Differentiable Planning for Optimal Liquidation

Alan A. Pennacchio, Leliane N. de Barros, Denis D. Mauá

Department of Computer Science – University of São Paulo
São Paulo, SP – Brazil

{pennacchio,leliane,ddm}@ime.usp.br

Abstract. *Optimal liquidation consists of selling large blocks of single stocks within given time frames optimally with respect to specified risk-sensitive objectives. In this paper, we extend the Almgren-Chriss model for the liquidation process to a more generic and realistic setting and present a differentiable planning algorithm to solve it. We evaluate the performance of the proposed method through experiments, demonstrating the potential of differentiable planning for optimal liquidation in realistic scenarios.*

Keywords: *Differentiable Planning, Risk-sensitive Objective, Markov Decision Process, Optimal Liquidation.*

1. Introduction

A decision problem that financial institutions often face concerns selling large blocks of single stocks. If the sale is executed too rapidly, it may move the market in an adverse direction and thus decrease the total amount received, while, on the other hand, if it's executed too slowly, there's the risk of the stock dropping in price before it's completed. This can be viewed as a sequential decision making problem for which the financial institution must specify a liquidation strategy that adheres to its risk tolerance while maximizing the sale proceeds.

On a high-level basis, automated planning refers to the subfield of AI that is concerned with sequential decision making under descriptive models of systems or processes. One particular subset of its methods that has been attracting increasing interest recently is differentiable planning [Wu et al. 2017, Bueno et al. 2019], mostly due to its applicability to continuous state-action spaces and wide success of differentiable models in other domains, such as computer vision and natural language processing [Bueno 2021].

The main contribution of this paper is demonstrating the potential of differentiable planning algorithms for optimal liquidation in realistic scenarios. Specifically, we extend the well-known Almgren-Chriss model for the liquidation process [Almgren and Chriss 2001] to a more generic and realistic setting where the stock price process follows a geometric Brownian motion with constant drift and local volatility [Dupire et al. 1994, Derman and Kani 1994]. We then show how this model can be naturally viewed as an exogenous Markov decision process (X-MDP) [Bueno 2021] and propose a differentiable planning algorithm to solve it. We conclude by analyzing the performance of the proposed method through experimental results.

The remainder of this paper is structured as follows. Section 2 describes the optimal liquidation problem and formally defines the model used throughout the paper. Section 3 briefly reviews X-MDPs and how the optimal liquidation problem can be framed as one, finalizing by presenting the differentiable planning algorithm proposed. Section 4 presents the experiments performed to evaluate the performance of the proposed method. Finally, Section 5 concludes the paper and points to future research directions.

2. Optimal Liquidation

In brief terms, the optimal liquidation problem consists of selling a (presumably) large block of a single stock within a given time frame following a strategy that is optimal with respect to our risk preferences, balancing between moving the market in an adverse direction (if the trade is executed too rapidly) and the market risk we're exposed to by holding the shares for too long.

In this section, we formally define this problem and present the components used throughout the rest of the paper to model it. Our specification is largely based on the Almgren-Chriss model [Almgren and Chriss 2001] and essentially extends it to a more generic and realistic setting at the cost of losing some of their analytical results.

2.1. Trading Strategy

Defining our scenario precisely, suppose we must liquidate $x_0 \in \mathbb{R}_{\geq 0}$ shares of a single stock by time $T \in \mathbb{R}_{>0}$. Divide this time frame into N intervals of length $\tau = \frac{T}{N}$, and let, for $k = 0, \ldots, N$, the discrete times be $t_k = k\tau$.

Our objective is to specify a strategy $x = (x_1, x_2, \ldots, x_N)$, where $x_k \in \mathbb{R}_{\geq 0}$ determines how many shares to hold at time t_k. A few constraints naturally follow: we must not buy additional shares, implying $x_j \geq x_k$ whenever $j \leq k$; since we must liquidate the entire block of shares, we have that $x_N = 0$.

