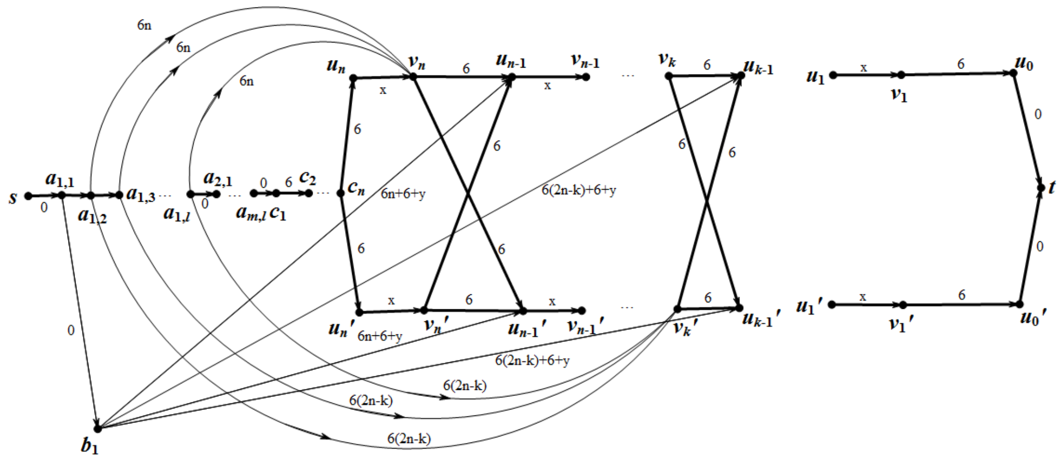

Fig. 6. Example figure for proving theorem 5.1, with a clause with $v_n$ and $\neg v_k$. Edges that might be traveled by the agent are in bold lines.

$$(l-1)\beta x > r_t + nx, \tag{8}$$
$$h_k < x + g_k + 6, \tag{9}$$
$$x + g_k - x + 6k < r_t, \tag{10}$$
$$x + g_k - x + 6k < h_k + 6(k-1), \tag{11}$$
$$\beta x + g_k - x + 6k < r_t, \tag{12}$$
$$6\beta^{-1} + 6(n-1) + 6n = r_t, \tag{13}$$
$$x + g_k - x + 6k < g_k\beta^{-1} - x + 6k. \tag{14}$$

PROOF. We have

$$\begin{aligned} h_k\beta^{-1} + 6(k-1) &= \beta^{-1}(6(2n-k)+6+y) + 6(k-1) \\ &> \beta^{-1}6(2n-k) + 6k \\ &= g_k\beta^{-1} + 6k. \end{aligned}$$

On the other hand,

$$\begin{aligned} g_k\beta^{-1} + 6k &= \beta^{-1}6(2n-k) + 6k \\ &\geq 6n + 6n\beta^{-1} \\ &= 12n + 6n(\beta^{-1}-1) \\ &= 12n + 6nx \\ &> nx + 12n + 6\beta^{-1} - 6 = nx + r_t. \end{aligned}$$

The last inequality is because the way we choose $n$: $nx > 2\beta^{-1}$. So we prove (6) and (7). We just choose an $l$ large enough for (8) to stand: here we have

$$r_t + nx = 12n - 6 + 6\beta^{-1} + nx \leq \beta x(l-1).$$

(9) is obvious from definition; For (10) the left hand side is

$$g_k + 6k = 6(2n-k) + 6k = 12n < r_t.$$

For (11) the right hand side is

$$h_k + 6(k-1) = 12n + y > 12n,$$

which is the left hand side. (12) is a direct induction of (10). (13) is the definition of $r_t$. For (14), it is sufficient to prove that

$$x < g_k(\beta^{-1} - 1).$$

In fact we have

$$g_k(\beta^{-1} - 1) = g_k x = 6(2n - k)x > x.$$

So we prove all the inequalities and equalities. □

LEMMA 5.4. *The agent continues to move if and only if there exists a path $P$ such that*

$$\beta^{-1}c(P) = \beta^{-1}c(e_1) + \sum_{e \in P : e \neq e_1} c(e) \leq \sum_{v \in P} r(v),$$

*where $e_1$ is the first edge of $P$.*

PROOF. This lemma can be obtained simply by multiplying $\beta^{-1}$ in the definition of motivating the agent. □

By this lemma, we assume that the agent evaluates

$$\beta^{-1}c(e_1) + \sum_{e \in P : e \neq e_1} c(e)$$

instead of $c(P)$ for every path $P$ at each node; this makes convenience for discussions follow.

