Navigating the Difficulty of Achieving Global Optimality under Variance-Induced Time Inconsistency

Anonymous Author(s)

ACM Reference Format:
Anonymous Author(s). 2024. Navigating the Difficulty of Achieving Global Optimality under Variance-Induced Time Inconsistency. In *Proceedings of International Conference on AI in Finance (ICAIF '24).* ACM, New York, NY, USA, 4 pages.

A AGENTS

Following, we provide further details on the MV agents. Full implementation can be found in the supplementary[1].

A.1 EPG

Both the trueEPG and approxEPG in this paper were modified from [4] with specifications made according to our environments. The objective (6) in the main paper is parallel to f_λ in [4, Section 4.2],

$$f_\lambda \coloneqq J(x^*) - \lambda g(V(x^* - b)) \tag{1}$$

For our context, we set the function $g(\cdot)$ in (1) to be the identity function $g(x) = x$, $b = 0$, x^* replaced by x_0, and J, V referred to as J, K. With these specifications, the updates in [4, Eq. (13)] become the updates in (10)-(12) in our work. Another distinction between our work and [4] is that we deal with a finite horizon problem, while they deal with an infinite horizon problem. Our finite-horizon modification is realized by making the given initial state x_0 a recurrent state x^* and setting the transition probability from any state to x_0 at time $t = T$ to 1. As such, our modification will not affect the theoretical properties of the original EPG algorithm.

The equations (8) and (9) in our paper are parallel to [4, Lemma 4.2]. In particular, we can unpack the expectation of cumulative reward as the sum of all possible cumulative rewards weighted by the probability of all possible trajectories. Then, $\nabla J(x)$ can be expressed as follows,

$$\begin{aligned}
\nabla J(x) &= \nabla \mathbb{E}[\mathbf{R}_0^{\pi_\theta}(X_0)|X_0 = x] \\
&= \nabla \sum_{\xi} P[\xi|X_0 = x]\mathbf{R}_0^{\pi_\theta}(X_0;\xi) \\
&= \sum_{\xi} \mathbf{R}_0^{\pi_\theta}(X_0;\xi)\nabla P[\xi|X_0 = x] \\
&= \sum_{\xi} \left[\mathbf{R}_0^{\pi_\theta}(X_0;\xi)\nabla P[\xi|X_0 = x]\frac{P[\xi|X_0 = x]}{P[\xi|X_0 = x]}\right] \\
&= \sum_{\xi} P[\xi|X_0 = x]\mathbf{R}_0^{\pi_\theta}(X_0;\xi)\nabla \log P[\xi|X_0 = x] \\
&= \mathbb{E}\left[\mathbf{R}_0^{\pi_\theta}(X_0;\xi)\nabla \log P[\xi|X_0 = x]\right]
\end{aligned}$$

Since $P[\xi|X_0 = x] \coloneqq \Pi_{t\in\mathcal{T}} P[x_{t+1}|x_t, a_t]\mu_\theta(a_t|x_t)$ and the transition probability is independent of θ, $\nabla \log P[\xi|X_0 = x] = \sum_{t\in\mathcal{T}} \nabla \log \mu_\theta(a_t|x_t)$. The formula for $\nabla K(x)$ can be obtained similarly. Note that $\nabla J(x)$ and $\nabla K(x)$ derivations follow directly from [4] and the change related to our finite-horizon problem is fully absorbed by the trajectories ξ.

The updates (10) and (11) in our paper are parallel to the 1st-2nd row of [4, Eq. (13)]; they are iterative estimation schemes with targets obtained from the definitions of J^θ, K^θ in Section 5,

$$\begin{aligned}
J^\theta(x_0) &= \mathbb{E}[R_0^{\pi_\theta}(x_0)] \\
K^\theta(x_0) &= Var[R_0^{\pi_\theta}(x_0)] \\
&= \mathbb{E}[(R_0^{\pi_\theta}(x_0))^2] - \mathbb{E}([R_0^{\pi_\theta}(x_0)])^2 \\
&= \mathbb{E}[(R_0^{\pi_\theta}(x_0))^2] - (J^\theta(x_0))^2
\end{aligned}$$

Update formula (12) is obtained by substituting equations (8) and (9) into formula (7).

Our two EPG versions, i.e., trueEPG and approxEPG, are summarized through Algorithm 1. For our implementation, we set $\mu_\theta(a|x) \coloneqq \frac{\exp(-\theta^T\phi(x,a))+\varepsilon}{\sum_{\tilde{a}}\exp(-\theta^T\phi(x,\tilde{a}))+\varepsilon|\mathcal{A}|}$ following [4], with small $\epsilon = 0.001$. To match the tabular setup in SPERL algorithm, $\phi(x, a)$ is a one-hot representation with $\theta \in \mathbb{R}^{|\mathcal{X}|\times|\mathcal{A}|}$.