As we shall see in Section 3, we'll represent this strategy by a parametrized policy whose parameters will be adjusted via differentiable planning. Having said that, you can consider it, for the purposes of this section, as a simple list of numbers.

2.2. Market Impacts

Following the Almgren-Chriss model [Almgren and Chriss 2001], our liquidation strategy impacts the stock prices both temporarily and permanently. Temporary impacts capture one-time transaction costs associated with single trades (e.g., bid-ask spread and fees), while permanent impacts refers to shifts in the equilibrium price of the stock caused by our trading.

Both types of impact are modeled by generic functions of the stock price and the number of shares sold on a given time period, which we constrain solely through the requirement for them to be differentiable in order for our problem to be amenable to the differentiable planning approach described in Section 3.

Specifically, if $n_k = x_{k-1} - x_k$ denotes the number of shares sold between times t_{k-1} and t_k, let $g \colon \mathbb{R}^2_{\geq 0} \to \mathbb{R}_{\geq 0}$ and $h \colon \mathbb{R}^2_{\geq 0} \to \mathbb{R}_{\geq 0}$ represent, respectively, the permanent and temporary impacts of selling those shares. A simple specification for these impacts, for instance, are the linear models $g(S_{k-1}, n_k) = \gamma n_k / \tau$ and $h(S_{k-1}, n_k) = \epsilon + \eta n_k / \tau$ parametrized by γ, ϵ and η, as presented on [Almgren and Chriss 2001].

As we'll see in the rest of this section, these impacts are factored into our model in different ways. Permanent impacts affect the stock price evolution through time, while temporary impacts affect only the effective price received for the sale of shares on specific time steps.

2.3. Stock Price Evolution

In contrast to the Almgren-Chriss model, which models the stock price process as a discrete-time arithmetic Brownian motion, we opt instead for a geometric Brownian motion. While it's well known that the latter is a more accurate (and the most widely used) model for stock prices evolution [Hull 2006], the former approximates it quite well for short liquidation periods T [Almgren and Chriss 2001].

For an initial stock price $S_0 \in \mathbb{R}_{\geq 0}$, a drift curve $\mu\colon (0, T] \to \mathbb{R}$ and local volatility $\sigma\colon (0, T] \times \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$, let $\mu_k = \mu(t_k)$ and $\sigma_k = \sigma(t_k, S_{k-1})$. Then, for $k = 1, \ldots, N$, the stock price evolution is given by

$$S_k = S_{k-1} e^{(\mu_k - \sigma_k^2/2)\tau + \sigma_k \sqrt{\tau} \xi_k} - g(S_{k-1}, n_k)\,,$$

where ξ_k are i.i.d. distributed according to a standard normal distribution and $g(S_{k-1}, n_k)$ is the permanent market impact of selling n_k shares between times t_{k-1} and t_k.

2.4. Capture and Cost of Trading

Between times t_{k-1} and t_k, we sell n_k shares of the stock and receive $n_k \tilde{S}_k$ in cash, where $\tilde{S}_k = S_{k-1} - h(S_{k-1}, n_k)$ is the effective price received per shares, given by the stock price at the start of the period discounted for the temporary impact of trading.

The capture of the strategy is defined as the present value of the cash payments received for sale of the shares. Let $r\colon (0, T] \to \mathbb{R}$ be a discount curve and $r_k = r(t_k)$, for $k = 1, \ldots, N$, then the capture of the strategy is given by $\sum_{k=1}^{N} e^{-r_k} n_k \tilde{S}_k$.

Finally, the total cost of trading of the strategy is defined as the difference between the value of our position at the start of the liquidation period and the capture of the strategy, being given by $C = x_0 S_0 - \sum_{k=1}^{N} e^{-r_k} n_k \tilde{S}_k$.

We can now restate our goal more precisely as choosing a strategy that minimizes the expected cost of trading $\mathbb{E}[C]$ such that the market risk we're exposed to, which we'll quantify as its standard deviation $\sqrt{\mathbb{V}[C]}$, is in line with our risk aversion. Specifically, we aim to minimize the loss given by $L = \mathbb{E}[C] + \lambda \sqrt{\mathbb{V}[C]}$, where $\lambda \in \mathbb{R}_{\geq 0}$ represents our intolerance to risk—the greater it is, the more risk-averse we are.

