

漸近展開法を用いた機械学習による制約付き最適ポートフォリオ問題への応用

内藤 誠* 竹原 浩太†

2024年2月16日

概要

効用関数を対数型以外にする場合、制約付き最適ポートフォリオを陽に解くことはほとんどできない。本稿では制約付き最適ポートフォリオ問題を数値的に解く方法を提案する。本手法は完備市場における制約なし最適ポートフォリオ問題で応用された漸近展開法 (Takahashi and Yoshida, 2004) と、最適ポートフォリオ問題を後退確率微分方程式 (以下, BSDE) に書き直す手法 (Hu et al., 2005), BSDE を機械学習を用いて解く手法 (E et al., 2017; Han et al., 2018) を組み合わせる。その際, BSDE を制約なし最適ポートフォリオ問題部分とそれ以外に分解することが重要である。数値シミュレーションの結果, 本手法は従来の手法と比較して効率的に制約付き最適ポートフォリオを推定できる可能性があることが示唆される。

1 はじめに

最適ポートフォリオ問題はファイナンスにおいて重要な課題の一つである。しかしながら、問題を陽に解くためにはしばしば強い仮定が必要であり、逆に仮定を緩めすぎると問題が解けない場合や解析的に求まらないことが多い。また、最適ポートフォリオ問題を考える上で投資ホライズンを1期間と捉えるのかそれとも多期間と捉えるかでは最適ポートフォリオを導出する難しさがかなり異なる。1期間モデルに対しては、任意の確率分布の下で効用関数が2次効用の場合または任意のリスク回避的効用関数の下で投資収益率が正規分布に従う状況では最適ポートフォリオは平均分散アプローチで解けることが知られている。一方で、多期間モデルにおいて投資収益率の確率分布が変動するような設定の下では、特別な効用関数を除いてその最適ポートフォリオ問題を解くことは難しい。Samuelson (1969) と Merton (1969) は1期間の問題を多期間に拡張するにあたり、効用関数が時間を通じて同一という仮定の下で各期の期待効用の総和を最大にするように期初に最適な消費と最適なポートフォリオを構築する方法を考えている。その下で彼らは投資家の効用関数が対数効用の場合またはベキ型効用かつ投資収益率が独立同分布に従う場合には多期間の問題が1期間の問題と一致することを示している。しかしながら、より一般的な仮定の下では、完備市場で制約のない問題であっても多期間の場合に陽に解くことは難しいことが知られている。特に、制約付きの最適ポートフォリオ問題を扱う場合には制約なしの問題を解く場合と比較して難易度が高い。

* 東京都立大学大学院 経営学研究科 経営学専攻 博士課程後期

† 東京都立大学大学院 経営学研究科 経営学専攻

Takahashi and Yoshida (2004) では漸近展開法を用いて完備市場における制約なし最適ポートフォリオ問題の近似解を得る方法を提案している。漸近展開法 (Kunitomo and Takahashi, 2001) はマリアン解析を理論的背景とする解析的近似解を求める手法であり、近年最適ポートフォリオ問題を含む多くのファイナンスの課題に応用されている。例えば、オプション価格に関しては Kunitomo and Takahashi (1992, 2001, 2003); Takahashi (1999); Kim and Kunitomo (1999); Li (2014); Nishiba (2013); Osajima (2006); Shiraya, Takahashi and Toda (2011); Shiraya, Takahashi and Yamazaki (2011); Shiraya, Takahashi and Yamada (2012); Shiraya and Takahashi (2013, 2014); Takahashi and Takehara (2007, 2008a,b, 2010a,b); Takahashi and Yamada (2011, 2012) があげられる。彼らは平均オプションやバスケットオプションを含む様々なオプション等に漸近展開法を応用している。Takahashi and Yoshida (2004) は完備市場における制約なし最適ポートフォリオ問題に対して、マルチンゲール法を適用し、そこに漸近展開法を用いることで近似解を解析的に求めている点の特徴的である。しかしながら、1次近似など低次の近似では誤差が大きいことも多く、最適ポートフォリオを精緻に求めるにはより高次の計算などが必要となる。

一方、最適ポートフォリオ問題を BSDE で表現することで数値的に解く方法も考えることができる。Hu et al. (2005) は完備市場における最適ポートフォリオ問題の後退確率微分方程式表現を与えている。彼らは効用関数が指数効用やベキ型効用である最適ポートフォリオ問題を BSDE に書き換え、その BSDE の解から最適ポートフォリオや価値関数の値を導出している。彼らの手法では対数効用のときのみ陽に最適ポートフォリオを導ける一方で、一般的な条件の下では何らかの数値解法を用いて BSDE を解き、最適ポートフォリオを導く必要がある。

また近年、このような研究とは独立に BSDE の解を数値的に解く方法が数多く提案され、その中には漸近展開法を用いる手法も含まれている。例えば、Takahashi and Yamada (2016) はピカールの逐次近似に漸近展開法を組み合わせることで効率的に BSDE を解く方法を提案している。その他にも BSDE への漸近展開法の応用には Fujii and Takahashi (2012, 2015, 2018a,b, 2019) などが挙げられる。

それとは別に機械学習手法を用いて BSDE を効率的に解く方法 (以下、この手法を DeepSolver 法と呼ぶ) も提案されている (E et al., 2017; Han et al., 2018)。DeepSolver 法は BSDE の解を機械学習手法の一つであるニューラルネットワークモデルで表現し、多くのシミュレーションデータをモデルにインプットしトレーニングを繰り返して解く手法である。さらに、DeepSolver 法を改良することでより効率的に BSDE を解こうとする先行研究もある。Takahashi et al. (2021) は高次元の半線形偏微分方程式を効率的に解くために、漸近展開法と DeepSolver 法を組み合わせつつ対応する BSDE を解くことで、単純な DeepSolver 法と比較して計算速度や精度が改善したことを示している。具体的には BSDE 自体を 2 つに分解し、片方の BSDE に漸近展開法を適用し近似解を求め、もう一方の BSDE にその近似解を適用することでより効率的に機械学習が行えるように工夫している。Takahashi et al. (2021) は BSDE のドライバーにリプシッツ性を仮定しているが、ドライバーが本稿で扱うような 2 次成長の場合にも漸近展開法と DeepSolver 法を適用する先行研究もある (Fujii et al. (2019))。

内藤・竹原 (2024) では Takahashi and Yoshida (2004) のポートフォリオにおける高次部分を補うために機械学習を用いてる。具体的には Hu et al. (2005) の方法を用いて制約なし最適ポートフォリオ問題を BSDE で表現し、その BSDE に対して漸近展開法と機械学習を適用することでより効率的に制約なし最適ポートフォリオを数値的に解く方法を提案している。数値シミュレーションの結果、内藤・竹原 (2024) の手法は従来の手法と比較して効率的に最適ポートフォリオを推定できる可能性があることを示唆している。

そこで本研究では制約付き最適ポートフォリオ問題を数値的に解く新しい手法を提案する。まず、Hu et al. (2005) の方法を用いて制約付き最適ポートフォリオ問題を BSDE で表現し、その BSDE を制約なし最適ポ

トフォリオ問題に関連する部分とそれ以外の部分に分解する。次に、制約なし最適ポートフォリオ問題に関連する BSDE は内藤・竹原 (2024) の手法を適用し効率的に求め、残りの BSDE には DeepSolver 法を適用することで制約なし最適ポートフォリオ問題を数値的に解く方法を提案する。数値シミュレーションの結果、本手法は従来の手法と比較して効率的に最適ポートフォリオを導出できる可能性があることを示唆している。後述するように同じく BSDE に対して漸近展開法と機械学習を組み合わせる Takahashi et al. (2021) や Fujii et al. (2019) は BSDE 自体を単純なドライバーを持つものとそれ以外に分け、後者に機械学習を適用する際に前者を事前知識として与えている。一方で本手法では、ポートフォリオ最適化についてすでに知られている結果を用いて、ポートフォリオを既知の部分とそれ以外に分けた上で後者に機械学習を適用する際に前者を事前知識として与えており、同じ漸近展開法を応用する手法であっても問題の分解方法や与える事前知識に違いがある。また、本研究の目的は最適ポートフォリオ問題の価値関数だけでなく、最適ポートフォリオ自体を求めることにある。多くの DeepSolver 法を用いた研究では CVA やオプション価格、すなわち BSDE の初期値にのみ焦点があたる。一方で、資産運用の実務においては極めて重要な問題の中には、最適ポートフォリオ問題のように BSDE の初期値の水準よりも得られたポートフォリオの最適性に関心があるものも多い。DeepSolver 法のように BSDE の初期値それ自体をニューラルネットワークで表現する場合、トレーニングで推定されたポートフォリオが達成する期待効用がポートフォリオと同時に推定された初期値と一致するとは限らない。本研究ではポートフォリオの最適性に焦点をあてている点もこれまでの先行研究とは異なる。

最後に、本論文の構成について述べる。第 2 節は提案手法の概要を述べる。第 3 節では、提案手法に関連する内藤・竹原 (2024) の手法を概観し、第 4 節は提案手法を整理し、第 5 節では具体的な数値例を示す。最後にまとめとして第 6 節で結論と今後の課題について述べる。

2 提案手法の概要

d 個のリスク資産と 1 個のリスクフリー資産が存在する市場を考える。 d 個のリスク資産 S_i , $i = 1, \dots, d$ は以下の確率微分方程式の解として表されるとする。

$$dS_i(t) = S_i(t) \left[b_i(t)dt + \sum_{j=1}^d \sigma_{ij}(t)dw_j(t) \right], \quad S_i(0) = s_i > 0, \quad i = 1, \dots, d,$$

ただし、 w_j ($j = 1, \dots, d$) は確率測度 \mathbf{P} の下での互いに独立な標準ブラウン運動である。リスクフリー資産 S_0 は以下の確率微分方程式の解として表されるとする。

$$dS_0(t) = r(t)S_0(t)dt, \quad S_0(0) = 1.$$

ただし、 $r(t)$, $b_i(t)$, $\sigma_{ij}(t)$, $i, j = 1, \dots, d$ は有界かつ $\{\mathcal{F}_t\}$ に関して発展的可測であり、 $\sigma(t) := \{\sigma_{ij}(t)\}_{i,j}$ は正則かつ非退化条件 (nondegeneracy condition) を満たすとする。また、初期富 W を保有し、各時点 t で投資額 $\pi(t) = (\pi_1(t), \dots, \pi_d(t))^\top$ をリスク資産に投資する投資家について考える。ただし、今回は各時点の消費については考えないことにする。このとき、富過程 W^π を以下の確率微分方程式の解とする。

$$dW^\pi(t) = [r(t)W^\pi(t)]dt + \pi(t)^\top [(b(t) - r(t)\mathbf{1})dt + \sigma(t)dw(t)], \quad W^\pi(0) = W > 0.$$

ただし、 $w(t) := (w_1(t), \dots, w_d(t))^\top$, $\mathbf{1} = (1, 1, \dots, 1)^\top \in \mathbf{R}^d$ である。また、 $\pi(t)$ が可積分条件を満たすとする。以下、表記を簡単にするために

$$\theta(t) := ((b(t) - r(t)\mathbf{1})^\top \sigma(t))^\top \tag{1}$$

とおく.

このとき, 以下の期待効用最大化問題^{*1}を考える.

$$V(W) = \sup_{\pi: \pi(t) \in \mathcal{A}} \mathbf{E}[U(W^\pi(T))], \quad (3)$$

ただし, $\mathcal{A} \subset \mathbf{R}^d$ はある制約を表す閉集合である. また, 期待値はすべて \mathbf{P} の下で取るとする.

さらに, 集合 $C \subset \mathbf{R}^d$ に対して

$$\Pi_C(a) := \{b \in C \mid |a - b| = \text{dist}_C(a)\}, \quad \text{dist}_C(a) := \inf_{b \in C} |a - b| \quad (4)$$

とし, $C(t) := \mathcal{A}\sigma_t = \{a\sigma(t) \mid a \in \mathcal{A}\}$ とする. ただし, C が閉集合である場合, $\Pi_C \neq \emptyset$ であることに注意する. このとき, Hu et al. (2005) より以下の BSDE を解くことで価値関数 $V(W)$ と各時刻 t における最適ポートフォリオウェイト $(\pi_t^*)/W_t^\pi$ を導けることが知られている.

