

ニューラルネットワークを用いた 航空座席管理問題の新たな解法の開発

古田 匠¹・瀬木 俊輔²

¹学生非会員 名古屋大学 大学院工学研究科土木工学専攻 (〒464-8603 愛知県名古屋市中種区不老町)

E-mail: furuta.takumi.z6@s.mail.nagoya-u.ac.jp

²正会員 工博 神戸大学准教授 大学院工学研究科市民工学専攻 (〒657-0013 神戸市灘区六甲台町 1)

E-mail: segi@people.kobe-u.ac.jp

一般的に、航空会社は、キャンセルの制限やアメニティにより、複数の座席を異なる運賃クラスとして差別化している。そこで、航空会社は収益を最大化するように運賃クラスごとの販売座席数を最適化する必要がある、この最適化問題を航空座席管理問題と言う。この問題には顧客の需要とキャンセルの不確実性下が存在しており、厳密解を求めるのは困難である。よって、これまでに様々な近似解法が提案されてきたが、事実に反する仮定を置いている解法や、パフォーマンスの低い解法など、それぞれに問題点が存在していた。そこで、本研究ではこれらの問題を克服するため、ニューラルネットワークを用いた新たな解法を提案する。この解法では遺伝的アルゴリズムを用いることで局所的最適解へ収束しづらくし、また過学習を防ぐアルゴリズムを搭載することで解の安定性を高めることができると考える。検証の結果、提案した解法を用いることで既存の解法よりも優れた収益を生み出すことができ、また解の安定性も向上した。

Key Words: airline revenue management, neural networks, genetic algorithm, reinforcement learning

1. はじめに

座席管理問題とは、顧客需要の不確実性下で、運賃クラスごとの販売座席数を最適化する問題のことを示す。この問題は確率的動的最適化問題（現在から将来にわたって確率的に発生する事象に対して、各時点において実行すべき方策を最適化する問題）のひとつとして知られている。

座席管理問題において、航空会社は、収益を最大化するために顧客の選好に応じた適切な座席販売戦略を取る必要がある。一般的に、航空会社はキャンセルの制限やアメニティにより複数の座席を異なる運賃クラスとして差別化しており、顧客の選好に合った座席が選択可能となっている。例えば、ビジネス客（仕事による移動を目的とする客）は、キャンセル可能な高価な座席を好み、出発の直前に予約を行う傾向がある。それに対してレジャー客（旅行による移動を目的とする客）は、キャンセル不可能な安価な座席を好み、ビジネス客と比べて早くから予約を行う傾向がある。また、座席はフライト出発時までには販売しなければ無価値となることから「生鮮品

としての性質を持つため、高価な座席を多く用意し過ぎることで、空席が発生した状態でのフライトの出発を迎えるのは適切ではない。一方、安価な座席の数を多くし過ぎれば、潜在的な高価な座席の需要を受け付けることができない可能性があり、適切とは言えない。そのため、航空会社は最適な座席販売上限数を設定し、収益を最大化するように両者を適切に受け入れる必要がある。また、顧客はランダムに予約やキャンセルを行うため、座席管理問題には不確実性が存在する。航空会社はキャンセルの埋め合わせや、空席が発生した状態でフライトを迎えないようにするため、キャンセルを見越したオーバーブッキングを行うことが求められる。

実際の座席需要の特徴として、(1) 時間変化するランダムな顧客の需要、(2) クラスに依存した時間変化するキャンセル確率、が挙げられる。(1) は、顧客の需要は日々変動する上、ある一日における需要にもばらつきがあることを示す。(2) は、実際の顧客の行動を反映し、フライトの出発間近にキャンセル確率が増加するという示す。かつてはこれらの状況を厳密に再現できている研究は少なく、また、規模の小さな（航空

機の総座席数が極めて少ない条件下の) フライトでの分析しか行われていなかった。

しかし、近年では深層強化学習 (DRL) を利用することによって上記の条件を概ね満たした上での近似解の導出が可能となった。これらの研究では、ニューラルネットワーク (NN) を利用し、顧客の需要と予約開始からの時間を入力、座席の販売戦略を出力とし、これらの関係をエージェント (行動方針を決定する主体) に学習させ、最適な座席配分を導出することを目的としている。