3. Differentiable Planning

Differentiable planning refers to a class of automated planning methods to solve sequential decision making problems in continuous state-action spaces via gradient-based optimization. The main motivation behind it stems from the difficulty that traditional search-based planning approaches have to handle problems with arbitrary non-linear dynamics and/or cost functions in continuous spaces [Bueno 2021].

In this section, we'll start by briefly describing exogenous Markov decision processes (X-MDPs) and how the optimal liquidation problem, as it was previously presented, naturally fits into one. To conclude it, we'll present the proposed differentiable planning algorithm to obtain a strategy for the optimal planning X-MDP.

3.1. X-MDP and Optimal Liquidation

Similarly to [Bueno 2021], we define an X-MDP by a tuple $(\mathcal{Y}, \mathcal{A}, \xi, f, c, y_0)$, where $\mathcal{Y} \subset \mathbb{R}^n$ is the state space, $\mathcal{A} \subset \mathbb{R}^m$ is the action space, ξ is a discrete-time Ξ-valued stochastic process, for $\Xi \subset \mathbb{R}^p$, called the noise process, $f \colon \mathcal{Y} \times \mathcal{A} \times \Xi \to \mathcal{Y}$ is the transition function, $c \colon \mathcal{Y} \times \mathcal{A} \to \mathbb{R}$ is the cost function and $y_0 \in \mathcal{Y}$ is the start state.

Essentially, it's a Markov decision process that explicitly decouples its stochastic and deterministic components, which allows us, for instance, to compute gradients of its costs with respect to actions—and thus be able to optimize the actions with respect to costs via gradient-based optimization methods. This property is evident when it's viewed as a stochastic computational graph, shown in Figure 1, where we can see there're no paths from actions to costs going through stochastic variables [Bueno 2021].

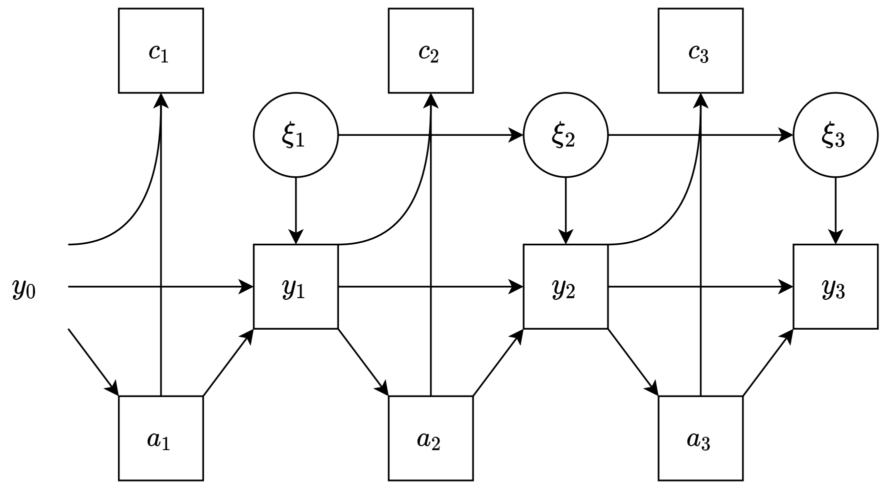

Figure 1. Stochastic computational graph of an X-MDP.

We can formulate the optimal liquidation problem, as it was defined in Section 2, as an X-MDP in a straightforward manner. Specifically, for a given time step k, let

- The state be given by the tuple (S_k, x_k, k), such that the initial state is $(S_0, x_0, 0)$;
- The cost be given by $-e^{-r_k} n_k \tilde{S}_k$;
- The action be given by a policy $\pi_\theta \colon \mathcal{Y} \to \mathcal{A}$ parametrized by $\theta \in \mathbb{R}^q$, such that $x_k = \pi_\theta(S_{k-1}, x_{k-1}, k-1)$;
- The noise random variable be ξ_k as previously defined.