定理 1 (Hu et al. (2005)) 価値関数は以下で与えられる.

$$V(W) = \frac{1}{\gamma} W^\gamma \exp(Y(0)), \quad x > 0, \quad (5)$$

ただし, $Y(0)$ は以下の BSDE の解 $(Y, Z) \in \mathcal{H}^\infty(\mathbf{R}) \times \mathcal{H}^2(\mathbf{R}^d)^{*2}$ で与えられる:

$$Y(t) = 0 - \int_t^T Z(s)^\top dw(s) - \int_t^T f(s, Z(s)) ds, \quad t \in [t, T], \quad (6)$$

ただし, ドライバー f は次の通り与えられる.

$$f(t, z) = \frac{\gamma(1-\gamma)}{2} \text{dist}_{C(t)}^2 \left(\frac{1}{1-\gamma} (z + \theta_t) \right) - \frac{\gamma|z + \theta_t|^2}{2(1-\gamma)} - \frac{1}{2}|z|^2 \quad (7)$$

このとき, 最適ポートフォリオは以下で特徴付けられる.

$$(\pi(t)^*)^\top / W^{\pi^*}(t) \in \Pi_{C(t)} \left(\frac{1}{1-\gamma} (Z(t) + \theta(t)) \right). \quad (8)$$

また, 機械学習を用いることで BSDE を効率的に解く方法が提案されている. E et al. (2017) と Han et al. (2018) が提案した DeepSolver 法は BSDE(6) をニューラルネットワークを用いて以下のように解く.

$$\inf_{\hat{Y}(0), \hat{Z}} \mathbf{E} \left[\left| 0 - \hat{Y}(T) \right|^2 \right] \quad (9)$$

ただし, $\hat{Y}(T)$ は (6) 式に対応する以下の確率微分方程式のオイラー・丸山近似である.

$$\hat{Y}(t) = \hat{Y}(0) + \int_0^t \hat{Z}(s)^\top dw(s) + \int_0^t f(s, \hat{Z}(s)) ds. \quad (10)$$

以下では混同の恐れがない場合には (10) 式を用いて (9) 式を解くことを単に (6) 式を解くと表現する.

^{*1} ただし, 本稿では効用関数 U はベキ型効用とする.

$$U(x) = \frac{1}{\gamma} x^\gamma, \quad x \geq 0, \quad \gamma \in (0, 1). \quad (2)$$

^{*2} $\mathcal{H}^2(\mathbf{R}^d) : \mathbf{E}[\int_0^T |\theta(t)|^k dt] < \infty$ を満たす \mathbf{R}^d 値の確率過程 θ の集合である. また, $\mathcal{H}^\infty(\mathbf{R}^d)$ はほとんど至る所有界な \mathbf{R}^d 値の確率過程の集合である.

一方で、本研究では BSDE(6) を以下のように制約なし最適ポートフォリオ問題を表す BSDE と残りの BSDE に分解する.

$$Y^1(t) = 0 - \int_t^T Z^1(s)^\top dw(s) + \int_t^T f_1(s, Z^1(s)) ds, \quad (11)$$

$$Y^2(t) = 0 - \int_t^T Z^2(s)^\top dw(s) + \int_t^T f_2(s, Z^1(s), Z^2(s)) ds, \quad (12)$$

ただし,

$$f_1(t, z^1) = -\frac{\gamma}{2(1-\gamma)} |z^1 + \theta(t)|^2 - \frac{1}{2} |z^1|^2, \quad (13)$$

$$f_2(t, z^1, z^2) = -\frac{\gamma}{2(1-\gamma)} [2|z^1 + \theta(t)|^\top z^2 + |z^2|^2] - \frac{1}{2} [2(z^1)^\top z^2 + |z^2|^2] \\ + \frac{\gamma(1-\gamma)}{2} \text{dist}_{C_t}^2 \left(\frac{1}{1-\gamma} (z^1 + z^2 + \theta_t) \right). \quad (14)$$

このうち、BSDE(11) を何らかの方法で近似的に解き、その近似解 $(\hat{Y}^{1,*}(0), \hat{Z}^{1,*})$ を得たとする. このとき、これらを BSDE(12) に代入した上で、DeepSolver 法を用いて $(\hat{Y}^2(0), \hat{Z}^2)$ を以下のように学習し、解を得ることができる.

$$\inf_{\hat{Y}^2(0), \hat{Z}^2} \mathbf{E} \left[\left| 0 - \hat{Y}(T) \right|^2 \right]. \quad (15)$$

ただし、 $\hat{Y}(T)$ は (12) に対応する以下の確率微分方程式のオイラー・丸山近似である.

$$\tilde{Y}(t) = (\hat{Y}^{1,*}(0) + \hat{Y}^2(0)) + \int_0^t (\hat{Z}^{1,*}(s) + \hat{Z}^2(s))^\top dw(s) + \int_0^t f(s, \hat{Z}^{1,*}(s) + \hat{Z}^2(s)) ds. \quad (16)$$

すなわち、制約付き最適ポートフォリオ問題の BSDE(6) を制約なし最適ポートフォリオ問題の BSDE(11) とそれ以外の BSDE(12) に分解する. 次節において、この分解が最適ポートフォリオを求める際の効率性に影響を与えるか否かを数値例を用いて検証する. 具体的には BSDE(6) を直接 DeepSolver 法で解く方法 (以下、DeepSolver 法という) と BSDE(11), (12) を順に DeepSolver 法で解く方法 (以下、DeepSolver+DeepSolver 法という) を比較検証する.

- DeepSolver 法 :
以下の問題を機械学習を用いて解く.

$$\inf_{Y(0), Z} \mathbf{E} \left[\left| 0 - \hat{Y}(T) \right|^2 \right] \quad (17)$$

ただし、 $\hat{Y}(T)$ は以下の BSDE に対応する確率微分方程式のオイラー・丸山近似であり,

$$Y(t) = 0 - \int_t^T Z(s)^\top dw(s) - \int_t^T f(s, Z(s)) ds, \quad t \in [t, T], \quad (18)$$

$f(t, z)$ は (7) で与えられる.

- DeepSolver + DeepSolver 法 :
まず、制約なし最適ポートフォリオ問題 (11) を DeepSolver 法で解く.

$$\inf_{Y^1(0), Z^1} \mathbf{E} \left[\left| 0 - \hat{Y}^1(T) \right|^2 \right] \quad (19)$$

ただし、 $\hat{Y}^1(T)$ は以下の BSDE(11) に対応する確率微分方程式のオイラー・丸山近似である。

$$Y^1(t) = 0 - \int_t^T Z^1(s)^\top dw_s - \int_t^T f(s, Z^1(s)) ds, \quad t \in [t, T]. \quad (20)$$

その後、上記で求めた $(\hat{Y}^{1,*}(0), \hat{Z}^{1,*})$ を (16) 式に代入し $(Y^2(0), Z^2)$ を DeepSolver 法で解く。

$$\inf_{Y^2(0), Z^2} \mathbf{E} \left[\left| 0 - \hat{Y}(T) \right|^2 \right]. \quad (21)$$

2.1 数値例

以下では $d = 1$ とし 1 次元の場合を数値例として扱う。資産価格の変動を表す確率微分方程式 (1) に加えて、リスクの市場価格 $\theta(t)$ が次のように時間変動することを仮定する。

$$d\theta(t) = k_2(\bar{\theta} - \theta(t))dt + \varepsilon\sigma_2dw(t), \quad (22)$$

ただし、 w は先程と同じ 1 次元標準ブラウン運動である。また、金利 r は 0 とし、危険資産価格 S を表す (1) 式において、 $\sigma(t) \equiv \sigma > 0$ とする。 S への投資金額であるポートフォリオ π を用いて、富 W^π は以下の通り定まる。

$$W^\pi(t) = W + \int_0^t \frac{W^\pi(u)\pi(u)}{S(u)} dS(u) = W + \int_0^t \pi(u)\sigma(\theta(u))du + dw(u). \quad (23)$$

このとき、投資家は期待効用最大化問題

$$V(W) = \sup_{\pi: \pi(t) \in \mathcal{A}} \mathbf{E}[U(W^\pi(T))], \quad (24)$$

において、制約条件としてポートフォリオが非負であることを課すことにする。すなわち、

$$\pi(t) \in \mathcal{A} := \mathbf{R}_+. \quad (25)$$

DeepSolver+DeepSolver 法においては、制約なし最適ポートフォリオ問題 (11) を DeepSolver 法で解く際にトレーニングをどこで打ち切るかが問題になってくる。そこで本稿では各手法の後ろに「_m(m = 0, 100, 200, 1000, 2000)」と記し問題 (11) に対する事前学習回数を記載することにする。例えば DeepSolver+DeepSolver 法の場合、制約なし最適ポートフォリオ問題 (11) に対して 100 回の学習を行った $(\hat{Y}^1(0), \hat{Z}^1)$ を用いる場合には「DeepSolver+DeepSolver 法_100」と記載する*3。

なお、機械学習によって得られた (Y, Z) の推定値 $(\hat{Y}(0), \hat{Z})$ 、あるいはそこから求められる $(\hat{V}(0), \hat{\pi})$ について、 $\hat{\pi}$ により達成される価値関数の値 $V^{\hat{\pi}}(0) = \mathbf{E}[U(W^{\hat{\pi}}(T))]$ と、 $\hat{V}(0)$ は必ずしも一致しないことに注意する。このため、ポートフォリオ自身に興味がある場合、 $\hat{V}(0)$ や損失関数の大小のみでは適切な規準となり得ない可能性がある。そこで、本研究ではポートフォリオ $\hat{\pi}$ に対して、アウトオブサンプル (以下、OS) での価値関数の推定値

$$V^{\hat{\pi}, n} := \frac{1}{n} \sum_{i=1}^n U(\hat{W}^{\hat{\pi}, i}(T)) \approx \mathbf{E}[U(W^{\hat{\pi}}(T))], \quad (26)$$

*3 以降では、「機械学習」と言った場合、ニューラルネットワークを用いることを意味するとする。本研究で用いるすべてのニューラルネットワークはインプットを $t, \gamma, \theta_t, \sigma$ の 4 つの変数として隠れ層の数は 3 層でそれぞれセル数は $4 \cdot 3 \cdot 2$ である。学習率は 10^{-3} とおく。また、バッチノーマライゼーションを各層に適用し最適化は Adam を使用する。

を評価の規準として加えた。ただし、 n は (トレーニングで使用していない) アウトオブサンプルのサンプルパスの数、 $\hat{W}^{\bar{\pi},i}(T)$ は富過程のオイラー・丸山近似の i 番目のパスにおける実現値である。またオイラー・丸山近似の離散幅を Δt と記載し以下の数値実験では $\Delta t = 1/50$ とする。

各パラメータの値は Detemple et al. (2003), Takahashi and Yoshida (2004) を参考に $\theta_0 = 0.1, \sigma = 0.2, T = 1, w_0 = 1, \epsilon = 0.03637, k_1 = 0.0824, k_2 = 0.6950, \bar{\theta} = 0.0871, \gamma = 0.5, \sigma_2 = 0.21/0.03637 \times 2$ とする。

まず、損失関数から確認する。図 1 は各手法における損失関数を示したものである。まず、DeepSolver

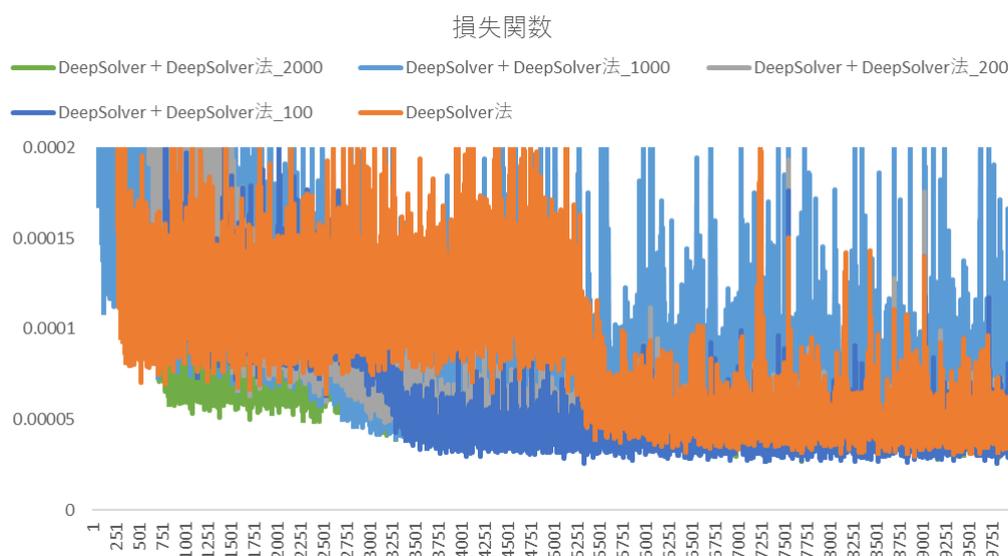


図 1: 損失関数

例 1 を DeepSolver 法・漸近展開法+ DeepSolver 法・MV+DeepSolver 法・DeepSolver+DeepSolver 法で解いた場合の損失関数を示す。横軸はトレーニング回数で縦軸が損失関数の値である。