しかし DRL には、局所的最適解に陥りやすい点や、学習データのみで過剰に適合したモデルを作ってしまう過学習と呼ばれる現象により、最適化が上手く行えない点、顧客のキャンセル確率増加などによる不確実性が増加した際に、学習の精度が落ちてしまう点が問題として挙げられる。よって、これらの問題点を克服するために、本研究では NN と遺伝的アルゴリズム (GA) を利用した新たな近似解法を開発することを目的とする。そして、開発した解法を、時間変化するランダムな顧客の需要と、クラスに依存した時間変化するキャンセル確率を伴う複数の運賃クラスのもとでの座席管理問題に適用し、本研究の手法の有用性を確認する。

2. 既往研究のレビュー

本節では、座席管理問題に対してどのような手法が適用されてきたかについてのレビューを行う。座席管理問題の研究の先駆けとして、Littlewood (1972)1) がレジャー客とビジネス客のみから構成される 2 期間の収益最大化問題の簡易解法を導出した。Belobaba (1987)2) は、Littlewood (1972)1) のモデルを多期間の場合について拡張することで、EMSR と呼ばれる解法を提案し、この解法は航空業界でも実際に用いられた。しかし、これら上記のモデルでは、高価な座席の販売は、安価な座席が販売された後に行われるという仮定がおかれていた。そこで、Lee and Hersh (1993)3) はこのような到着パターンの仮定を置かず、時間を離散的に扱い、動的計画法により問題を解いた。また、彼らは座席の需要が時間依存のポアソン分布となることを仮定し、後の多くの研究もこの需要分布を仮定した。しかし、動的計画法は、運賃クラス数の増加や航空機の総座席数の増加により解くのが困難になる上、この研究では依然としてキャンセルとオーバーブッキングを考慮しておらず、現実的な仮定を満たしていない。Subramanian et al. (1999)4) は Lee and Hersh (1993)3) のモデルを拡張し、キャンセルとオーバーブッキングを考慮した問題を動的計画法によって解いた。彼らの解法は実際の航空会社のデータにも適用可能であることが示されたが、彼らはキャンセル確率が運賃クラスに依存しない

ことを仮定していた。Gosavi et al. (2002)5) はキャンセル確率が運賃クラスに依存することを許容して、問題を準マルコフ決定過程としてモデル化し、強化学習を用いた λ -SMART と呼ばれるアルゴリズムを開発した。また、このアルゴリズムは EMSR 法の一種である EMSR-b 法よりも優れていることを示した。Gosavi et al. (2007)6) は、現実的な仮定の下、シミュレーション最適化により、予約可能期間の販売上限数を一律に最適化した。この手法では予約可能期間内の各時点において最適な販売上限数を導出することはできず、予約状況に応じた販売戦略をとることができないため、収益が最大化できていない可能性がある。また、彼らは多起点多終点型のネットワークに対しても分析を行っているが、本研究では扱わない。Lawhead et al. (2019)7) は、強化学習の一種であるアクター・クリティック・アルゴリズムを応用した解法を提案した。このアルゴリズムは、EMSR-b 法の性能を上回った。Shihab et al. (2022)8) は上記の研究とは違い、現実の航空需要と似た傾向を表現するために、時間変化するポアソン分布 (NHPP) (Weatherford et al. 1993)9) を仮定した上で、問題に深層強化学習 (DRL) を適用した。DRL とは、深層 NN により、状況 (座席の予約状況と経過時間) と行動 (座席販売戦略) の組み合わせに対する期待収益を近似し、最適な行動を取ることで最適戦略を実現する手法である。DRL エージェントはシミュレーションを繰り返すことで最適な戦略を学習することができ、この方法は EMSR-b 法の収益を上回ることが示された。

しかしこの研究では、運賃クラス数が少なく、運賃クラスごとに座席の需要タイミングが分離した設定で学習が行われており、複雑な条件に対応できない可能性がある。また、DRL ではエージェントの行動の価値を適切に表現するために、サンプリングされた中での最適行動価値と、現在の NN における行動価値との差に基づいたパラメータの勾配を用いて学習を行うため、局所的最適解に陥ってしまう可能性がある。また、NN の表現力が高い場合 (NN の隠れ層が多く、最適化するパラメータ数が多い場合)、NN が学習時のデータに過剰に適合した近似関数となってしまう過学習という現象が起り、テスト時に最適な行動を出力できなくなることがある。