The definitions for the state space, the action space and the other X-MDP components follow directly from the above. We illustrate it graphically in Figure 2, similarly to the generic one presented in Figure 1. Note how there're no paths from the cost of trading C and the policy parameters θ, indicating that we can differentiate any differentiable function of C that we can compute with respect to θ.

3.2. Algorithm

As mentioned in the end of Section 2, our goal is to minimize, with respect to θ, the loss $L = \mathbb{E}[C] + \lambda\sqrt{\mathbb{V}[C]}$ for a particular value of $\lambda \in \mathbb{R}_{\geq 0}$. In order to do that, using gradient-based optimizers, we must of course be able to compute (or approximate) its

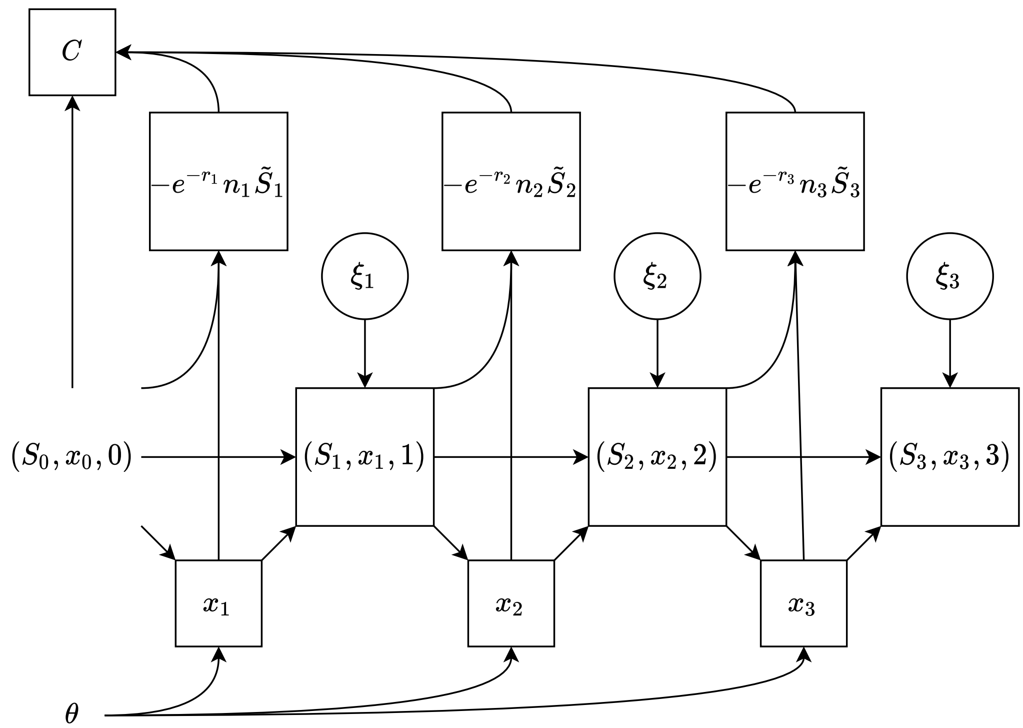

Figure 2. Stochastic computational graph of the optimal liquidation X-MDP.

gradient $\nabla_\theta L$. We can't compute neither the loss L nor its gradients $\nabla_\theta L$ analytically though, but we can't approximate the loss L by

$$\hat{L} = \bar{C} + \lambda \sqrt{\frac{1}{M} \sum_{m=1}^{M} (C_m - \bar{C})^2}\,,$$

where $\bar{C} = \frac{1}{M} \sum_{m=1}^{M} C_m$ and C_m, for $m = 1, \ldots, M$, are samples from forward passes through the optimal liquidation X-MDP for a fixed θ.