法と比較して事前学習回数 m の異なる全ての DeepSolver+DeepSolver 法が速く収束していることが分かる。仮に BSDE(11) に対する DeepSolver 法 1 回当たりの学習時間と、BSDE(12) に対する DeepSolver 法の学習時間が同じだとしても DeepSolver+DeepSolver 法の方が少ない学習時間でより低い損失関数を達成することが分かる。次に、DeepSolver+DeepSolver 法の中で比較すると DeepSolver+DeepSolver 法_2000 や DeepSolver+DeepSolver 法_1000 が速く収束しているが、事前学習である BSDE(11) の学習時間を考慮すると DeepSolver+DeepSolver 法_200 や DeepSolver+DeepSolver 法_100 とあまり差がないことが確認できる。

次に初期ウェイトを確認する。図 2 は各手法における初期ウェイトを示したものである。損失関数と同様に、DeepSolver 法と比較して全ての DeepSolver+DeepSolver 法が速く収束していることが分かる。次に、DeepSolver+DeepSolver 法の中で比較すると DeepSolver+DeepSolver 法_2000 や DeepSolver+DeepSolver 法_1000 が速く収束していることが確認でき、これも損失関数のときと同様である。

最後にアウトオブサンプルでの期待効用を規準としてこれまでと同様に、手法内および手法間のパフォーマンスを比較する。図 3 はアウトオブサンプルでの期待効用を比較したものである。損失関数と初期ウェイトと同様に、DeepSolver 法と比較して全ての DeepSolver+DeepSolver 法が、少ないトレーニング回数で DeepSolver 法と比較して高い値を取ることが確認できる。一方で、 $m = 2000$ の場合を除いて、DeepSolver

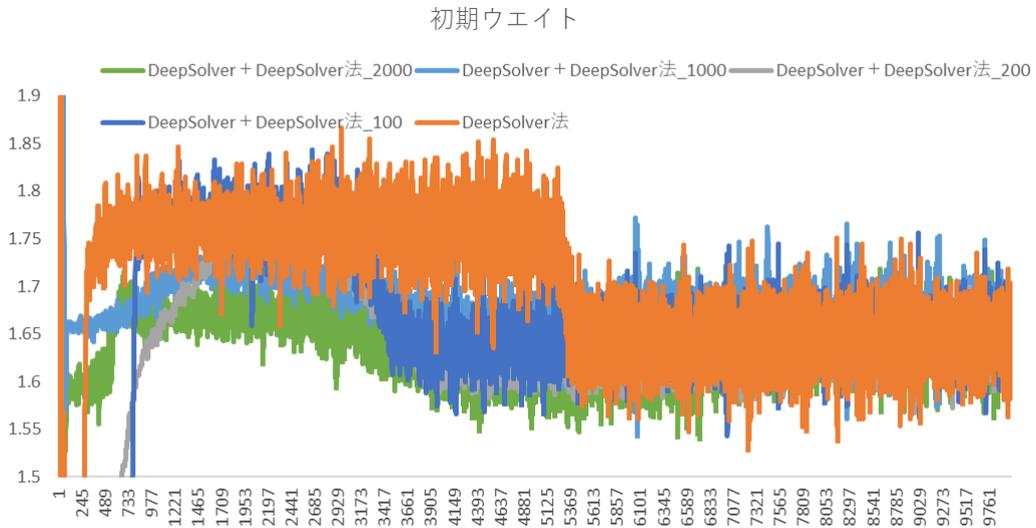


図 2: 初期ウエイト

例 1 を DeepSolver 法・漸近展開法+ DeepSolver 法・MV+DeepSolver 法・DeepSolver+DeepSolver 法で解いた場合の初期ウエイトを示す。横軸はトレーニング回数で縦軸が初期ウエイトの値である。

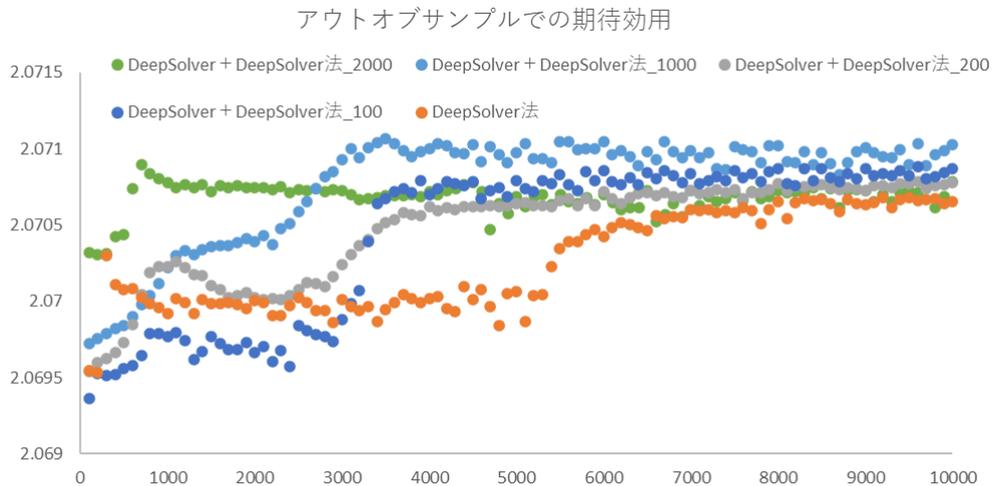


図 3: アウトオブサンプルでの期待効用

例 1 を DeepSolver 法・漸近展開法+ DeepSolver 法・MV+DeepSolver 法・DeepSolver+DeepSolver 法で解いた場合のアウトオブサンプルでの期待効用を示す。横軸はトレーニング回数で縦軸がアウトオブサンプルでの期待効用の値である。

法の場合よりも高い期待効用を達成するためには BSDE(12) に対する 3000 回程度のトレーニングを必要とすることもわかる。すなわち、制約付き最適ポートフォリオ問題の BSDE(6) をそのまま数値的に解くよりも、制約なし最適ポートフォリオ問題に該当する BSDE(11) とそれ以外の BSDE(12) に分解することでより効率的に解くことができるものの、またその際には十分な事前学習が必要であることが示唆される結果となった。

以上をまとめると、損失関数や初期ウエイト、アウトオブサンプルでの期待効用において DeepSolver+DeepSolver 法が単純な DeepSolver 法よりも高いパフォーマンスを上げることが確認できる。このことから、制約付き最適ポートフォリオ問題の BSDE(6) をそのまま数値的に解くよりも、制約なし最適ポートフォリオ問題に該当する BSDE(11) とそれ以外の BSDE(12) に分解することでより効率的に解くことができる可能性が示唆される。

また、制約なし最適ポートフォリオ問題の BSDE(11) を DeepSolver 法以外で効率的に解く方法も存在する。内藤・竹原 (2024) はポートフォリオを漸近展開法により近似しそれを DeepSolver 法で補完することで効率的に BSDE(11) を解く方法を提案している。内藤・竹原 (2024) では、次節で見るとように $\hat{Z}^{1,*}$ をさらに

$$\hat{Z}^{1,*} = Z_{\text{asy}}^1 + \hat{Z}_{\text{rem}}^{1,*} \quad (27)$$

と分解して求めており、これと組み合わせることで、最終的に (16) 式は以下のように表される。

$$\tilde{Y}(t) = (\hat{Y}^{1,*}(0) + \hat{Y}^2(0)) + \int_0^t (Z_{\text{asy}}^1(s) + \hat{Z}_{\text{rem}}^{1,*}(s) + \hat{Z}^2(s))^\top dw + \int_0^t f(s, Z_{\text{asy}}^1(s) + \hat{Z}_{\text{rem}}^{1,*}(s) + \hat{Z}^2(s)) ds \quad (28)$$

ここで、関連する先行研究に言及しておく。まず、Takahashi and Yamada (2016) については DeepSolver 法を用いておらず BSDE を漸近展開法を用いて直接近似している点が本研究とは異なる。また、前提条件としてドライバー f のリプシッツ性を要求するが、(7) 式はこれを満たさない。実際、最も単純なブラック・ショールズモデルの下で制約のない最適ポートフォリオ問題に対して予備的に数値実験を行った際も θ の値が大きいケースでは理論値から大きく乖離する結果となった。

次に、Takahashi et al. (2021) は本手法と同様に BSDE を分解し一方を漸近展開法を用いて近似し後者を DeepSolver 法で求める。ただし、分解の方法は本手法とは異なり、線形部分と非線形部分への分解である。さらに、本研究において関心のある設定下では BSDE の終端条件が 0 であるため、前者が恒等的に 0 となり、単純な DeepSolver 法と同等になる。このため、本研究における数値実験においてはこれらを含めていない。

3 内藤・竹原 (2024) の手法

本節では制約なし最適ポートフォリオ問題を扱った内藤・竹原 (2024) の手法を説明する。

まず、前節 (11) 式が制約なし問題に対応することに注意しておく。内藤・竹原 (2024) ではこの制約なし問題において、(11) 式における Z^1 を漸近展開法部分 Z_{asy}^1 と残りの部分 Z_{rem}^1 の和

$$Z^1 = Z_{\text{asy}}^1 + Z_{\text{rem}}^1 \quad (29)$$

に分け Z_{asy}^1 は定理 2 で与える漸近展開法の結果を用い Z_{rem}^1 のみを学習することで BSDE(11) を次のように解き、近似解 $(\hat{Y}^{1,*}(0), \hat{Z}_{\text{rem}}^{1,*})$ を得る。

$$\inf_{\hat{Y}^1(0), \hat{Z}_{\text{rem}}^1} \mathbf{E} \left[\left| 0 - \tilde{Y}^1(T) \right|^2 \right]. \quad (30)$$

次節では Z_{asy}^1 の導出について簡単に見る。

3.1 マルコフ型状態変数による表現

本節では, Takahashi and Yoshida (2004) に基いてマルコフ型状態変数を用いてモデルを記述する.

まず, 状態変数 X_t^ε を以下の確率微分方程式に従う d 次元の確率過程とする.

$$dX^\varepsilon(u) = v_0(X^\varepsilon(u), \varepsilon)du + v(X^\varepsilon(u), \varepsilon)dw(u), \quad X_t^\varepsilon = x.$$

ただし, $u \in [t, T]$ で, $\varepsilon \in (0, 1]$ は漸近展開に用いられるパラメータであり,

$$v_0 \in C_b^\infty(\mathbf{R}^d \times (0, 1]; \mathbf{R}^d), \quad v = (v_\beta)_{\beta=1}^d \in C_b^\infty(\mathbf{R}^d \times (0, 1]; \mathbf{R}^d \otimes \mathbf{R}^d)$$

とする*4. また, I は $d \times d$ の単位行列である.

以下の議論では, $r \in C_b^\infty(\mathbf{R}^d; \mathbf{R}_+)$, $\theta \in C_b^\infty(\mathbf{R}^d; \mathbf{R}^d)$ と仮定し, r, θ, σ は全て状態変数 $X^\varepsilon(u)$ により次のように決定されるとする.

$$r(u) = r(X^\varepsilon(u)), \quad \theta(u) = \theta(X^\varepsilon(u)), \quad \sigma(u) = \sigma(X^\varepsilon(u)).$$

また, 今後の計算を簡単にするために, 以下の確率微分方程式で定義される $d \times d$ 行列値確率過程 $y_t^\varepsilon = \{y_{t,u}^\varepsilon : u \in [t, T]\}$ に対する以下の確率微分方程式が一意的な解を持つとする.

$$dy_{t,u}^\varepsilon = \sum_{\alpha=0}^d \partial_x v_\alpha(X_u^\varepsilon, \varepsilon) y_{t,u}^\varepsilon dw_u^\alpha, \quad y_{t,t}^\varepsilon = I.$$

3.2 漸近展開法

本節では, 前節での仮定の下で Takahashi and Yoshida (2004) に基づいて最適ポートフォリオウエイトに対する漸近展開の結果を紹介する*5.