3. 座席管理問題のモデルの定式化

(1) 座席配分モデル

本研究は、Shihab et al. (2022)⁸⁾ が提案したモデルを応用する。想定するフライトは、0 から $T-1$ 日目までの購入可能期間、総収容可能数 K 、運賃クラス数 n を仮定した、一起点・一終点のレグ (leg) とする。 t ($0 \leq t \leq T-1$) 日目において、顧客は座席の購入またはキ

キャンセルを行う。顧客がクラス i ($1 \leq i \leq n$) の座席を購入した場合、顧客は運賃 r_i を支払うため、航空会社は r_i の収益を得る。キャンセルが行われる場合、運賃クラスによってはキャンセル価格 m_i が発生するため、顧客は $r_i - m_i$ の払い戻しを受けてキャンセルを行う。よって、航空会社は $r_i - m_i$ の損失を被る。また、航空会社はキャンセルを見越したオーバーブッキングが可能である。フライト出発日にオーバーブッキングが発生したままの場合、航空会社は最も安い座席の客から順に搭乗を拒否し、補償費用 u_i を顧客に支払う。

したがって、航空会社は、座席の販売数、キャンセル数、オーバーブッキング数を考慮した上で、フライト出発日の期待収益を最大化することを目的として座席配分を行う。以上から、目的関数は以下の式で表すことができる。

$$\max_{\{b_{i,t}\}_{1 \leq i \leq n, 0 \leq t < T}} E_0 \left[\sum_{i=1}^n \left\{ \sum_{t=0}^{T-1} \{r_i y_{i,t} - (r_i - m_i) c_{i,t}\} - u_i \eta_i \right\} \right] \quad (3.1)$$

ここで、

r_i : クラス i の運賃

$y_{i,t}$: t 日目におけるクラス i の座席の販売数

m_i : クラス i のキャンセル価格

$c_{i,t}$: t 日目におけるクラス i の座席のキャンセル数

u_i : クラス i の補償費用

η_i : フライト出発日 ($t = T$) におけるクラス i の搭乗拒否数

$r_i y_{i,t}$ は座席の販売による収益、 $(r_i - m_i) c_{i,t}$ は座席のキャンセルによる損失、 $u_i \eta_i$ は搭乗拒否の補償費用である。本研究では、式(3.1)を目的関数として、 t 日目におけるクラスごとの座席の販売上限数 $b_{i,t}$ を最適化する。

(2) 需要発生モデル

顧客の座席購入は、顧客の需要に基づいて行われる。

t 日目におけるクラス i の需要数 $D_{i,t}$ は、平均 $\lambda_i(t)$ の時間変化するポアソン分布 (NHPP) を持つとする。

$\lambda_i(t)$ は以下の式から算出される。

$$\lambda_i(t) = \int_t^{t+1} \lambda'_i(\tau; \hat{\alpha}_i, \hat{\beta}_i) \hat{A}_i d\tau \quad (3.2)$$

ここで、

λ'_i : パラメータ $\hat{\alpha}_i$ と $\hat{\beta}_i$ を持つ、 τ 日目におけるベータ分布の確率密度関数

$\hat{\alpha}_i$: ベータ分布の形状パラメータ

$\hat{\beta}_i$: ベータ分布の形状パラメータ

\hat{A}_i : 各フライトでの合計平均需要数

ベータ分布は形状パラメータを変更することで様々な形状を示すことができるため、ベータ分布を用いることで顧客の需要パターンを柔軟に表現することが可能となる。ベータ分布の確率密度関数 $\lambda'_i(\tau)$ は以下のように定義される。

$$\lambda'_i(\tau) = \frac{\left(\frac{\tau}{T}\right)^{\hat{\alpha}_i-1} \left(1 - \frac{\tau}{T}\right)^{\hat{\beta}_i-1}}{B(\hat{\alpha}_i, \hat{\beta}_i)} \quad (3.3)$$