We'll then approximate the gradient of the loss $\nabla_\theta L$ by the gradient of the approximate loss $\nabla_\theta \hat{L}$. Note, however, that this isn't an unbiased estimator of $\nabla_\theta L$, so that the usual convergence guarantees of stochastic optimization methods don't apply [Robbins and Monro 1951]. On the other hand, we provide evidence through our experimental results in Section 4 that it may be a sufficiently good approximation for this particular problem.

The exact process described so far to adjust our parametrized policy π_θ is presented in pseudocode in Algorithm 1. It's essentially the same algorithm proposed in [Bueno et al. 2019], the only difference being in the gradient estimator used—as our loss isn't simply an expected value. It can be implemented easily using any modern numerical framework that implements reverse mode automatic differentiation, some of which work on computational graphs directly, such as TensorFlow [Abadi et al. 2015].

Briefly going through its steps, it starts by initializing the policy parameters θ to, in principle, any arbitrary values (although this of course influences the algorithm convergence). Next, for a given number of iterations J, it samples M paths from forward passes

Algorithm 1: Differentiable Planning for Optimal Liquidation

Input : number of iterations J and other values as previously defined (e.g., X-MDP and loss $\hat{L}$)

Output: adjusted parameters θ

Initialize parameters θ
for $j = 1$ **to** J **do**
  for $m = 1$ **to** M **do**
    Sample noise process $\xi = (\xi_1, \xi_2, \dots, \xi_N)$
    Compute m-th cost of trading C_m for θ and ξ
  end
  Compute approximate loss $\hat{L}$ and its gradient $\nabla_\theta \hat{L}$ with respect to θ
  Update parameters $\theta \leftarrow \mathrm{optimizer}(\theta, \nabla_\theta \hat{L})$
end

return θ

through the optimal liquidation X-MDP, computes the approximate loss $\hat{L}$ and its gradient $\nabla_\theta \hat{L}$ with respect to θ, updating θ at the end using any gradient-based optimizer—for example, Adam [Kingma and Ba 2015].

4. Experiments

In this section, we present and discuss the experiments performed to validate the proposed method. We start by briefly going through the setup used for the experiments, consisting of the specific liquidation process and policy parametrization and implementation, followed by the experiments themselves.

4.1. Setup

For the optimal liquidation problem, we'll take almost all parameters to have the same values as in [Almgren and Chriss 2001], as summarized in Table 1. For instance, we'll assume a constant drift $\mu_k = \mu$ and volatility $\sigma_k = \sigma$ for the stock price diffusion, constant discount rate $r_k = r$ and linear temporary $h(S_{k-1}, n_k) = \epsilon + \eta n_k/\tau$ and permanent impacts $g(S_{k-1}, n_k) = \gamma n_k/\tau$ of trading, parametrized by ϵ, η and γ. The liquidation period T varies between experiments.

The policy π_θ is implemented as standard multi-layer perceptron (MLP) [Goodfellow et al. 2016] with 10 hidden units, whose last layer outputs the proportion of the current holding x_k to hold next, being multiplied by the current holding x_k to arrive at x_{k+1}. Its parameters θ are adjusted using Algorithm 1 with $M = 4 \times 10^4$ paths, $J = 3 \times 10^4$ iterations and the Adam optimizer. The experiments were implemented using JAX, a Python library for numeric computation that implements reverse mode automatic differentiation [Bradbury et al. 2018].

4.2. Short Liquidation Periods

For short liquidation periods, the Almgren-Chriss model approximates ours very closely. In that sense, under short-term horizons, we can consider the Almgren-Chriss optimal

Table 1. Optimal liquidation parameter values

Parameter	Value
S_0	50 \$
x_0	10^6 shares
τ	1 day
μ	10% / year
σ	30% / year$^{1/2}$
r	8% / year
ϵ	0.0625 \$
η	2.5×10^{-6} \$/(shares/day)
γ	2.5×10^{-7} \$/(shares/day)

analytical solution to be approximately optimal for our model as well.