定理 2 ベキ効用に対する最適ポートフォリオウエイトの ε オーダーまでの漸近展開は以下で与えられる.

$$\begin{aligned} (\pi^*(t))^\top / W^{\pi^*}(t) &= \frac{1}{1-\gamma} \theta^\top(x) \sigma^{-1}(x) + \frac{\gamma}{1-\gamma} \varepsilon \left(\int_t^T \partial r^{[0]}(u) y_{t,u} du + \frac{1}{1-\gamma} \times \right. \\ &\quad \left. \times \sum_{\alpha=1}^d \int_t^T \theta_\alpha^{[0]}(u) \partial \theta_\alpha^{[0]}(u) y_{t,u} du \right) \partial_\varepsilon v(x, 0) \sigma^{-1}(x) + o(\varepsilon). \end{aligned} \quad (31)$$

このとき, 3 節の Z_{asy}^1 は以下のように具体的に表現することができる.

$$\begin{aligned} Z_{\text{asy}}^1(t) &= (1-\gamma) \times \left(\frac{1}{1-\gamma} \theta^\top(x) \sigma^{-1}(x) + \frac{\gamma}{1-\gamma} \varepsilon \left(\int_t^T \partial r^{[0]}(u) y_{t,u} du + \frac{1}{1-\gamma} \times \right. \right. \\ &\quad \left. \left. \times \sum_{\alpha=1}^d \int_t^T \theta_\alpha^{[0]}(u) \partial \theta_\alpha^{[0]}(u) y_{t,u} du \right) \partial_\varepsilon v(x, 0) \sigma^{-1}(x) \right) \sigma(x) - \theta(x). \end{aligned} \quad (32)$$

*4 $C_b^\infty(\mathbf{R}^d \times (0, 1]; E)$: 滑らかな関数 $f: \mathbf{R}^d \times (0, 1] \rightarrow E$ の偏微分係数 $\partial_x^n \partial_\varepsilon^m f(x, \varepsilon)$ がすべて有界であるものの集合

*5 漸近展開法の詳細については Takahashi and Yoshida (2004) と内藤・竹原 (2024) を参照.

4 提案手法の整理

ここで再び、BSDE(6)によって表現される制約付き最適ポートフォリオ問題に戻る。これを機械学習を用いて数値的に解く方法として以下の4つを考えることができる*6。

1. DeepSolver 法：2節参照。
2. DeepSolver + DeepSolver 法：2節参照。
3. 漸近展開法 + DeepSolver 法：

制約なし最適ポートフォリオ問題 (11) を内藤・竹原 (2024) の方法で解いてから残りの BSDE(12) を解く方法である。具体的には、 Z^1 を漸近展開法による近似 Z_{asy}^1 と残りの部分 Z_{rem}^1 に分け、 Z_{rem}^1 のみ学習する。

$$\inf_{\hat{Y}^1(0), \hat{Z}_{\text{rem}}^1} \mathbf{E} \left[\left| 0 - \hat{Y}^1(T) \right|^2 \right] \quad (33)$$

ただし、 $\hat{Y}^1(T)$ は以下の BSDE に対応する確率微分方程式のオイラー・丸山近似である。

$$\tilde{Y}^1(t) = 0 - \int_t^T (Z_{\text{asy}}^1(s) + Z_{\text{rem}}^{1,*}(s))^\top dw(s) - \int_t^T f_1(s, Z_{\text{asy}}^1(s) + Z_{\text{rem}}^{1,*}(s)) ds, \quad t \in [t, T]. \quad (34)$$

その後、上記で求めた $(\hat{Y}^{1,*}(0), \hat{Z}^{1,*} = Z_{\text{asy}}^1 + \hat{Z}_{\text{rem}}^{1,*})$ を (16) 式に代入し $(Y^2(0), Z^2)$ を DeepSolver 法により求める。

$$\inf_{\hat{Y}^2(0), \hat{Z}^2} \mathbf{E} \left[\left| 0 - \hat{Y}(T) \right|^2 \right]. \quad (35)$$

4. MV + DeepSolver 法：

漸近展開法 + DeepSolver 法において、 Z^1 に対する漸近展開による近似 Z_{asy}^1 の代わりに、平均分散 (以下、MV と示す) の項である

$$Z_{\text{MV}}^1 := (1 - \gamma) \times \frac{1}{1 - \gamma} \theta(x)^\top \sigma^{-1}(x) \times \sigma(x) - \theta(x) \quad (36)$$

を用い、 Z^1 をこれと残りの部分 Z_{rem}^1 に分け、 Z_{rem}^1 を学習する方法を MV + DeepSolver 法とする。

$$\inf_{\hat{Y}^1(0), \hat{Z}_{\text{rem}}^1} \mathbf{E} \left[\left| 0 - \hat{Y}^1(T) \right|^2 \right] \quad (37)$$

つまり、 Z_{MV}^1 は Z_{asy}^1 のゼロ次部分に相当する。

5 数値例

問題設定は2節のものと同じものを使用し、上記の4つの手法について比較する。

以下では手法間の比較を行うために、損失関数・初期ウエイト・アウトオブサンプルをでの期待効用のそれぞれにおいて事前学習回数別に比較する。なお、本節では制約なし最適ポートフォリオ問題 (11) に対する機

*6 ただし、期待効用最大化問題 (3) を直接機械学習で解く方法も考えることができるが、より単純である制約なし最適ポートフォリオ問題の場合にパフォーマンスが悪かったためここでは省略する (内藤・竹原 (2024) 参照)。

機械学習の適用回数を「事前学習回数」と呼び、残りの BSDE(12) に対する機械学習の適用回数を「トレーニング回数」と呼ぶことにする。

■損失関数 まず、損失関数から確認する。

事前学習回数が 0 の場合

事前学習回数が 0 の場合の損失関数を図 4 に示す。図から MV+Deepsolver 法_0 が最も低い値を取り

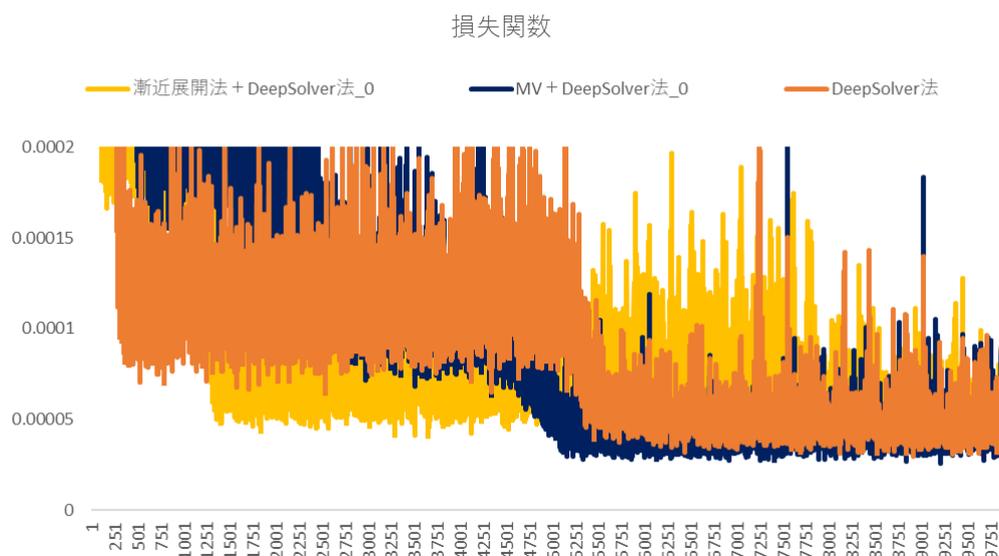


図 4: 損失関数

DeepSolver 法・漸近展開法 + DeepSolver 法・MV+DeepSolver 法・DeepSolver+DeepSolver 法で解いた場合の損失関数を示す。横軸はトレーニング回数で縦軸が損失関数の値である。

DeepSolver 法^{*7}よりも速く収束していることが分かる。一方で、漸近展開法 + DeepSolver 法_0 は 1000 回から 4000 回程度までのトレーニング回数では損失関数が最も低い値を取るがそれ以降では他の手法に逆転される傾向にある。

事前学習のトレーニング回数が 200 の場合

図 5 は事前学習のトレーニング回数が 200 の場合の損失関数である。明らかに漸近展開法 + DeepSolver 法_200 が最も速く収束していることが分かる。また、MV+DeepSolver 法は最も収束が遅く、8000 回程度のトレーニングを行うことでようやく漸近展開法 + DeepSolver 法の値にまで損失関数が低下する。DeepSolver+DeepSolver 法_200 は MV+DeepSolver 法よりも速く収束しているが、漸近展開法 + DeepSolver 法の値にまで低下するには 4000 回程度のトレーニングを行う必要があることが確認できる。

事前学習回数が 2000 の場合

図 6 は事前学習回数が 2000 の場合の損失関数である。漸近展開法 + DeepSolver 法や MV+DeepSolver 法は DeepSolver+DeepSolver 法よりも収束が速いが、図 4 や 5 と異なり、各手法の間に収束の速さや損失関数の水準に差が生じにくくなっている。これは内藤・竹原 (2024) の制約なし最適ポートフォリオ問題の数値例と同様にトレーニング回数が増加するに従い各手法の損失関数の値に差が生じにくくなっていることと整合的

*7 ただし、DeepSolver+DeepSolver 法_0 は DeepSolver 法と同じものを意味していることに注意する。

損失関数

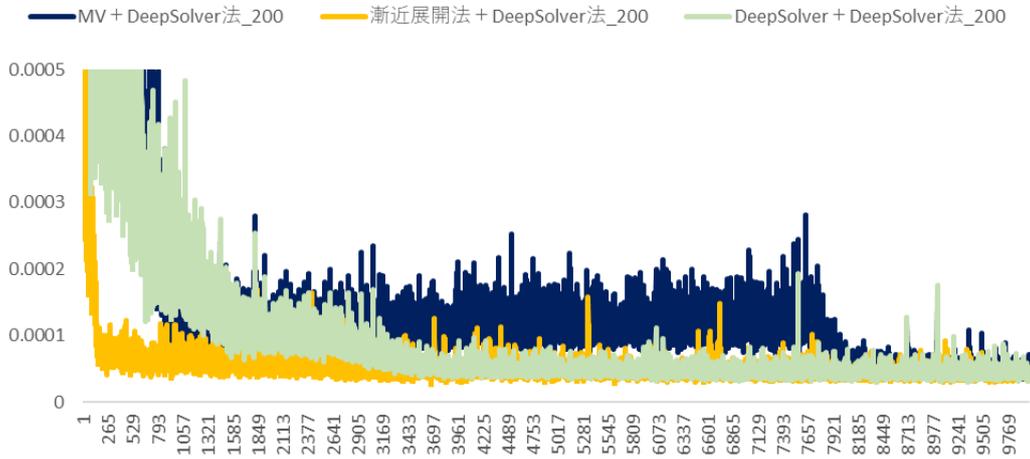


図 5: 損失関数

DeepSolver 法・漸近展開法 + DeepSolver 法・MV+DeepSolver 法・DeepSolver+DeepSolver 法で解いた場合の損失関数を示す。横軸はトレーニング回数で縦軸が損失関数の値である。

損失関数

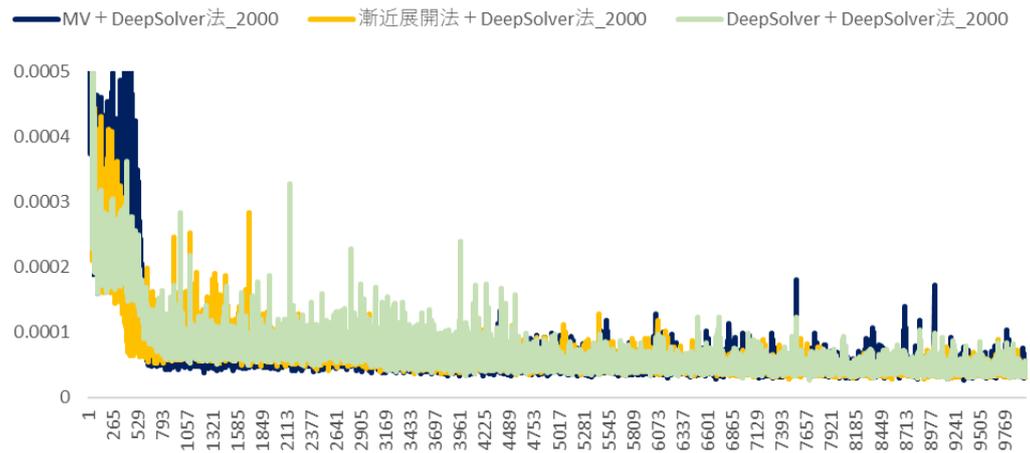


図 6: 損失関数

DeepSolver 法・漸近展開法 + DeepSolver 法・MV+DeepSolver 法・DeepSolver+DeepSolver 法で解いた場合の損失関数を示す。横軸はトレーニング回数で縦軸が損失関数の値である。