ただし、

$$B(\hat{\alpha}_i, \hat{\beta}_i) = \frac{\Gamma(\hat{\alpha}_i)\Gamma(\hat{\beta}_i)}{\Gamma(\hat{\alpha}_i + \hat{\beta}_i)} \quad (3.4)$$

ここで、

Γ : ガンマ関数

(3) キャンセル発生モデル

顧客は、購入した座席をキャンセルすることができる。 t 日目におけるクラス i のキャンセル数 $c_{i,t}$ は、 t 日目開始時点での座席の販売数 $x_{i,t}$ に対する、キャンセル確率 $p_{i,t}$ を平均とする二項分布により与えられる。

(4) 顧客の座席購入行動・キャンセル行動モデル

t 日目におけるクラス i の需要と t 日目におけるクラス i のキャンセルはランダムな順番で起こる。そこで、まず t 日目の顧客の総行動数 J_t を以下のように定義する。

$$J_t = \sum_{i=1}^n (D_{i,t} + c_{i,t}) \quad (3.5)$$

そして、 J_t 個の行動をランダムに並び替え、 j ($1 \leq j \leq J_t$) 番目の行動を $\text{type}(j)$ として表す。 $\text{type}(j)$ が需要であった場合、以下のようなモデルで座席の購入の可否が判断される。

$$\tilde{y}_{i,t}(j) = \begin{cases} \tilde{y}_{i,t}(j-1) + 1 & \text{if } \left(x_{i,t} + \tilde{y}_{i,t}(j-1) - \tilde{c}_{i,t}(j-1) \right) < b_{i,t} \\ \tilde{y}_{i,t}(j-1) & \text{(otherwise)} \end{cases} \quad (3.6)$$

ただし、

$$\tilde{y}_{i,t}(0) = 0 \quad (3.7)$$

$$y_{i,t} = \tilde{y}_{i,t}(J_t) \quad (3.8)$$

ここで、

$\tilde{y}_{i,t}(j)$: t 日目での、 j 番目の行動までのクラス i の販

売数

式(3.6)は、 t 日目、 $j-1$ 番目の行動までのクラス i の座席の販売数が販売上限数 $b_{i,t}$ 未満であるという条件を満たしたとき、座席が購入されるということを表す。

$\text{type}(j)$ がキャンセルであった場合、以下のような式で座席のキャンセルが行われる。

$$\tilde{c}_{i,t}(j) = \tilde{c}_{i,t}(j-1) + 1 \quad (3.9)$$

ただし、

$$\tilde{c}_{i,t}(0) = 0 \quad (3.10)$$

$$c_{i,t} = \tilde{c}_{i,t}(J_t) \quad (3.11)$$

ここで、

$\tilde{c}_{i,t}(j)$: t 日目での、 j 番目の行動までのクラス i のキャンセル数

(5) オーバーブッキングモデル

航空会社はキャンセルを見越したオーバーブッキングが可能である。フライト出発時にオーバーブッキングが発生したままの場合、航空会社は最も安い座席の客から順に搭乗を拒否する。

運賃クラスについて、 i の値が小さいクラスの方が運賃は高いとする。まず、フライト出発時の総オーバーブッキング数 η^T を以下の式から求める。

$$\eta^T = \max\left(\sum_{i=1}^n x_i - K, 0\right) \quad (3.12)$$

η^T は、オーバーブッキングが発生している場合は正の値を取り、オーバーブッキングが発生していない場合は 0 を取る。

次に、各運賃クラスの搭乗拒否数 η_i を求める。搭乗拒否は安い運賃クラスの客から順に、 $i = n, n-1, n-2, \dots, 2, 1$ という順番で起こっていき、合計で η^T 人だけ搭乗拒否が行われる。この過程は以下の式で表される。

$i = n$ のとき

$$\eta_n = \min\{\eta^T, x_{n,T}\} \quad (3.13)$$

$i \neq n$ のとき

$$\eta_i = \min\left\{\eta^T - \sum_{i'=i+1}^n \eta_{i'}, x_{i,T}\right\} \quad (3.14)$$