## 5.5. Achievability

PROPOSITION 5.5. *If the 3-CNF is satisfiable, then $G$ has a minimum total reward of at most $nx + r_t$.*

PROOF. If the 3-CNF is satisfiable, we construct a reward as follows: Let $r(v_i) = x$ if variable $i$ is true, otherwise let $r(v_i') = x$; and $r(t) = r_t$. All other nodes have reward 0. We now prove that this is a valid setting: We define the length value of a path $P$ from $v$ to $v'$, denoted $l(P)$, to be the sum of all weights on the path minus sum of all rewards on the path except $v$ and $v'$:

$$l(P) = \sum_{e \in P} c(e) - \sum_{u \in P : u \neq v, v'} r(u).$$

The distance of a node $v$ to $t$, denoted $d(v)$, is defined as the minimum length value of all paths getting from $v$ to $t$. In fact, it's easy to see that in this setting all $v_i, v_i'$ have $d(v_i) = d(v_i') = 6 + (k-1)(6 + x - x) = 6k$, and if variable $i$ is true then $d(u_i) = 6k$, otherwise $d(u_i') = 6k$.

(1) node s will go $a_{1,1}$ because (12) ($d(a_{1,1}) \leq \beta x + g_k - x + d(v_k) = x + g_k - x + 6k \leq r_t$).
(2) $a_{i,1}$ will go to $a_{i,2}$: The path that goes through $a_{i,2}$ and then $v_k$(not losing generality, suppose $v_k$ is true) has length value $x + g_k - x + d(v_k) = x + g_k - x + 6k$, and the path that goes through $b_i$ has length value $h_k + 6(k-1)$. So we derive this from (10) and (11).
(3) $a_{i,j}, j < l$ will go to $a_{i,j+1}$, because (10) and (14) (The path that uses $a_{i,j}$ to $v_k$ has length value $g_k\beta^{-1} + 6k$).
(4) $a_{i,l}, i < m$ will go to $a_{i+1,1}$, because the path from $a_{i+1,1} \to a_{i+1,2} \to v_k \to t$ has length value $\beta x + g_k - x + 6k$. So we derive this from (12) and (14).
(5) $a_{m,l}$ will go to $c_1$, because (13).
(6) $c_i$ will go to $c_{i+1}$ for $1 \leq i \leq n-1$, because (13).

(7) $c_n$ will go to $u_n^*$ (here $u_n^* = u_n$ if variable $n$ is true, otherwise $u_n^* = u'_n$) because (13).
(8) The agent then travels from $u_n^*$ to $t$ , which is trivial by easy computation on the weight values.

□

### 5.6. Satisfiability

PROPOSITION 5.6. *If the 3-CNF is satisfiable, then $G$ has a minimum total reward of at most $nx + r_t$.*

PROOF. First notice that in the minimum reward setting the agent cannot traverse through any $b_i$ to $u_k$, since otherwise it requires at least $\beta^{-1}h_k + 6(k-1) > r_t + nx$ (by (6)) rewards.

On the other hand, in a minimum setting the edge $(a_{i,j}, v_k)$ also cannot be used, since otherwise it requires at least $\beta^{-1}g_k + 6k > r_t + nx$ (by (7)) rewards.

Combine the two facts above, we know that the agent will reach $c_1$. Notice that all path from $c_1$ to $t$ has the form $c_1c_2 \cdots c_n u_n^* v_n^* u_{n-1}^* \cdots v_1^* t$, where $u_i^* \in \{u_i, u'_i\}, v_i^* \in \{v_i, v'_i\}$. On $c_1$, the agent evaluates the cost of any of these paths (excluding the rewards put) to be $6\beta^{-1} + 6(n-1) + 6n = r_t + nx$ (by (13)), so at least $r_t + nx$ rewards has to be put on nodes $\{c_1, c_2, ..., c_n, u_n, \cdots, u_1, u'_n, \cdots, u'_1, v_n, \cdots, v_1, v'_n, \cdots, v'_1, t\}$, in all three versions of $\mathbf{fmtr}_\beta$. And so if if $G$ has a minimal reward of at most $r_t + nx$, no reward is put at nodes $a_{i,j}$ or $b_i$. Also in $\mathbf{fmtr}_\beta$ III, no negative rewards can be given. consider the path that the agent chooses the shortest on $c_1$, we know that there must be a path from $c_1$ to $t$ that has a total reward (including reward on $t$) of $r_t + nx$. Then since there's no path from $v_k$ to $v'_k$, $u_k$ to $u'_k$ (vice versa), we can't put rewards on both $v_k$ and $v'_k$ or $u_k$ and $u'_k$.