である。

以上をまとめると、損失関数においては漸近展開法 + DeepSolver 法が最も速く収束するような振る舞いを示しており、事前学習を 200 回程度で効率的に損失関数を減少させることが可能であることが示唆される。一方で、MV+DeepSolver 法や DeepSolver+DeepSolver 法は事前学習回数を行う 2000 回程度にすることで収束を速めることができるが、事前知識の獲得に時間がかかるというデメリットがある。

■初期ウエイト 次に、初期ウエイトについて確認する。

事前学習回数が 0 の場合

事前学習回数が 0 の場合の初期ウエイトを図 7 に示す。漸近展開法 + DeepSolver 法_0 と MV+DeepSolver

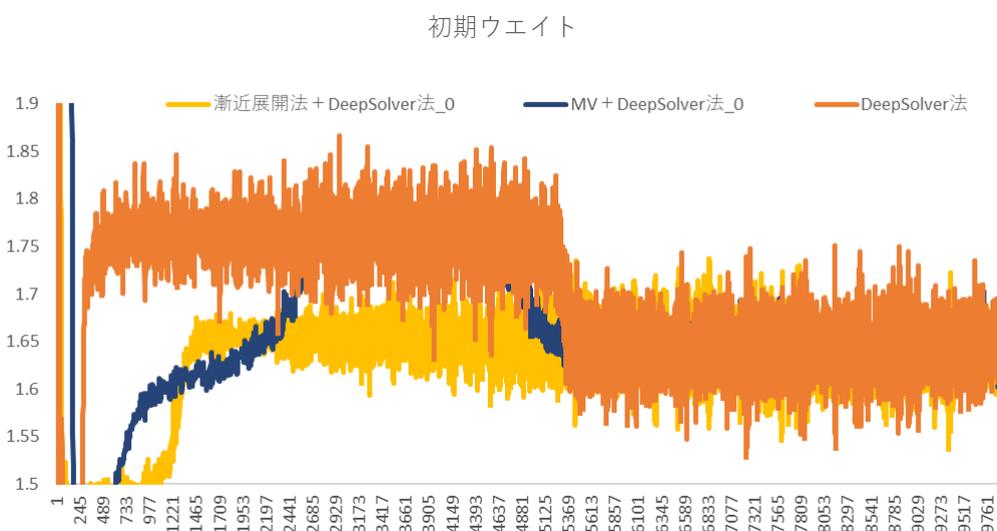


図 7: 初期ウエイト

DeepSolver 法・漸近展開法 + DeepSolver 法・MV+DeepSolver 法・DeepSolver+DeepSolver 法で解いた場合の初期ウエイトを示す。横軸はトレーニング回数で縦軸が初期ウエイトの値である。

法_0 はともに DeepSolver 法よりも速く収束していることが分かる。さらに、漸近展開法 + DeepSolver 法_0 はトレーニング回数が 1500 回程度で 1.65 の周りに収束しており他の手法よりも収束が速いことが確認できる。一方で、MV+DeepSolver 法_0 は安定した収束までに時間がかかる。

事前学習回数が 200 の場合

図 8 は事前学習回数が 200 の場合の初期ウエイトである。まず、漸近展開法 + DeepSolver 法_200 が最も速く収束しており、200 回前後のトレーニング回数ですでに 1.65 の周りに収束している。一方で、MV+DeepSolver 法_200 は DeepSolver+DeepSolver 法よりも収束が遅く、これは図 5 の損失関数と整合的な結果である。

事前学習回数が 2000 の場合

図 9 は事前学習回数が 2000 の場合の初期ウエイトである。図 6 と同様に各手法の間に収束の速さに差が生じていくようになっており、損失関数のときの結果と整合的である。

以上をまとめると、初期ウエイトにおいても損失関数と同様に漸近展開法 + DeepSolver 法が最も速く収束するような振る舞いを示しており、200 回程度の事前学習で効率的に初期ウエイトを求めることができている可能性がある。一方で、MV+DeepSolver 法や DeepSolver+DeepSolver 法は初期ウエイトの収束を速めるた

初期ウエイト

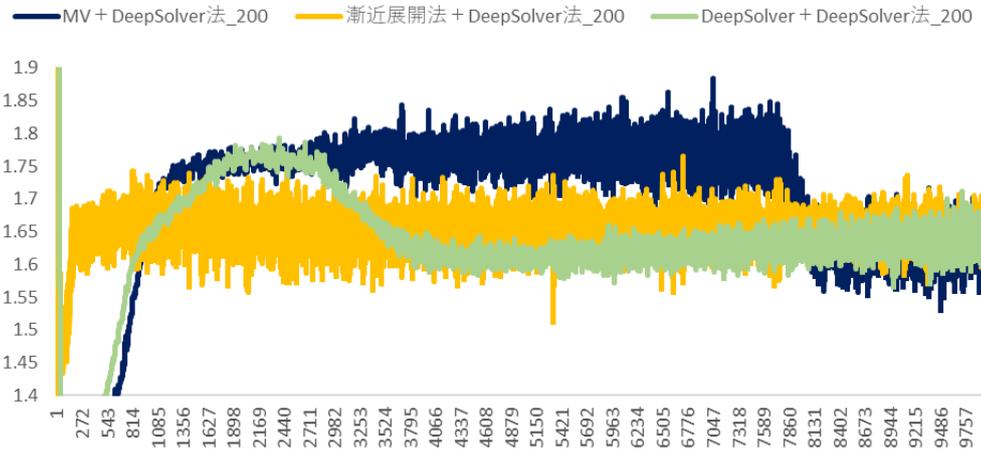


図 8: 初期ウエイト

DeepSolver 法・漸近展開法 + DeepSolver 法・MV + DeepSolver 法・DeepSolver + DeepSolver 法で解いた場合の初期ウエイトを示す。横軸はトレーニング回数で縦軸が初期ウエイトの値である。

初期ウエイト

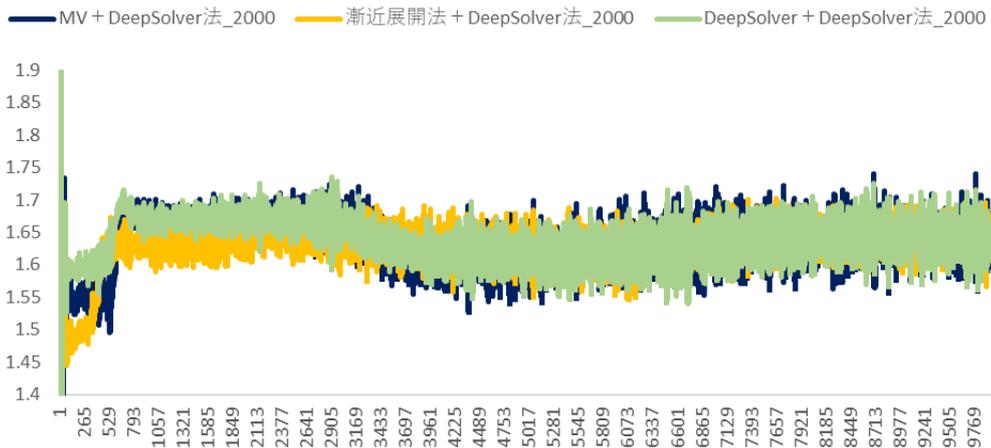


図 9: 初期ウエイト

DeepSolver 法・漸近展開法 + DeepSolver 法・MV + DeepSolver 法・DeepSolver + DeepSolver 法で解いた場合の初期ウエイトを示す。横軸はトレーニング回数で縦軸が初期ウエイトの値である。

めに 2000 回程度の事前学習を必要とする点も図 6 と同様の結果である。

■**アウトオブサンプルでの期待効用** 最後に、各手法のアウトオブサンプルでの期待効用を確認する。

事前学習回数が 0 の場合

事前学習回数が 0 の場合のアウトオブサンプルでの期待効用を図 10 に示す。漸近展開法 +DeepSolver 法_0

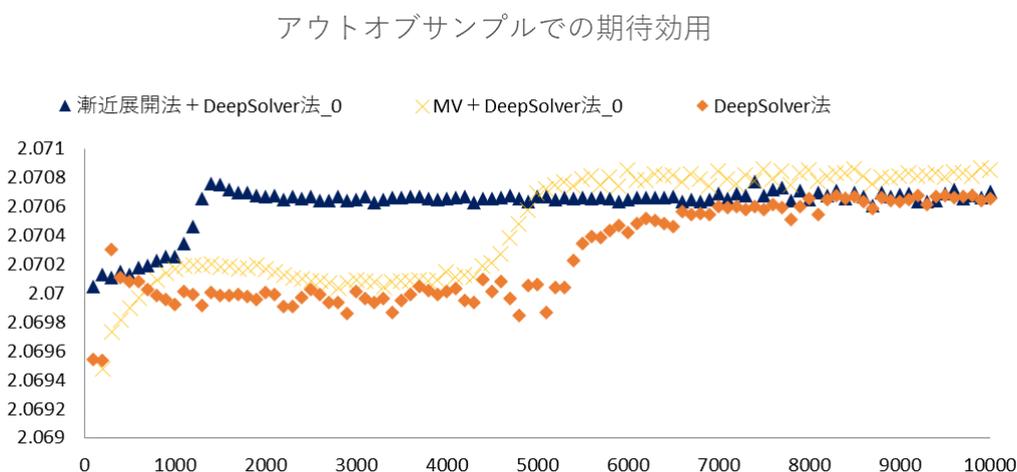


図 10: アウトオブサンプルでの期待効用

DeepSolver 法・漸近展開法 + DeepSolver 法・MV+DeepSolver 法・DeepSolver+DeepSolver 法で解いた場合のアウトオブサンプルでの期待効用を示す。横軸はトレーニング回数で縦軸がアウトオブサンプルでの期待効用の値である。

と MV+DeepSolver 法_0 はともに DeepSolver 法よりも高い値を取ることが分かる。漸近展開法 +DeepSolver 法_0 と MV+DeepSolver 法_0 を比較すると、漸近展開法 +DeepSolver 法_0 の方が 5000 回程度までのトレーニングでは高い値を取っており、少ないトレーニング回数においてパフォーマンスが良いことが分かる。一方で、MV+DeepSolver 法_0 は 5000 回以上のトレーニングを実施することで漸近展開法 +DeepSolver 法_0 と同程度の水準を達成するために 5000 回以上のトレーニングを必要とすることも確認できる。これは MV だけの情報ではなく漸近展開法の 1 次の情報を使うことでより効率的に最適ポートフォリオに近づけることを示唆していると考えられる。

事前学習回数が 200 の場合

図 11 は事前学習回数が 200 の場合のアウトオブサンプルでの期待効用である。図 10 と同様に漸近展開法 +DeepSolver 法_200 が最も速く高い値を取っていることが分かる。漸近展開法 +DeepSolver 法_200 は 200 回程度のトレーニングだけで最も高い値を取る一方で、DeepSolver+DeepSolver 法_200, MV+DeepSolver 法_200 が漸近展開法 +DeepSolver 法_200 と同程度の水準を達成するためには、それぞれ 4000 回、8000 回程度のトレーニングが必要であることも分かる。

事前学習回数が 2000 の場合

図 12 は事前学習回数が 2000 の場合のアウトオブサンプルでの期待効用である。損失関数 (図 6) や初期ウエイト (図 9) と同様に各手法間では差が生じにくくなっている。これは制約なし最適ポートフォリオ問題の例と同様にトレーニング回数が増加するに従い各手法の損失関数の値や初期ウエイトに差が生じていないことと整合的である。