We can thus compare our proposed method's performance against this optimal solution to analyze how well the former converges towards the latter. This comparison is shown graphically in Figure 3, where the expected cost of trading and its standard deviation are plotted for different policies and a liquidation time T of 5 days.

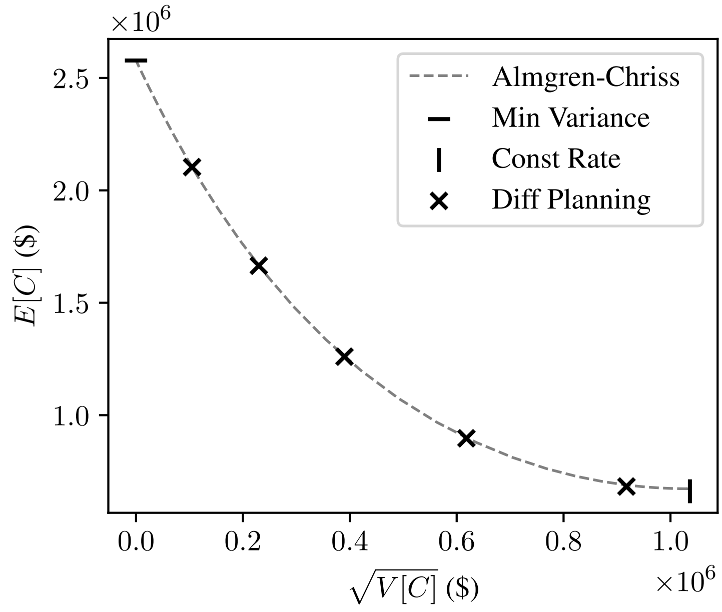

Figure 3. Expected values and standard deviations of the cost of trading for different policies and liquidation time T of 5 days.

The "Almgren-Chriss" dashed curve shows these values for the Almgren-Chriss analytical solution obtained by varying λ from 0 to ∞. This curve is referred to as the optimal frontier, as any given liquidation strategy with a certain expected cost of trading must have a standard deviation of cost of trading greater than or equal to the value for the same expected cost of trading along this curve.

The "Min Variance" and "Const Rate" bars refer to the minimum variance and constant rate policies, respectively. The former consists of selling all shares as soon as possible (i.e., $x_k = 0$, for $k \geq 1$) and is the most risk-averse strategy possible, for which $\sqrt{\mathbb{V}[C]} = 0$. The latter, on the other extreme, consists of selling a constant amount of shares at each time step, which minimizes the market impacts of selling and can be seen as the most risk-neutral strategy possible.

Finally, the "Diff Planning" markers refer to policies adjusted via the proposed differentiable planning algorithm for different values of λ—from left to right, 4, 3, 2, 1 and 0.3. Note that the differentiable planning policies achieve the optimal trade-off, for a each level of risk-aversion λ, between minimizing the expected value and standard deviation of the cost of trading C, as they lie on the optimal frontier.

4.3. Long Liquidation Periods

For long liquidation periods, the difference between the Almgren-Chriss model and ours makes itself more evident. For instance, under this setting, we would expect the Almgren-Chriss analytical strategy to be sub-optimal, in the sense of there existing less risky strategies for a given expected cost of trading.