4. 座席管理問題の解法

(1) 提案する解法

本研究では、 t 日目において、 $m+1$ 個の状態変数 $\mathbf{s} = (t, \mathbf{x}) = (t, x_{1,t}, x_{2,t}, \dots, x_{m,t})$, すなわち、 $s_0 = t$, $s_1 = x_{1,t}$, $s_2 = x_{2,t}, \dots$, $s_m = x_{m,t}$ を入力、 n 個の操作変数 $\mathbf{b} = (b_{1,t}, b_{2,t}, \dots, b_{n,t})$ を出力とする NN に対して、GA を用いて重みとバイアス、並びに構造（隠れ層における活性化関数の数）を最適化し、収益を最大化する座席配分の近似解を、(5)で説明するアルゴリズムを用いて導出する。このような最適化手法はパラメトリック最適化と言われている。（これについては 4.4 節で詳しく説明する。）この手法では、従来の手法のように顧客の需要パターンの仮定を置く必要性がなく、クラスに依存した時間変化するキャンセル確率とオーバーブッキングを考慮することができる。また GA の利用により、DRL における局所的最適解への収束を克服することができると考えられる。さらに本研究では、NN の最適化アルゴリズムに過学習を防止するための工夫を施す。詳しくは (5)で述べる。

(2) 遺伝的アルゴリズムによるパラメータ最適化

強化学習により NN のパラメータを最適化する際には、NN で最適な行動価値を表現する必要がある。そこで、パラメータの勾配が用いられ、勾配が 0 となるようにパラメータが更新される。しかし、複雑な問題の場合、一般にパラメータの勾配が 0 となる点は複数存在しており、パフォーマンスの悪い局所的最適解に収束してしまうことがある。そこで、本研究では GA を用いてパラメータを最適化する。GA による最適化手順は以下の通りである。

1. 複数の個体 (NN のパラメータ) をランダムに生成し、これを 1 世代目とする。
- 2.1 高い収益を生み出す販売上限数を出力できた個体を複数抽出し、複製する (選択)。
- 2.2 選択個体同士の数値を混合 (交叉)、また個体の数値の一部を変更する (突然変異)。
1. 2 を繰り返す、最終世代において最も優れた販売上限数を出力できた個体を NN のパラメータとして採用する。

このように、GA では最適化アルゴリズムにおいて、勾配を用いておらず、また突然変異があるため、局所的最適解に収束しにくいと言える。

(3) 過学習について

(3)で用いる文字は過学習の説明のために用いられて

おり、他の節での文字とは関係がない。

まず、NN におけるパラメータ数について説明する。全結合、隠れ層無しの NN において、入力層のノード数が I 、出力層のノード数が K であるとする。この場合、重みの個数は $I \times K$ 個、バイアスの個数は K 個となる。よって最適化すべきパラメータ数の合計は、 $I \times K + K = (I + 1)K$ 個となる。また入力層と出力層の間に、ノード数が L 個の隠れ層を 1 層だけ挿入した場合、パラメータ数の合計は、 $I \times L + L + L \times K + K = (I + 1)L + (L + 1)K$ 個となる。このように、各層のノード数が積として考慮され、また層数が追加された分だけパラメータ数が加算される。よって、パラメータ数は各層のノード数と層数が多くなるにつれて多くなることが分かる。このようにしてパラメータ数が増えることで NN の表現力は豊かになり、優れた近似関数を生み出すことができる。しかし、表現力が豊かになりすぎると学習時のデータにのみ過剰に適合して過学習を引き起こしてしまう場合がある。そこで本研究では、最適化対象パラメータ数を減らすことで、過学習を防止することを試みる。過学習防止のアルゴリズムは、4(5)で併せて説明する。

(4) パラメトリック最適化

(4), (5)で用いる文字はアルゴリズムの説明のために用いられており、他の節での文字とは関係がない。

本研究では、 I 個の状態変数 $\mathbf{x} = (x_1, \dots, x_I)$ 、 K 個の操作変数 $\mathbf{y} = (y_1, \dots, y_K)$ を有する確率的動的最適化問題をパラメトリック最適化を用いて解く。パラメトリック最適化とは、確率的動的最適化問題のヒューリスティックな解法の一つである。この手法では、まず、状態変数 \mathbf{x} を入力、操作変数 \mathbf{y} を出力とするベクトル関数 $\mathbf{P}: \mathbb{R}_I \rightarrow \mathbb{R}_K$ を、パラメータ θ を含む形で定式化する。

$$\mathbf{y} = \mathbf{P}(\mathbf{x}, \theta) \quad (4.1)$$

この関数 \mathbf{P} は Policy function (政策関数) と呼ばれる。政策関数のパフォーマンスはパラメータ θ に依存する。このパラメータを、確率的動的最適化問題の目的関数を最大化するように決定するのがパラメトリック最適化の概略である。