アウトオブサンプルでの期待効用

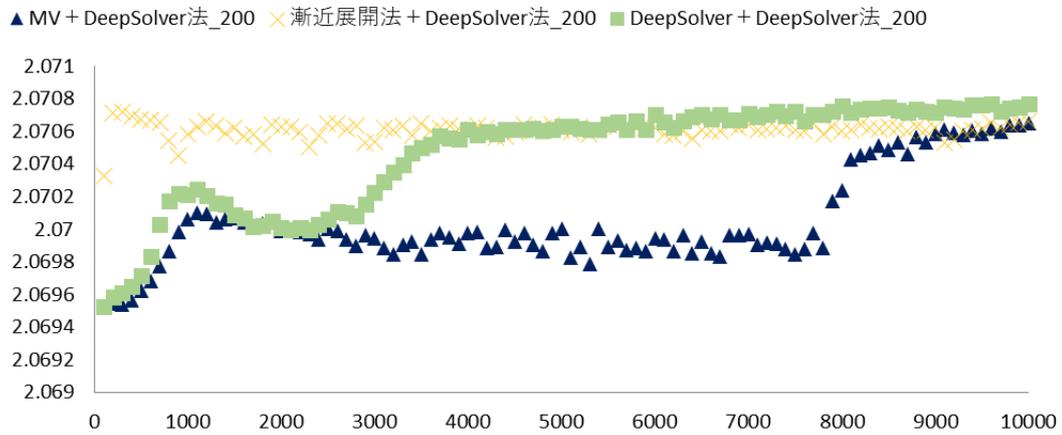


図 11: アウトオブサンプルでの期待効用

DeepSolver 法・漸近展開法 + DeepSolver 法・MV+DeepSolver 法・DeepSolver+DeepSolver 法で解いた場合のアウトオブサンプルでの期待効用を示す。横軸はトレーニング回数で縦軸がアウトオブサンプルでの期待効用の値である。

アウトオブサンプルでの期待効用

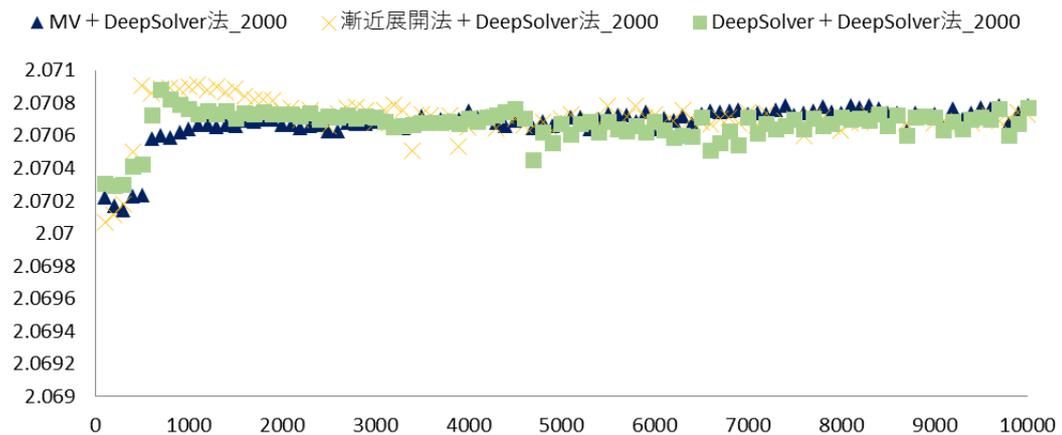


図 12: アウトオブサンプルでの期待効用

DeepSolver 法・漸近展開法 + DeepSolver 法・MV+DeepSolver 法・DeepSolver+DeepSolver 法で解いた場合のアウトオブサンプルでの期待効用を示す。横軸はトレーニング回数で縦軸がアウトオブサンプルでの期待効用の値である。

以上をまとめると、アウトオブサンプルでの期待効用においても漸近展開法 + DeepSolver 法が最も高いアウトオブサンプルでの期待効用を少ないトレーニング回数で達成することを示しており、特に 200 回程度の事前学習で効率的に最適ポートフォリオを求めることができている可能性があることを示唆している。一方で、MV+DeepSolver 法や DeepSolver+DeepSolver 法については、漸近展開法 + DeepSolver 法と同等の効率性を示すためには、2000 回程度の事前学習を必要とすることも確認できる。

6 頑健性

前節の結果の頑健性を確認するために、効用関数のリスク回避度 γ を変えて数値実験を行う。前節の例では $\gamma = 0.5$ としていたが、 $\gamma = 0.2$ に低下させる。リスク回避度 γ をゼロに近づけることで漸近展開法における MV 項以外の効果が弱まるため、MV+DeepSolver 法と漸近展開法 + DeepSolver 法の差が小さくなることが期待される。このような例においても本提案手法である漸近展開法 + DeepSolver 法のパフォーマンスの優位性が維持されるか否かを確認する。

■損失関数 まず、各手法の損失関数から確認する。

事前学習回数が 0 の場合

事前学習回数が 0 の場合の損失関数を図 13 に示す。漸近展開法 + DeepSolver 法_0 が最も収束が速く、最

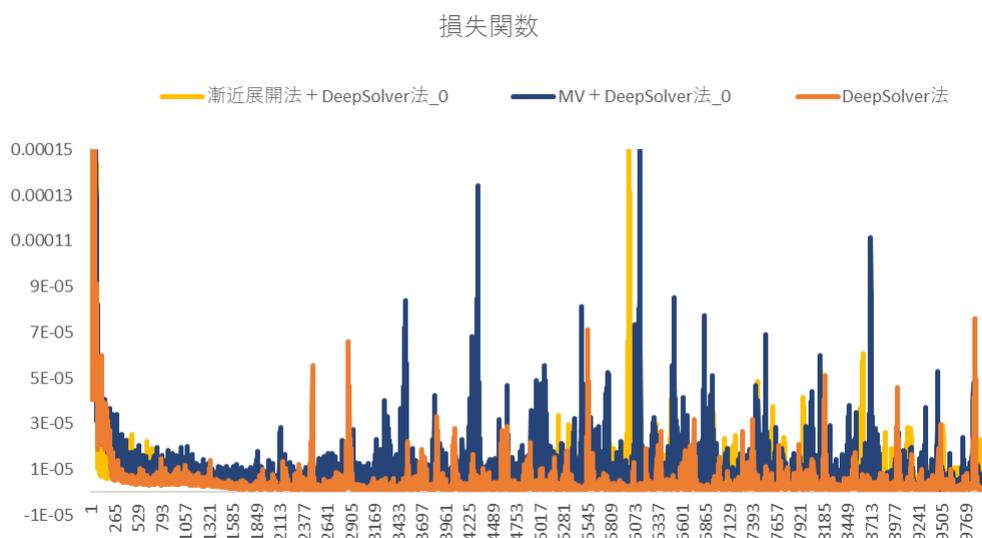


図 13: 損失関数

DeepSolver 法・漸近展開法 + DeepSolver 法・MV+DeepSolver 法・DeepSolver+DeepSolver 法で解いた場合の損失関数を示す。横軸はトレーニング回数で縦軸が損失関数の値である。

初の数回のトレーニングで DeepSolver 法よりも低い損失関数の値を達成していることが分かる。一方で、MV+DeepSolver 法_0 は DeepSolver 法と同程度の損失関数を達成するためには 500 解程度のトレーニングを要しており、事前知識として MV だけでなく漸近展開法の 1 次まで取り入れることでパフォーマンスが改善していることが分かる。

事前学習回数が 200 の場合

図 14 は事前学習回数が 200 の場合の損失関数である。漸近展開法 +DeepSolver 法_200 だけが 100 回

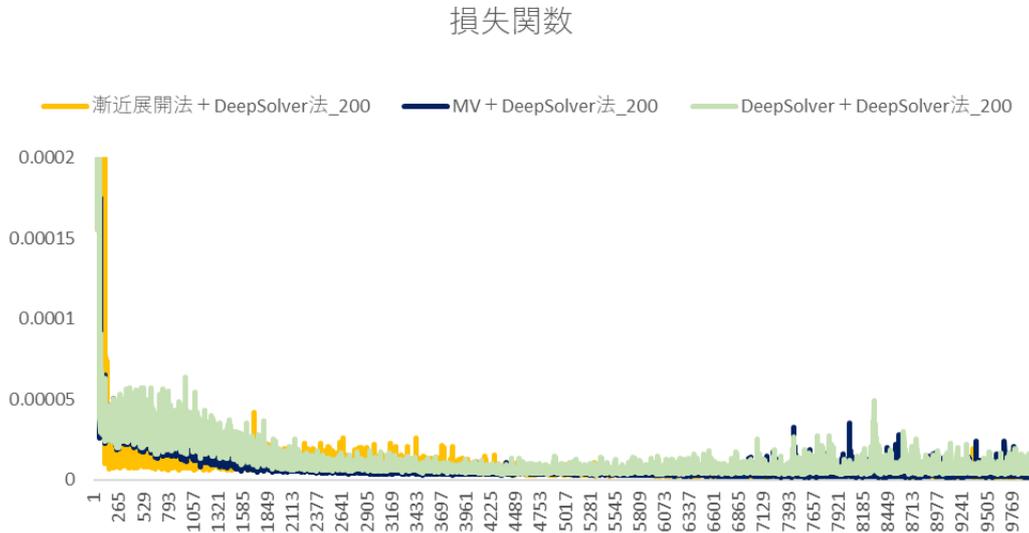


図 14: 損失関数

DeepSolver 法・漸近展開法+ DeepSolver 法・MV+DeepSolver 法・DeepSolver+DeepSolver 法で解いた場合の損失関数を示す。横軸はトレーニング回数で縦軸が損失関数の値である。

程度のトレーニングで速く収束していることが確認できる。一方で、MV+DeepSolver 法_200 や DeepSolver+DeepSolver 法_200 は漸近展開法 +DeepSolver 法と同程度の損失関数を達成するために 2000 回程度のトレーニングを要しており、漸近展開法 +DeepSolver 法_200 よりも効率的に損失関数を減少できていないことが分かる。

事前学習回数が 2000 の場合

図 15 は事前学習回数が 2000 の場合の損失関数である。図 13 や 14 と異なり、各手法の間に収束の速さや損失関数の水準に差が生じにくくなっている。これは例 1 の図 6 と同様の結果である。

以上をまとめると、損失関数においては前節と同様に漸近展開法 +DeepSolver 法が最も速く収束するような振る舞いを示しているが、一方で前節の例 ($\gamma = 0.5$) と比較すると、事前学習回数によらず各手法間に差が生じにくくなっていることが分かる。

■初期ウエイト 次に、各手法の初期ウエイトを確認する。

事前学習回数が 0 の場合

事前学習回数が 0 の場合の初期ウエイトを図 16 に示す。漸近展開法 +DeepSolver 法_0 は DeepSolver 法と比較して収束の速さは同程度だが変動幅が小さいように見える。一方で、MV+DeepSolver 法は漸近展開法 +DeepSolver 法_0 や DeepSolver 法よりも速く 0.7 付近を中心とした値を取るようになるが、変動幅は最も大きい。

事前学習回数が 200 の場合

図 17 は事前学習回数が 200 の場合の初期ウエイトである。漸近展開法 +DeepSolver 法_200 が最も速く収束しており、数回のトレーニング回数ですでに 0.70 の周りに収束している。一方で、MV+DeepSolver 法_200 は DeepSolver+DeepSolver 法_200 よりも収束が速いことが確認できる。これは図 14 の損失関数と整合的な

損失関数

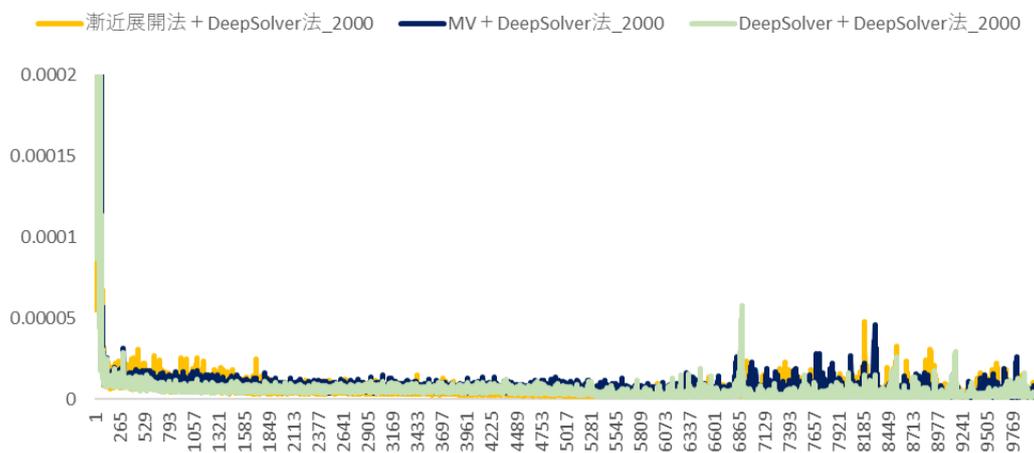


図 15: 損失関数

DeepSolver 法・漸近展開法 + DeepSolver 法・MV+DeepSolver 法・DeepSolver+DeepSolver 法で解いた場合の損失関数を示す。横軸はトレーニング回数で縦軸が損失関数の値である。