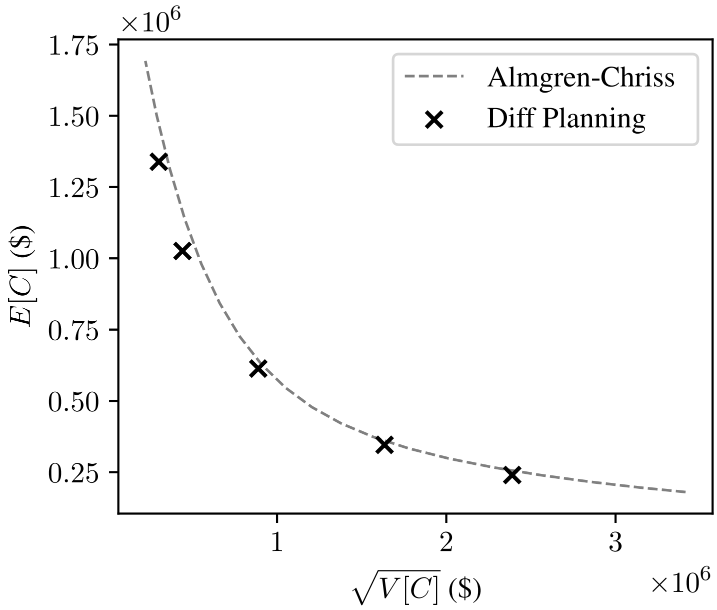

Figure 4. Expected values and standard deviations of the cost of trading for different policies and liquidation time T of 60 days.

This is exactly what is shown in Figure 4. As in Figure 3, it shows the expected cost of trading and its standard deviation for the Almgren-Chriss analytical policies and the proposed differentiable planning policies, but now considering a liquidation time T of 60 days. Note that the Almgren-Chriss analytical policies are clearly sub-optimal as the differentiable planning policies have smaller standard deviations of cost of trading for the same expected cost of trading values.

5. Conclusion

In this work, we have extended the Almgren-Chriss model to a more generic and realistic setting, where the stock price process follows a geometric Brownian motion with deterministic drift and local volatility, and shown how it naturally fits into an X-MDP. Next, and most importantly, we've proposed a differentiable planning algorithm that allows us to solve this optimal planning X-MDP approximately, requiring only the differentiability of some of its components to be amenable to gradient-based optimization methods.

We've shown through experiments that, for short-terms liquidation horizons, the proposed differentiable planning strategies lie in the optimal frontier defined by the analytical solution to the Almgren-Chriss model, which approximates well our model under this setting. For long-term horizons, for which the Almgren-Chriss analytical solution is sub-optimal as its model is no longer a good approximation of ours, we've shown that our differentiable planning strategy indeed overperforms it.

For future work, a straightforward potential improvement to our differentiable planning algorithm would be developing a guaranteedly unbiased gradient estimator for it and analyzing how it compares to ours, especially for more complex scenarios than the ones analyzed here. One alternative, but related, direction to explore is using stochastic compositional gradient descent optimizers with our method, which deal directly with losses such as ours [Wang et al. 2017].

References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

Almgren, R. and Chriss, N. (2001). Optimal execution of portfolio transactions. *Journal of Risk*, 3:5–40.

Bradbury, J., Frostig, R., Hawkins, P., Johnson, M. J., Leary, C., Maclaurin, D., Necula, G., Paszke, A., VanderPlas, J., Wanderman-Milne, S., and Zhang, Q. (2018). JAX: composable transformations of Python+NumPy programs.

Bueno, T. P. (2021). *Planning in stochastic computation graphs: solving stochastic nonlinear problems with backpropagation*. PhD thesis, Universidade de São Paulo.

Bueno, T. P., de Barros, L. N., Mauá, D. D., and Sanner, S. (2019). Deep reactive policies for planning in stochastic nonlinear domains. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 7530–7537.

Derman, E. and Kani, I. (1994). Riding on a smile. *Risk*, 7(2):32–39.

Dupire, B. et al. (1994). Pricing with a smile. *Risk*, 7(1):18–20.

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep learning*. MIT press.

Hull, J. C. (2006). *Options, futures and other derivatives*. Prentice Hall, Upper Saddle River, US, 6th ed edition.

Kingma, D. P. and Ba, J. (2015). Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Robbins, H. and Monro, S. (1951). A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407.

Wang, M., Fang, E. X., and Liu, H. (2017). Stochastic compositional gradient descent: algorithms for minimizing compositions of expected-value functions. *Mathematical Programming*, 161(1):419–449.

Wu, G., Say, B., and Sanner, S. (2017). Scalable planning with tensorflow for hybrid nonlinear domains. *Advances in Neural Information Processing Systems*, 30.