Consider the following arrangement: if node $v_k$ has rewards on it, let variable $v_k$= true; otherwise let variable $v_k$= false. In the following, we prove this arrangement gives a valid arrangement of the original 3-SAT clause.

Note that if $v_k$ has rewards, then $v'_k$ doesn't have; vice versa. So to prove this proposition we only need to prove that for every clause $i$, the 3 $v$ kind nodes (call them $v_{k_1}^*, v_{k_2}^*, v_{k_3}^*$, $v_k^* \in \{v_k, v'_k\}$) that $a_{i,2}$ connects to has at least one of them has costs. To prove this, first notice that the shortest path from $a_{i,2}$ to $t$ must be through some $v_{k_j}^*$: The path through $a_{i,j} (j > 2)$ and then to $v_{k_j}^*$ has same costs but an additional $\beta x$ cost; and the path from $a_{i+1,1}$ has cost at least $\beta x(l-1) > nx + r_t$ (by (8) and that $a_{i,j}$ doesn't have reward), which is larger than the total sum reward.

If none of $v_{k_1}^*, v_{k_2}^*, v_{k_3}^*$ has reward on it, then we have

$$d(a_{i,2}) = \min_{1 \le j \le 3, v_{k_j}^*} g_{k_j} + d(v_{k_j}^*) = \min_{1 \le j \le 3} \min\{g_{k_j} + 6 + d(u_{k_j-1}), g_{k_j} + 6 + d(u'_{k_j-1})\} := S.$$

So at $a_{i,1}$ the edge to $a_{i,2}$ is evaluated as $\beta x \cdot \beta^{-1} + S = x + S$, since $a_{i,1}$ does not have reward. On the other hand, the path to $b_i$ is evaluated as (also, $b_i$ does not have reward)

$$\begin{aligned} 0 + d(b_i) &= \min_{1 \le j \le 3} \min\{h_{k_j} + d(u_{k_j-1}), h_{k_j} + d(u'_{k_j-1})\} \\ &< \min_{1 \le j \le 3} \min\{g_{k_j} + x + 6 + d(u_{k_j-1}), g_{k_j} + x + 6 + d(u'_{k_j-1})\} (\text{by}(9)) \\ &= x + S. \end{aligned}$$

So the agent will go to $b_i$, which makes contradiction with above. So $v_{k_1}, v_{k_2}, v_{k_3}$ will have at least one of them has rewards. Since it's impossible that both $v_k$ and $v'_k$ has rewards, we proved the proposition. □

Combine the two sides above, we prove the main theorem.

## 6. CONCLUSION AND FURTHER WORK

So far we have extended the work of [Kleinberg and Oren 2014]. Firstly, we have shown the close relationship between the special structure $\mathcal{F}_k$ and exponential cost ratio, thus we should avoid this structure when we design the way of work if we do not want to waste due to time-inconsistency. Also, we have shown the hardness of deleting edges or distributing the reward to motivate the agent. This may be a bad properties of the motivation model.

As an extension of our work, we can consider other forms of the problem, for example, for **fmtr**$_\beta$, we can give the following extensions:

**fmtr**$_\beta$ IV: We can put rewards on every edge and node of the graph. In this situation, the agent gets the reward directly when he travels through it, instead of multiplying it by $\beta$;

**fmtr**$_\beta$ V: We can put any reward on any node, but only pays those reward that the agent actually claimed.

It's in fact easy to see that **fmtr**$_\beta$ IV is in $\mathcal{P}$: we can simply find the shortest path(without the agent) of the graph, and put on each edge the cost of this edge. This configuration have total cost the same as the shortest path, which is of course the smallest possible.

On the other hand, **fmtr**$_\beta$ V seems to be NPC. This problem is beyond the scope of our method; we can put rewards on both $v$ and $v'$ nodes. But maybe with slight changes of the model, we can still prove this hardness.

Besides extensions of the problem, it's also beneficial to explore approximation algorithms of this problem. Also, practical algorithms that works for reality problems are also a good exploration. We can also consider the case of multiple agents who cooperate to get the target node.

## REFERENCES

AKERLOF, G. A. 1991. Procrastination and obedience. *The American Economic Review*, 1–19.

KLEINBERG, J. AND OREN, S. 2014. Time-inconsistent planning: a computational problem in behavioral economics. In *Proceedings of the fifteenth ACM conference on Economics and computation*. ACM, 547–564.