Specifically, the expectations in this EPG method can be computed by the law of total probability, by considering all possible

Algorithm 1 MVarEPG: Mean-Variance Globally Optimal Control

```
**Input:** MDP simulator, $\lambda$, $T$, learning rates $\alpha, \beta$
**Output:** Converged policy parameter, $\theta$.
Initialize $\theta$ and $J, K$ (for approxEPG)
**while** $\theta$ not stable **do**
    Sample trajectories $x_0$,$A_0$,. . .,$X_{T-1}$,$A_{T-1}$,$X_T \sim \boldsymbol{\pi}_\theta$
    Update $\theta$ with (7) (for trueEPG)
    Update $J, K, \theta$ with (10)-(12) (for approxEPG)
**end while**
```

[1] For EPG variants and SPERL, initializations, learning rates, and trajectory generations are tuned according to our specific environments; please refer to run-main.py.

ICAIF '24, November 14–16, 2024, Brooklyn, NY

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM. . . $15.00

trajectories ξ starting with $X_0 = x$,

$$\begin{aligned}
J(x) &= \mathbb{E}\left[\mathbf{R}_0^{\boldsymbol{\pi}\theta}(X_0)|X_0 = x\right] \\
&= \sum_{\xi} P[\xi|X_0 = x]\mathbf{R}_0^{\boldsymbol{\pi}\theta}(x;\xi) \\
K(x) &= \mathbb{E}\left[(\mathbf{R}_0^{\boldsymbol{\pi}\theta}(X_0))^2|X_0 = x\right] \\
&= \sum_{\xi} P[\xi|X_0 = x_0](\mathbf{R}_0^{\boldsymbol{\pi}\theta}(x;\xi))^2 \\
\nabla J(x) &= \mathbb{E}\left[\mathbf{R}_0^{\boldsymbol{\pi}\theta}(X_0)\sum_{t=0}^{T-1}\nabla\log\mu_\theta(A_t|X_t)\middle|X_0 = x\right] \\
&= \sum_{\xi} P[\xi|X_0 = x_0]\mathbf{R}_0^{\boldsymbol{\pi}\theta}(x;\xi)Z^\theta(\xi) \\
\nabla K(x) &= \mathbb{E}\left[(\mathbf{R}_0^{\boldsymbol{\pi}\theta}(X_0))^2\sum_{t=0}^{T-1}\nabla\log\mu_\theta(A_t|X_t)\middle|X_0 = x\right] \\
&\quad - 2J(x)\nabla J(x) \\
&= \sum_{\xi} P[\xi|X_0 = x_0](\mathbf{R}_0^{\boldsymbol{\pi}\theta}(x;\xi))^2 Z^\theta(\xi) \\
&\quad - 2J(x)\nabla J(x)
\end{aligned}$$

with $Z^\theta(\xi) = \sum_{t=0}^{T-1}\nabla\log\mu_\theta(A_t|X_t; X_0 = x, \Xi = \xi)$. Then, the update on policy parameter θ can be computed by

$$\theta_{k+1} = \theta_k + \alpha_k(\nabla J(x_0) - \lambda\nabla K(x_0)).$$

It was suggested by [4] that a decreasing learning rate is needed to ensure convergence. However, in trueEPG, we found that decreasing the learning rate naively can cause the algorithm to be stuck in poor local optima in the experiment. Therefore, we adopt the following two-phase scheduling of learning rates in our trueEPG implementation. In the first phase, the learning rate is set to be a small value and would be increased (e.g., doubled) when the change in initial value is too small after one update. Since the agent has access to the environment parameters, it can compute the change in initial value accurately. Empirically, such an increase in learning rate would boost the performance significantly. When there is no more improvement after increasing the learning rate, it enters the second phase. In this phase, the learning rate is kept constant by default. Unless when the improvement of the initial value is negative as compared to the previous iteration, the current update will then be skipped and the learning rate will be set to a lower value (e.g., halved).