初期ウエイト

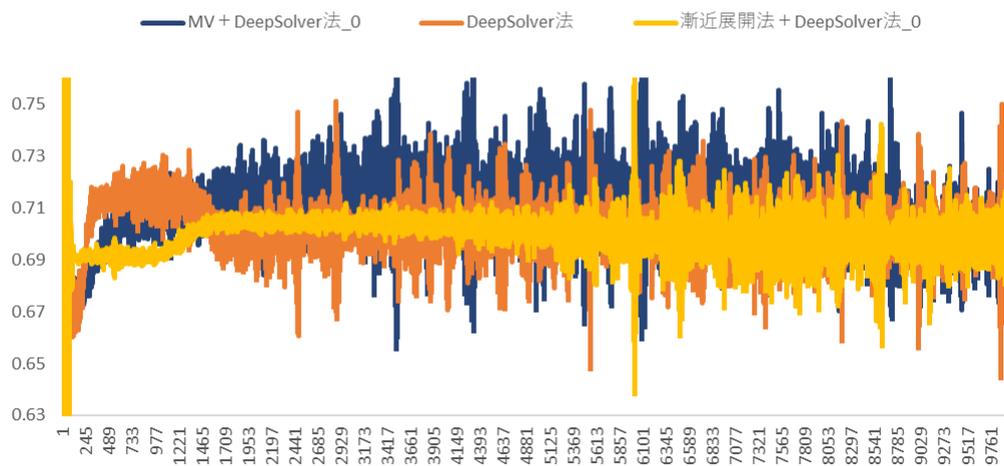


図 16: 初期ウエイト

DeepSolver 法・漸近展開法 + DeepSolver 法・MV+DeepSolver 法・DeepSolver+DeepSolver 法で解いた場合の初期ウエイトを示す。横軸はトレーニング回数で縦軸が初期ウエイトの値である。

初期ウエイト

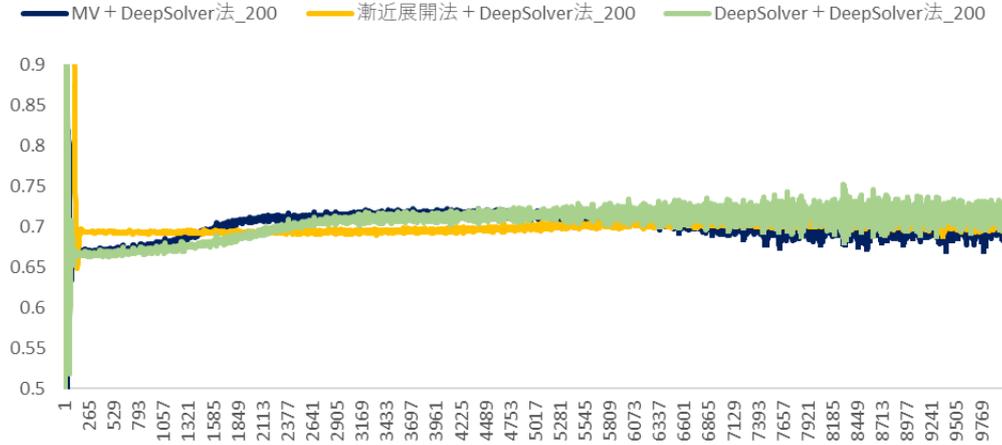


図 17: 初期ウエイト

DeepSolver 法・漸近展開法 + DeepSolver 法・MV+DeepSolver 法・DeepSolver+DeepSolver 法で解いた場合の初期ウエイトを示す。横軸はトレーニング回数で縦軸が初期ウエイトの値である。

結果である。

事前学習のトレーニング回数が 2000 の場合

図 18 は事前学習のトレーニング回数が 2000 の場合の初期ウエイトである。図 15 と同様に各手法の間に収

初期ウエイト

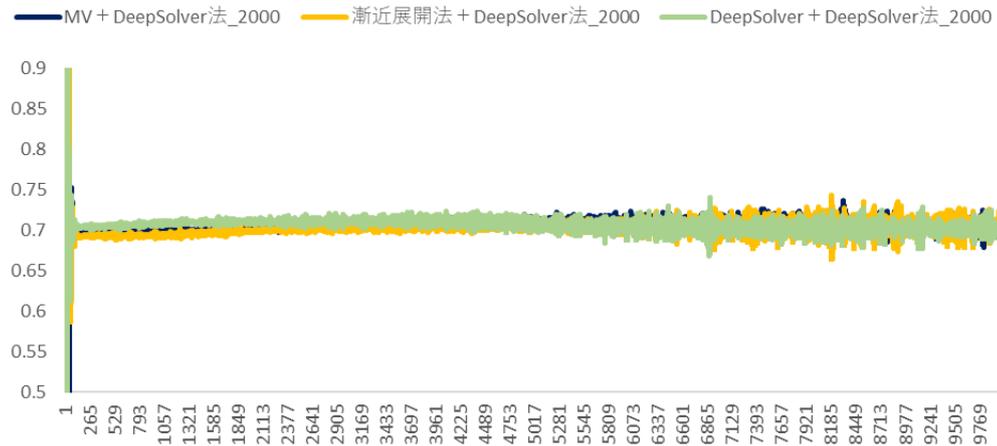


図 18: 初期ウエイト

DeepSolver 法・漸近展開法 + DeepSolver 法・MV+DeepSolver 法・DeepSolver+DeepSolver 法で解いた場合の初期ウエイトを示す。横軸はトレーニング回数で縦軸が初期ウエイトの値である。

束の速さに差が生じにくくなっている。

以上をまとめると、初期ウエイトにおいても損失関数と同様に漸近展開法 + DeepSolver 法が最も速く収束するような振る舞いを示しており、少ないトレーニングで効率的に初期ウエイトを求めている可能性がある。一方で、各手法間の差は前節の例と比較して小さくなっている点も図 15 と同様の結果である。

■ **アウトオブサンプルでの期待効用** 最後に、各手法のアウトオブサンプルでの期待効用を確認する。

事前学習回数が 0 の場合

事前学習回数が 0 の場合のアウトオブサンプルでの期待効用を図 19 に示す。漸近展開法 + DeepSolver

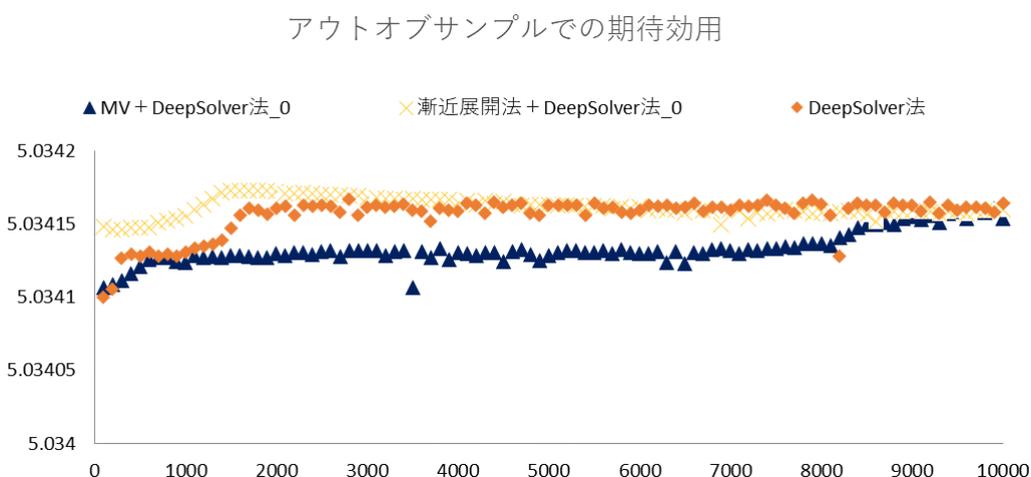


図 19: アウトオブサンプルでの期待効用

DeepSolver 法・漸近展開法 + DeepSolver 法・MV+DeepSolver 法・DeepSolver+DeepSolver 法で解いた場合のアウトオブサンプルでの期待効用を示す。横軸はトレーニング回数で縦軸がアウトオブサンプルでの期待効用の値である。

法_0 が少ないトレーニング回数で最も高い値を達成していることが分かる。一方で、MV+DeepSolver 法は DeepSolver 法と同程度の水準に到達するために 900 回程度のトレーニングを必要とすることが分かる。初期ウエイトでは漸近展開法 + DeepSolver 法と DeepSolver 法に収束の大きな違いはなかったが（図 16）、全期間の推定されたポートフォリオベースで計測するアウトオブサンプルでの期待効用では漸近展開法 + DeepSolver 法が DeepSolver 法よりも高いパフォーマンスを示していることが分かる。

事前学習回数が 200 の場合

図 20 は事前学習回数が 200 の場合のアウトオブサンプルでの期待効用である。図 19 と同様に漸近展開法 + DeepSolver 法_200 は少ないトレーニング回数で DeepSolver 法よりも高い値を取り、全トレーニング回数において DeepSolver 法と同水準の高さを達成している。一方で、MV+DeepSolver 法_200 は漸近展開法 + DeepSolver 法_200 と同水準の値を達成。また、DeepSolver+DeepSolver 法_200 は 10000 回トレーニングを実施後も漸近展開法 + DeepSolver 法_200 の水準に到達しておらずより多くのトレーニングが必要なことが確認できる。

事前学習回数が 2000 の場合

図 21 は事前学習回数が 2000 の場合のアウトオブサンプルでの期待効用である。漸近展開法 + DeepSolver 法_2000 は DeepSolver 法よりも高い値を少ないトレーニング回数で示している。損失関数（図 15）や初期ウ

アウトオブサンプルでの期待効用

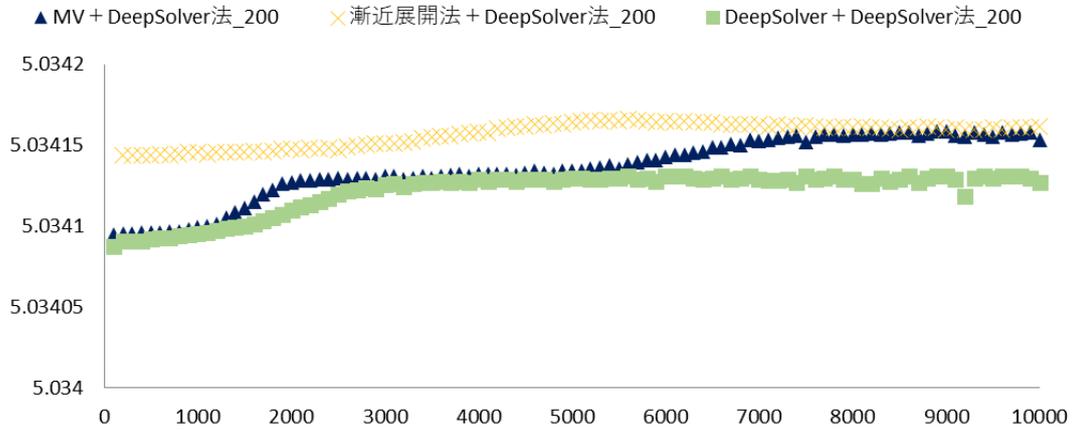


図 20: アウトオブサンプルでの期待効用

DeepSolver 法・漸近展開法 + DeepSolver 法・MV+DeepSolver 法・DeepSolver+DeepSolver 法で解いた場合のアウトオブサンプルでの期待効用を示す。横軸はトレーニング回数で縦軸がアウトオブサンプルでの期待効用の値である。