確率的動的最適化問題においては、目的関数の期待値が最大化 (もしくは最小化) の対象となる。一般に、目的関数の期待値の厳密な評価は困難であり、その評価は乱数シミュレーションにより近似的に行わざるを得ない。ここでは、合計 M 回の乱数シミュレーションにより目的関数の期待値を評価するものとし、その評価値を、

$$V(\theta, \{\xi_m\}_{m=1}^M) = \frac{1}{M} \sum_{m=1}^M F(\mathbf{P}(\cdot, \theta), \xi_m) \quad (4.2)$$

で表す。ここで、 ξ_m は m 回目のシミュレーションで使用される乱数のベクトルである。関数 F は確率的動的最適化問題の目的関数を、政策関数 \mathbf{P} と乱数ベクトル ξ の関数として表したものである。航空座席管理問題の場合には、関数 F の値は航空会社の収益 (座席の売り上げからキャンセルされた座席に対する払い戻しと、出発時におけるオーバーブッキングの補償費用を差し引いたもの) を表す。政策関数 \mathbf{P} の関数形と乱数ベクトル ξ が具体的に与えられれば、目的関数値はシミュレーションにより確定的に評価できるため、関数 F が定義可能となる。乱数シミュレーションにより評価された目的関数の期待値は θ と $\{\xi_m\}_{m=1}^M$ に依存する。この依存関係を関数 V として表す。

パラメトリック最適化の基本的な考え方は、関数 V を θ について最適化することである。最適化の間は、乱数ベクトル $\{\xi_m\}_{m=1}^M$ の実現値を固定する。この理由は、目的関数の評価の度に乱数ベクトルを変化させると、目的関数の評価値が変動するため、最適化が進行しないことである。ただし、乱数の実現値を固定すると、 θ の最適値に固定された乱数の偏りが反映されるため、政策関数のパフォーマンスが悪化する可能性がある。また、一般に関数 V は、 θ についての勾配がほとんどの点においてゼロの関数になる (航空座席管理問題においてクラスごとの座席の販売上限が \mathbf{b} である場合、例えば、 θ が微小に変化して上限数が 10.3 から 10.4 になっても、航空会社の行動は変化しない)。そのため、関数 V の最適化には遺伝的アルゴリズムのような、目的関数の勾配に依存しない最適化手法を使わざるを得ない。ただし、このような最適化手法を用いると、パラメータ θ の次元が大きい場合には、目的関数値の改善が緩慢にしか進まないという問題が生じやすい。そこで、本研究では、以下に説明するアルゴリズムを採用して、パラメータ θ を最適化する。

(5) NN の最適化アルゴリズム

(4), (5)で用いる文字はアルゴリズムの説明のために用いられており、他の節での文字とは関係がない。

まず、二つのパラメータベクトル θ^A と θ^B があるとき、この二つのベクトルのパフォーマンスの優劣を比較するために、以下の指標を定義する。

$$G(\theta^A, \theta^B, \{\xi_m\}_{m=1}^{10M}) = \frac{\mu(\theta^A, \theta^B, \{\xi_m\}_{m=1}^{10M})}{\sqrt{\frac{s(\theta^A, \theta^B, \{\xi_m\}_{m=1}^{10M})}{10M}}} \quad (4.3)$$

