Algorithm 1 ValH1
1: Input: IDP-MDP $M$, state $B_{i,j}$, matrix $V$ of horizon-values, matrix $\pi$ of horizon-policy, $h$
2: if $h = 0$ then
3: $V^h(B_{i,j}) = 0$ \{no more time steps\}
4: else
5: for $k = i, j - 1$ do
6: if $V^{h-1}(B_{k,k}) = 0$ then
7: $[V^{h-1}(B_{i,j}), V, \pi] = \text{ValH1}(M, B_{i,j}, V, \pi, h-1)$ \{Calc. value if agent accepts $\delta_k$\}
8: end if
9: if $V^{h-1}(B_{k+1,j}) = 0$ then
10: $[V^{h-1}(B_{k+1,j}), V, \pi] = \text{ValH1}(M, B_{k+1,j}, V, \pi, h-1)$ \{Calc. value if agent rejects $\delta_k$\}
11: end if
12: $Q^h(B_{i,j}, \delta_j) = p(B_{i,j} | B_{i,j}, \delta_k) + p(B_{i,j} | B_{i,j}, \delta_k) + V^{h-1}(B_{k,k}) + V^{h-1}(B_{k+1,j})$
13: end for
14: $Q^h(B_{i,j}, \delta_j) = \min Q^h(B_{i,j}, \cdot)$
15: $V^h(B_{i,j}) = \min Q^h(B_{i,j}, \cdot)$
16: $\pi^h(B_{i,j}) = \arg\min Q^h(B_{i,j}, \cdot)$
17: end if
18: Return $[V^h(B_{i,j}), V, \pi]$

state $B_{i,j}$, and a given offered incentive $\delta_k$, with probability $\sum_{i \leq l \leq k} B_{i,j}(l)$ the agent will accept the offer and the new state will be $B_{i,k}$, and with probability $1 - \sum_{i \leq l \leq k} B_{i,j}(l)$ the agent will reject the offer, and the new state will be $B_{k+1,j}$. The cost model of the MDP takes the form $c(B_{i,j}, \delta_k, B_{k,k}) = \delta_k + c_1$ for transitions where the agent accepts the incentive, and $c(B_{i,j}, \delta_k, B_{k+1,j}) = c_2$ when the agent rejects.

We now show the special structure of this problem allows us to introduce more efficient planning algorithms that require only $O(K^3)$ and $O(\min(H, K)K^3)$ computation time for the infinite and finite horizon case, respectively. First note that the principal only needs to consider offering incentives $\delta_k$ within the range of $i \leq k \leq j$ for a state $B_{i,j}$, as other incentives will be automatically rejected (or accepted but at higher cost than necessary for the principal). We start by considering the infinite horizon case. Let the expected sum of discounted costs of state $B_{i,j}$ be $V(B_{i,j})$. We define a MDP state $s$ to be non-recurring with respect to a policy $\pi$ if the state can only be reached once while executing the $\pi$. A state $s$ of the MDP is called self-absorbing if executing $\pi$ from $s$ results in a self-transition to state $s$ with probability 1. A policy $\pi$ is NonRecur-Absorb if every MDP state is either non-recurring or self-absorbing with respect to $\pi$.

First recall that for an infinite horizon MDP, there always exists an optimal policy that is stationary (the decision depends on only the current state and is independent of the time step). Therefore we will consider only stationary policies. Next note that for an infinite horizon IDP-MDP, if incentive $\delta_j$ is offered in state $B_{i,j}$, then the MDP remains in same state $B_{i,j}$. If one of the other possible actions (offering incentives $\delta_i, \ldots, \delta_{j-1}$) is taken, then the MDP will transition to a new state, and will never return to the state $B_{i,j}$ since the number of non-zero entries in the probability vector $B_{i,j}$ monotonically decreases. This implies all infinite-horizon policies for an IDP-MDP are NonRecur-Absorb. As each state can only be reached once (or is an absorbing state) during execution, we only need to compute a Bellman backup for the state-action values $Q$ once for each state-action pair. We compute a decision policy by computing the state-action values of the initial state by recursively computing the value of each reachable state. The values of subsequent states are stored and cached so that they can be re-used if the same state is reached through a different trajectory of offered incentives and responses. There are $O(K^2)$ MDP states and $O(K)$ actions for each state, so there are $O(K^3)$ state-action pairs. The cost of computing the Bellman backup for a given state-action pair can be done in constant time given the values of the possible next states (since there are only at most 2 possible next states). Therefore the algorithm takes $O(K^3)$ time. Note this is significantly faster than value iteration on generic MDPs with $N$ states and $N_a$ actions which requires $O(N_a^N)$ time to compute a near-optimal policy.

For the finite horizon case, let $V^h(B_{i,j})$ denote the expected sum of costs to the principal for state $B_{i,j}$ for a horizon of $h$ future interactions with the agent. Every policy is not NonRecur-Absorb because the principal could offer the same incentive for multiple time steps (staying in the same state) and then offer a new incentive and transition to another state. However, we can prove the following result:

**Theorem 5.2.** There exists an optimal policy for an IDP-MDP with $N = 1$ that is NonRecur-Absorb.

The proof intuition is that we can always re-order an optimal policy so that it only offers incentives that keep the MDP in the same state at the final time steps. The algorithm for the finite horizon case is displayed in Algorithm 1. Here the state’s value and policy will depend on the time step. Each time the horizon decreases, either a state becomes self-absorbing with a cost of $\delta_j h$ or least one non-zero element of the probability vector $B_{i,j}$ becomes zero. Since the initial state $B_{1,k}$ has $K$ non-zero elements, it will take at most $K$ time steps until a state is reached.