A.2 SPERL

In our PE step, we applied a similar technique to [3, 5], differing in that we are predicting a finite-horizon Q-function instead of an infinite-horizon value function. Correspondingly, note that Proposition 2 is a finite-horizon, tabular, Q-function version of [5, Proposition 2]. To account for some environment setups, rewards may also depend on x', in which case Proposition 2 can be slightly adjusted as follows,

$$\begin{aligned}
J_t^{\boldsymbol{\pi}}(x,a) = \sum_{x'\in\mathcal{S}} P^a[t+1,x'|t,x]\,(r_t(x,a,x') \\
+\gamma J_{t+1}^{\boldsymbol{\pi}}(x',\pi_{t+1}(x'))) \\
M_t^{\boldsymbol{\pi}}(x,a) = \sum_{x'\in\mathcal{S}} P^a[t+1,x'|t,x]\,\big(r_t^2(x,a,x') \\
+\gamma^2 M_{t+1}^{\boldsymbol{\pi}}(x',\pi_{t+1}(x')) \\
+2\gamma r_t(x,a,x')J_{t+1}^{\boldsymbol{\pi}}(x',\pi_{t+1}(x'))\big)
\end{aligned}$$

Our PE extends the PE step in [2] (which handles the variance of terminal wealth) to account for the variance of accumulated rewards. Algorithm 2 summarizes SPERL's PE method.

Our PI step draws directly on [2]'s characterization of equilibrium by greedy action selection, summarized in Algorithm 3.

Remark. The convergence of Algorithm 3 to equilibrium is guaranteed by [2] when line 8-9 implemented with full sweeps over all possible x, a, x', a'. Without full sweeps (in the sample-based case), convergence results are only present for PE by [5]. As both results do not extend trivially to our case, the convergence of MVarSPERL remains open.

Algorithm 2 MVarPE: Mean-Variance Policy Evaluation

```
1: Input: MDP simulator, $\lambda$, fixed policy $\boldsymbol{\pi}$, learning rate $\alpha_J, \alpha_M$
2: Output: Q-function estimates $Q_t(x,a), \forall t, x, a$
3: Initialize $J(t,x,a), M(t,x,a) \leftarrow 0, \forall t < T, x, a$
4: Set $J(T,x,\cdot), M(T,x,\cdot) \leftarrow r_T(x), r_T^2(x), \forall x$
5: for $(t,x,a,t+1,x',a') \sim \boldsymbol{\pi}$ do
6:     Compute $\delta_J(t,x,a), \delta_M(t,x,a)$ by (16)
7:     $J(t,x,a) \leftarrow J(t,x,a) + \alpha_J\delta_J(t,x,a)$
8:     $M(t,x,a) \leftarrow M(t,x,a) + \alpha_M\delta_M(t,x,a)$
9:     $Q_t(x,a) \leftarrow J(t,x,a) - \lambda\left(M(t,x,a) - J(t,x,a)^2\right)$
10: end for
```

Algorithm 3 MVarSPERL: Mean-Variance Equilibrium Control

```
1: Input: MDP simulator, $\lambda$, horizon $T$, learning rates $\alpha_J, \alpha_M$
2: Output: Approximate Equilibrium $\boldsymbol{\pi}$
3: Initialize $\boldsymbol{\pi}, J, M$
4: while $\boldsymbol{\pi}$ not stable do
5:     Sample trajectories $x_0$,$A_0$,. . .,$X_{T-1}$,$A_{T-1}$,$X_T \sim \boldsymbol{\pi}_\theta$
6:     Update $J(T,X_T,\cdot), M(T,X_T,\cdot) \leftarrow r_T(X_T), r_T^2(X_T)$
7:     for $t \leftarrow T-1$ to 0 do
8:         Set $x = X_t, a = A_t, x' = X_{t+1}, a' = A_{t+1}$
9:         Compute $Q_t(x,a)$ by Algorithm 2, line 6-9
10:        if $Q_t(x,\pi_t(x)) > Q_t(x,a)$ then
11:            Update $\pi_t(x) \leftarrow \arg\max_a Q_t(x,a)$
12:        end if
13:    end for
14: end while
```

B EMPIRICAL STUDY

B.1 Environment Specifications

B.1.1 Portfolio Management (PM). Our first environment is derived from the portfolio management problem in [4], where an investor, needs to decide on her proportions of investment into two types of assets: one liquid risk-free asset and one non-liquid risky asset, at each time t over a fixed horizon T. A liquid asset has a fixed interest rate r^l and can be sold at any time $t \in \mathcal{T}$, while a non-liquid asset has a time-varying interest rate $r_t^{nl} \in \{\underline{r}^{nl}, \bar{r}^{nl}\}$, with a switching probability $p_{\text{switch}} := P[r_{t+1} \neq r_t | r_t]$, and can only be sold after a fixed period $N \in \mathbb{Z}^+$, with a default probability $p_{\text{risk}} := P[\mathbb{1}_{\text{def, t}} = 1]$.