ただし,

$$\mu(\boldsymbol{\theta}^A, \boldsymbol{\theta}^B, \{\boldsymbol{\xi}_m\}_{m=1}^{10M}) = \frac{1}{10M} \sum_{m=1}^{10M} [F(\mathbf{P}(:, \boldsymbol{\theta}^A), \boldsymbol{\xi}_m) - F(\mathbf{P}(:, \boldsymbol{\theta}^B), \boldsymbol{\xi}_m)] \quad (4.4)$$

$$s(\boldsymbol{\theta}^A, \boldsymbol{\theta}^B, \{\boldsymbol{\xi}_m\}_{m=1}^{10M}) = \frac{1}{10M-1} \sum_{m=1}^{10M} \left[\begin{array}{c} F(\mathbf{P}(:, \boldsymbol{\theta}^A), \boldsymbol{\xi}_m) \\ -F(\mathbf{P}(:, \boldsymbol{\theta}^B), \boldsymbol{\xi}_m) \\ -\mu(\boldsymbol{\theta}^A, \boldsymbol{\theta}^B, \{\boldsymbol{\xi}_m\}_{m=1}^{10M}) \end{array} \right]^2 \quad (4.5)$$

関数 G は, $\boldsymbol{\theta}^A$ と $\boldsymbol{\theta}^B$ の下での目的関数の期待値の差が 0 であるという帰無仮説の検定統計量を表す. 検定統計量は $10M$ 回の乱数シミュレーションにより評価される. 本研究の採用するアルゴリズムでは, 乱数ベクトル $\{\boldsymbol{\xi}_m\}_{m=1}^M$ の実現値を変更して最適化を繰り返す. 乱数ベクトルの実現値が変われば, 得られる $\boldsymbol{\theta}$ の最適値も異なったものになる. このとき, どちらの最適値がより優れているのかを判断するために, 関数 G を利用する.

次に, $\boldsymbol{\theta}$ を R 個のブロックに分割する: $\boldsymbol{\theta} = (\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \dots, \boldsymbol{\theta}_R)$. そのうえで, 各ブロックについて, 他のブロックのパラメータを固定し, 当該ブロックのみのパラメータを動かして最適化を行うという処理を実行する. これはベイズ推定の Gibbs sampling と同様の考えに基づく処理であり, 最適化の際に動かす変数の次元を減らすことにより, 目的関数値の改善が緩慢にしか進まないという問題を避ける. ただし, 各ブロックについての部分最適化のみを実行していると, パフォーマンスの悪い局所的最適解に陥る可能性がある. そこで, 全パラメータについての最適化も実行する.

以上の準備の下で, 本研究は以下の逐次的最適化アルゴリズムを採用する.

[逐次的最適化アルゴリズム]

1. $\boldsymbol{\theta}$ の初期値 $\boldsymbol{\theta}^0$ を定める
2. $\text{iter} \leftarrow 1$
3. $\text{terminate} \leftarrow \text{True}$
4. 乱数ベクトル $\{\boldsymbol{\xi}_m\}_{m=1}^M$ を新たに生成する
5. $\hat{\boldsymbol{\theta}}^{\text{iter},0} \leftarrow \arg \max_{\boldsymbol{\theta}} V(\boldsymbol{\theta}, \{\boldsymbol{\xi}_m\}_{m=1}^M)$
6. 乱数ベクトル $\{\boldsymbol{\xi}_m\}_{m=1}^{10M}$ を新たに生成する
7. $g \leftarrow G(\hat{\boldsymbol{\theta}}^{\text{iter},0}, \boldsymbol{\theta}^{\text{iter}-1}, \{\boldsymbol{\xi}_m\}_{m=1}^{10M})$
8. $g \geq 0$ なら $\boldsymbol{\theta}^{\text{iter},0} \leftarrow \hat{\boldsymbol{\theta}}^{\text{iter},0}$ とし, そうでなければ $\boldsymbol{\theta}^{\text{iter},0} \leftarrow \boldsymbol{\theta}^{\text{iter}-1}$ とする
9. $g \geq 1$ なら $\text{terminate} \leftarrow \text{False}$ とする
10. $r \leftarrow 1$
11. 乱数ベクトル $\{\boldsymbol{\xi}_m\}_{m=1}^M$ を新たに生成する

12. $\hat{\boldsymbol{\theta}}^{\text{iter},r} \leftarrow$

$$\arg \max_{\boldsymbol{\theta}_r} V \left((\boldsymbol{\theta}_1^{\text{iter},r-1}, \dots, \boldsymbol{\theta}_r, \dots, \boldsymbol{\theta}_R^{\text{iter},r-1}), \{\boldsymbol{\xi}_m\}_{m=1}^M \right)$$