アウトオブサンプルでの期待効用

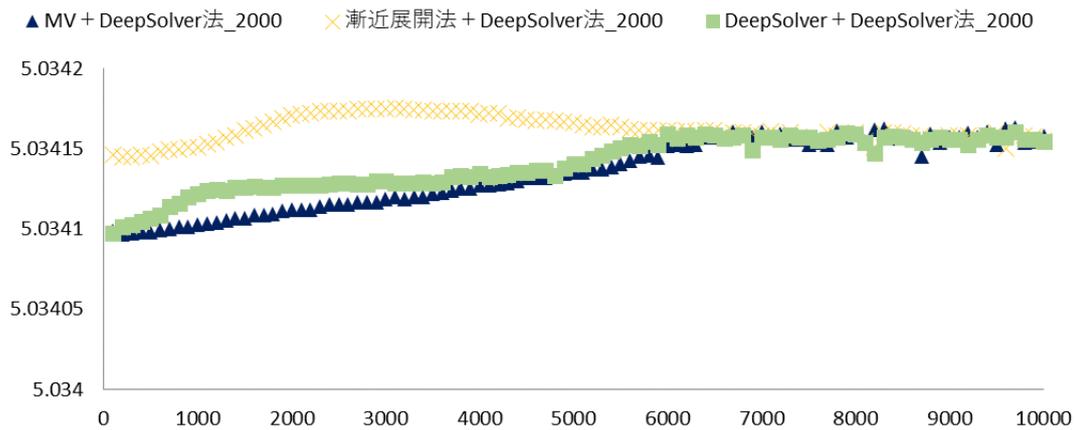


図 21: アウトオブサンプルでの期待効用

DeepSolver 法・漸近展開法 + DeepSolver 法・MV+DeepSolver 法・DeepSolver+DeepSolver 法で解いた場合のアウトオブサンプルでの期待効用を示す。横軸はトレーニング回数で縦軸がアウトオブサンプルでの期待効用の値である。

エイト (図 18) では事前学習を用いた手法: 漸近展開法 + DeepSolver 法_2000 や MV+DeepSolver 法_2000, DeepSolver+DeepSolver 法_2000 の間に違いを確認することはできなかったが, 最終的に推定されたポートフォリオが達成する期待効用で評価すると漸近展開法 + DeepSolver 法_2000 が最も良いパフォーマンスを示すことが分かる. 一方で, MV+DeepSolver 法_2000 と DeepSolver+DeepSolver 法_2000 は 6000 回以上のトレーニングを行うことで初めて DeepSolver 法や漸近展開法 + DeepSolver 法の水準に到達することが確認できる.

以上をまとめると, アウトオブサンプルでの期待効用においても漸近展開法 + DeepSolver 法が最も高いアウトオブサンプルでの期待効用を少ないトレーニング回数で達成することを示しており, 100 回程度の事前学習で効率的に最適ポートフォリオを求めている可能性があることを示唆している. 一方で, MV+DeepSolver 法や DeepSolver+DeepSolver 法は 2000 回の事前を行ったとしても漸近展開法 + DeepSolver 法と同程度の水準に達するにはかなりのトレーニングを行う必要があることが分かる.

7 結論

本稿では制約付き最適ポートフォリオ問題を効率的に解く新しい手法について取り上げた. 提案手法では, まず制約付き最適ポートフォリオ問題を Hu et al. (2005) の方法で BSDE に書き直し, その BSDE を制約なし最適ポートフォリオ問題に基づく BSDE(11) とそれ以外の BSDE(12) に分解する. そして, 制約なし最適ポートフォリオ問題に基づく BSDE に対して内藤・竹原 (2024) の手法を用いて事前学習を行い, その解を事前知識として制約付き最適ポートフォリオ問題の BSDE を DeepSolver 法で解く方法である. このとき, 内藤・竹原 (2024) の手法は Takahashi and Yoshida (2004) が提案した漸近展開法を用いて制約なし最適ポートフォリオ問題の BSDE を効率的に求めるものである. 数値例を通じた分析の結果, 本提案手法 (漸近展開法 + DeepSolver 法) は事前学習として 100 から 200 回程度の事前学習を行うことで, 主問題である制約付き最適ポートフォリオ問題の BSDE を他の手法と比較して効率的に解くことが確認できる. また, 本提案手法以外に事前知識獲得のために制約なし最適ポートフォリオ問題の BSDE(11) の解法に DeepSolver 法を適用する手法 (DeepSolver+DeepSolver 法) と漸近展開法の 0 次にあたる平均分散項のみを適用する手法 (MV+DeepSolver 法) についても分析した. その際に, DeepSolver+DeepSolver 法と MV+DeepSolver 法は 2000 回程度の事前学習が必要であり, 時間的な効率性を考慮すると漸近展開法 + DeepSolver 法の方がより高いパフォーマンスを示していることが分かる.

今後の課題としては漸近展開法の精度を上げることが考えられる. ただし, 先行研究では漸近展開法の次数を上げることによる改善効果は限定的であることが示唆されている. このほか, 実務的に重要なより多くの例を通して手法の有効性の頑健性を行うことも必要であると考えている.

参考文献

- 内藤誠・竹原浩太 (2024) 「制約なし最適ポートフォリオの機械学習による応用」, preprint.
- Cvitanic, J. and Karatzas, I. (1992) “Convex duality in constrained portfolio optimization,” *The Annals of Applied Probability*, 2, 767–818.
- Detemple, J. R., Garcia, J. R. and Rindisbacher, M. (2003) “A Monte Carlo Method for Optimal Portfolios,” *The Journal of Finance*, 58, 401–446.
- E, W., Han, J. and Jentzen, A. (2017) “Deep learning-based numerical methods for high-dimensional parabolic partial differential equations and backward stochastic differential equations,” *Commun. Math. Stat.*, 5(4), 349–380.
- Fujii, M. and Takahashi, A. (2012) “Analytical approximation for non-linear FBSDEs with perturbation scheme,” *International Journal of Theoretical and Applied Finance*, 15, 5, 1250034 (24).
- Fujii, M. and Takahashi, A. (2015) “Perturbative expansion technique for non-linear FBSDEs with interacting particle method,” *Asia-Pacific Financial Markets*, 22(3), 283–304.
- Fujii, M. and Takahashi, A. (2018) “Solving backward stochastic differential equations with quadratic-growth drivers by connecting the short-term expansions,” *Stochastic Processes and their Applications*, in press.
- Fujii, M. and Takahashi, A. (2018) “Quadratic-exponential growth BSDEs with jumps and their Malliavin’s differentiability,” *Stochastic Processes and their Applications*, 128(6), 2083–2130.
- Fujii, M. and Takahashi, A. (2019) “Asymptotic Expansion for Forward-Backward SDEs with Jumps,” *Stochastics*, 91(2), 175–214.
- Fujii, M. and Takahashi, A. (2019) “Asymptotic Expansion as Prior Knowledge in Deep Learning Method for High dimensional BSDEs,” *Asia-Pac Financ Markets*, 26, 391–408.
- Han, J., Jentzen, A. and E, H. (2018) “Solving high-dimensional partial differential equations using deep learning,” *Proc. Natl. Acad. Sci.*, 115(34), 8505–8510.
- Hu, Y., Imkeller, P. and Müller, M. (2005) “Utility Maximization in Incomplete Markets,” *The Annals of Applied Probability*, 15(3), 1691–1712.
- Karatzas, I., Lehoczky, J. P., Shreve, S. E. and Xu, G. L. (1991) “Martingale and duality methods for utility maximization in a incomplete market,” *SIAM J. Control Optimization*, 29, 702–730.
- Kim, Y. and Kunitomo, N. (1999) “Pricing options under stochastic interest rates,” *Asia-Pacific Financial Markets*, 6, 49–70.
- Kunitomo, N. and Takahashi, A. (1992) “Pricing average options,” *Jpn. Financial Rev.*, 14, 1–20.
- Kunitomo, N. and Takahashi, A. (2001) “The Asymptotic Expansion Approach to the Valuation of Interest Rate Contingent Claims,” *Mathematical Finance*, 11, 117–151.
- Kunitomo, N. and Takahashi, A. (2003) “On validity of the asymptotic expansion approach in contingent claim analysis,” *The Annals of Applied Probability*, 13.
- Li, C. (2014) “Closed-form Expansion, Conditional Expectation, and Option Valuation,” *Mathematics of Operations Research*, 39(2).
- Merton, R. C., (1969) “Lifetime portfolio selection under uncertainty: the continuous-time case,” *Review*

- of *Economics and Statistics*, 51, 247–257.
- Merton, R. C., (1971) “Optimum consumption and portfolio rules in a continuous-time model,” *Journal of Economic Theory*, 3, 373–413.
- Nishiba, M. (2013) “Pricing Exotic Options and American Options: A Multidimensional Asymptotic Expansion Approach,” *Asia-Pacific Financial Markets*, 20(2), 147–182.
- Ocone, D. and Karatzas, I., (1991) “A generalized clark representation formula, with application to optimal portfolios,” *Stochastics and Stochastics Reports*, 34, 187–220.
- Osajima, Y. (2006) “The Asymptotic Expansion Formula of Implied Volatility for Dynamic SABR Model and FX Hybrid Model,” *Preprint, Graduate School of Mathematical Sciences, the University of Tokyo*.
- Samuelson, P. A., (1969) “Lifetime Portfolio Selection by Dynamic Stochastic Programming,” *Review of Economics and Statistics*, 51, 239–246.
- Shiraya, K. and Takahashi, A. (2013) “Pricing Basket Options under Local Stochastic Volatility with Jumps,” *CARF-F*, 336.
- Shiraya, K. and Takahashi, A. (2014) “Pricing Multi-Asset Cross Currency Options,” *Journal of Futures Markets*, 34(1), 1–19.
- Shiraya, K., Takahashi, A. and Toda, M. (2011) “Pricing Barrier and Average Options under Stochastic Volatility Environment,” *Journal of Computational Finance*, 15(2), 111–148.
- Shiraya, K., Takahashi, A. and Yamazaki, A. (2011) “Pricing Swaptions under the LIBOR Market Model of Interest Rates with Local-Stochastic Volatility Models,” *Wilmott*, 2011(54), 61–73.
- Shiraya, K., Takahashi, A. and Yamada, T. (2012) “Pricing Discrete Barrier Options under Stochastic Volatility,” *Asia Pacific Financial Markets*, 19(3), 205–232.
- Takahashi, A. (1999) “An asymptotic expansion approach to Pricing contingent claims,” *Asia-Pacific Financial Markets*, 6, 115–151.
- Takahashi, A. and Takehara, K. (2007) “An Asymptotic Expansion Approach to Currency Options with a Market Model of Interest Rates under Stochastic Volatility Processes of Spot Exchange Rates,” *Asia-Pacific Financial Markets*, 14, 69–121.
- Takahashi, A. and Takehara, K. (2008a) “Fourier Transform Method with an Asymptotic Expansion Approach: an Applications to Currency Options,” *International Journal of Theoretical and Applied Finance*, 11(4), 381–401.
- Takahashi, A. and Takehara, K. (2008b) “A Hybrid Asymptotic Expansion Scheme: an Application to Currency Options,” *CARF-F*, 116.
- Takahashi, A. and Takehara, K. (2010a) “A Hybrid Asymptotic Expansion Scheme: an Application to Long-term Currency Options,” *International Journal of Theoretical and Applied Finance*, 13(8), 1179–1221.
- Takahashi, A. and Takehara, K. (2010b) “Asymptotic Expansion Approaches in Finance: Applications to Currency Options,” *Finance and Banking Developments*, 185–232.
- Takahashi, A. and Yoshida, N. (2004) “An Asymptotic Expansion Scheme for Optimal Investment Problems,” *Statistical Inference for Stochastic Processes*, 7, 153–188.
- Takahashi, A. and Yamada, T. (2012) “An Asymptotic Expansion with Push-Down of Malliavin Weights,” *SIAM Journal on Financial Mathematics*, 3, 95–136.

- Takahashi, A. and Yamada, T. (2011) “A Remark on Approximation of the Solutions to Partial Differential Equations in Finance,” *Recent Advances in Financial Engineering*, 3, 95–136.
- Takahashi, A. and Yamada, T. (2016) “An Asymptotic Expansion for Forward-Backward SDEs: A Malliavin Calculus Approach,” *Asia-Pacific Finan Markets*, 23, 337–373.
- Takahashi, A., Tsuchida, Y. and Yamada, T. (2021) “A new efficient approximation scheme for solving high-dimensional semilinear PDEs: control variate method for Deep BSDE solver,” *arXiv*.