At each time t, the state $x_t \in [0,1]^{N+1} \times \{\underline{r}^{nl}, \bar{r}^{nl}\}$ captures respectively the current (1-dim) proportion of investment in the liquid asset, (N-dim) proportions of investment in the non-liquid assets that are to mature after $1, 2, \cdots, N$ time-steps, and non-liquid interest r_t^{nl}. At the initial time $t = 0$, all investments are put in the liquid asset. Then, at each time t, the action $a_t \in \{1, 0\}$ determines whether to invest a pre-determined proportion $\kappa \in (0,1)$ of the investor's total cash (obtained by selling the liquid asset) into the non-liquid asset or do nothing and drive the following transitions,

- the liquid asset position will gain interest and gain/loss value from defaults and current investments:
$$\tilde{x}_{t+1,1} = (1 + r^l)(x_{t,1} + x_{t,2}(1 - \mathbb{1}_{\text{def},t}) - a_t\kappa).$$
- the illiquid asset positions will gain interest and get closer to maturity:
$$\tilde{x}_{t+1,i} = \begin{cases} (1 + r_t^{nl})x_{t,i+1}, & \text{if } 2 \leq i \leq N \\ (1 + r_t^{nl})a_t\kappa, & \text{if } i = N + 1 \end{cases}$$
- the investment proportions will follow accordingly, as we let them sum up to 1:
$$x_{t+1,i} = \frac{\tilde{x}_{t+1,i}}{\sum_{j=1}^{N+1} \tilde{x}_{t+1,j}}, \forall i = 1, 2, \ldots, N+1.$$

To maintain positive balances at all times ($x_{t,1} \geq 0, \forall t$), we let $a_t = 1$ only if there are sufficient liquid assets to fund the investment. Rewards are defined as the percentage increase in total wealth, i.e., $R(x_t, a_t) := \sum_{i=1}^{N+1}(\tilde{x}_{t+1,i} - x_{t,i})$.

For our experiments, we set the investment horizon $T = 5$ and correspondingly, $N = 3$. To construct cube-PM, we consider various values of default probability $p_{risk} \in \{0.05, 0.25, 0.45, 0.65, 0.85\}$, interest rate switching probability $p_{switch} \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$, and illiquid asset's interest lower bound $\underline{r}^{nl} \in \{0.75, 0.80, 0.85, \cdots, 1.10\}$. The two chosen specifications for Table 1 are then (PM-1) with $\underline{r}^{nl} = 0.75$, $p_{\text{switch}} = 0.2$, $p_{risk} = 0.3$, and (PM-2) with $\underline{r}^{nl} = 1.1$, $p_{\text{switch}} = 0.4$, $p_{\text{risk}} = 0.7$, with fixed $r^l = 1, \bar{r}^{nl} = 2, \kappa = .2$. Due to our shorter horizon $T = 5$, we have adjusted the specifications from [4] to control the environment riskiness s.t. all algorithms will not end up in trivial policies (e.g., $a_t(x) = a_{t'}, \forall t \neq t'$).

B.1.2 Optimal Execution (OE). Our second environment is derived from the optimal execution problem with market impact in [1], where a liquidation trader needs to decide on how to pace selling Y shares of one stock within a fixed horizon T. The stock price S_t is modeled to be *temporally* and *permanently* affected by our selling actions (trade size at each time t). In particular, the permanent price S_t follows the dynamics, $S_t = S_{t-1} + \sigma\tau^{\frac{1}{2}}W_t - \tau g\left(\frac{n_t}{\tau}\right)$, where σ is the stock volatility, W_t is a random variable following a discretized $\mathcal{N}(0,1)$, with support $\mathcal{I} := \{-3, -1, 1, 3\}$ and $P[W_t = i] = \Phi(i+1) - \Phi(i-1), \forall i \in \mathcal{I}$, and $g : \mathbb{R} \to \mathbb{R}^+$ is a permanent impact function of the average trading rate $\frac{n_t}{\tau}$, with n_t the number of shares sold at time t and τ the length of one time-step. The actual price $\tilde{S}_t$ is then modeled to follow $\tilde{S}_t = S_t - h(\frac{n_t}{\tau})$, with $h : \mathbb{R} \to \mathbb{R}^+$ a temporal impact function of the average trading rate. In our work, the market impact functions are specified linearly as $g(v) := \zeta v$ and $h(v) := \rho\, sign(v) + \eta v$, following [1], where ζ and η indicate the strength of permanent and temporary market impact and ρ reflects the fixed cost of a transaction (we set $\rho = .0625$).

At each time t, the state $x_t \in \mathbb{R} \times [0,1]^2$ captures the log of relative stock price at time t w.r.t the initial stock price $\log \frac{S_t}{S_0}$, the proportion of time remaining $\frac{t}{T}$, and the proportion $\frac{y_t}{Y}$ of held shares; in our implementation, state space is discretized into a finite state space. At each time t, actions capture the proportion of remaining shares to sell, i.e., $a_t \in \{0, .1, \cdots, .9, 1.0\}$, such that $n_t = \frac{a_t y_t}{Y}$. At $t = T - 1$, a_t is set to 1. to ensure all held shares are liquidated in time. Rewards are set to $\frac{\tilde{S}_t - S_0}{S_0}\frac{n_t}{Y}$[2]. We assume no buying actions and require all shares sold by the time $t = T$, i.e. $y_T = 0$.

For our cube-OE experiments, we consider various values of permanent impact strength $\zeta \in \{x \times 10^{-8} | x = 15, 20, 25, 30, 35\}$, temporary impact strength $\eta \in \{x \times 10^{-7} | x = 15, 20, 25, 30, 35\}$, and volatility $\sigma \in \{0.015, 0.020, 0.025, 0.030\}$. The two chosen specifications for Table 2 are then (OE-1) with $\sigma = 0.029$, $\tau = 1$, and (OE-2) with $\sigma = 0.015$, $\tau = 1$, while fixing $Y = 10^6$ as in [1]. As in PM, we set the horizon $T = 5$. In Figure 6, the column $t = 4$ is omitted as actions at $t = 4$ are fixed to 1.0 by problem definition.

B.2 Cube-PM and Cube-OE

Results: The results of these experiments are consistent with the main paper. In particular, we observe a pattern about the difference in initial values in the graphs of ΔV for each environment (Figure 3 and 4). The PM environments are simple enough for trueEPG to approach global optimality.Hence, trueEPG has higher initial values than SPE (represented by a blue dot in Figure 3). In some other environment specification where SPE coincides with the global optimum (i.e., weak-TC), SPE can be seen to perform as good as trueEPG (represented by a grey dot in Figure 3). The resulting figure agrees with this hypothesis. Although there are some exceptions where true EPG policies fail to produce an optimal value at time $t = 0$ (represented by orange dots in Figure 3), it can be shown that they happened by chance due to the stochastic nature of EPG as the orange dots would turn grey when the experiments are conducted with another random seed.

To elaborate on the pattern in PM, environments with higher p_{switch}, lower p_{risk}, and lower $\underline{r}^{nl}$ occupy the strong-TIC regions, driving trueEPG's outperformance of trueSPE. Whereas in OE environments, departing from trueEPG and trueSPE's inconclusive attainment of global optimality, larger temporary market impact strength η can be seen to drive larger trueSPE's outperformance of

[2] Note that negative rewards should be expected due to the temporal and permanent impacts tendency to lower the actual price $\tilde{S}_t$ (or cash received during selling), resulting in liquidation costs. Moreover, by cumulative rewards' definition as total liquidation costs over the initial value of shares with specific environment setups (used in practice), the magnitude of cumulative reward is expected to be small (in the order of 1e-2).

trueEPG. The effect of the remaining two parameters is however not significant.

REFERENCES

[1] Robert Almgren and Neil Chriss. 2001. Optimal Execution of Portfolio Transactions. *Journal of Risk* 3, 2 (2001), 5–39.

[2] Nixie Lesmana and Chi Seng Pun. 2021. A Subgame Perfect Equilibrium Reinforcement Learning Approach to Time-inconsistent Problems. *SSRN Electronic Journal* (01 2021). https://doi.org/10.2139/ssrn.3951936

[3] Matthew J. Sobel. 1982. The Variance of Discounted Markov Decision Processes. *Journal of Applied Probability* 19, 4 (Dec. 1982), 794–802. https://doi.org/10.2307/3213832

[4] Aviv Tamar, Dotan Di Castro, and Shie Mannor. 2012. Policy Gradients with Variance Related Risk Criteria. In *Proceedings of the 29th International Coference on International Conference on Machine Learning* (Edinburgh, Scotland) *(ICML'12)*. Omnipress, Madison, WI, USA, 1651–1658.

[5] Aviv Tamar, Dotan Di Castro, and Shie Mannor. 2016. Learning the Variance of the Reward-To-Go. *Journal of Machine Learning Research* 17, 13 (2016), 1–36. http://jmlr.org/papers/v17/14-335.html