13. 乱数ベクトル $\{\boldsymbol{\xi}_m\}_{m=1}^{10M}$ を新たに生成する
14. $g \leftarrow G(\hat{\boldsymbol{\theta}}^{\text{iter},r}, \boldsymbol{\theta}^{\text{iter},r-1}, \{\boldsymbol{\xi}_m\}_{m=1}^{10M})$
15. $g \geq 0$ なら $\boldsymbol{\theta}^{\text{iter},r} \leftarrow \hat{\boldsymbol{\theta}}^{\text{iter},r}$ とし, そうでなければ $\boldsymbol{\theta}^{\text{iter},r} \leftarrow \boldsymbol{\theta}^{\text{iter},r-1}$ とする
16. $g \geq 1$ なら $\text{terminate} \leftarrow \text{False}$ とする
17. $r = R$ なら 18 に進み, そうでなければ $r \leftarrow r + 1$ とし 11 に戻る
18. $\boldsymbol{\theta}^{\text{iter}} \leftarrow \boldsymbol{\theta}^{\text{iter},r}$
19. $\text{terminate} \leftarrow \text{True}$ なら $\boldsymbol{\theta}^{\text{iter}}$ を $\boldsymbol{\theta}$ の最適値として返却し, そうでなければ $\text{iter} \leftarrow \text{iter} + 1$ とし 3 に戻る

4 番から 9 番ではパラメータベクトル全体に関する処理を行っており, 10 番から 17 番では個別のブロックに関する処理を行っている. 4 番と 12 番では固定された乱数ベクトルを用いて最適化を行っている. 固定された乱数ベクトルを用いて最適化を行う場合, 得られる $\boldsymbol{\theta}$ の最適値のパフォーマンスは, 事前に得られている $\boldsymbol{\theta}$ の最適値のものよりも悪化する可能性がある. そこで, 7, 8 番と 14, 15 番では, $10M$ 回の乱数シミュレーションにより, 最適化前後の $\boldsymbol{\theta}$ のパフォーマンスを比較し, 優れている方を採用するようにしている. terminate はアルゴリズムの終了判定に用いる二値変数であり, 関数 G により評価される検定統計量が 1 以上になることがあれば, False が代入される. すなわち, 最適化により $\boldsymbol{\theta}$ のパフォーマンスが有意に改善していると判定される場合には, $\boldsymbol{\theta}$ にはまだ改善の余地があると考え, アルゴリズムを終了しない.

本研究は, 4 番と 12 番の最適化に Yoon and Kim (2011)¹²⁾ の実数値遺伝的アルゴリズムを採用する. また, 政策関数 \mathbf{P} に線形関数, もしくは, 一層の隠れ層を有する NN を採用する.

$$y_k = c_k + \sum_{i=1}^I v_{ki} z_i \quad (1 \leq k \leq K) \quad (4.6)$$

$$z_i = \begin{cases} x_i & (\text{if } U = 0) \\ \frac{\alpha_u}{\sum_{u'=1}^U \alpha_{u'}} h_{ui} & (\text{if } U \geq 1) \end{cases} \quad (1 \leq i \leq I) \quad (4.7)$$

$$h_{uj} = \text{relu} \left(b_{uj} + \sum_{i=1}^I w_{uji} x_i \right) \quad (1 \leq u \leq U, i \leq j \leq I) \quad (4.8)$$

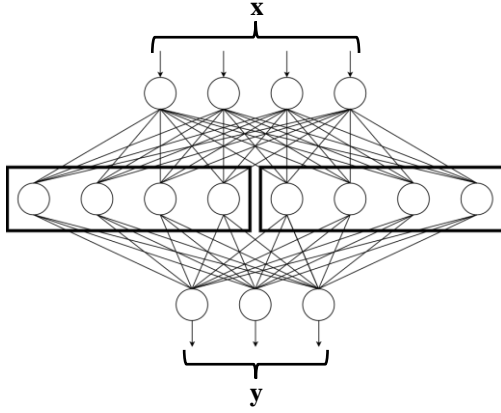


図4.1 $I = 4, K = 3$ で, $U = 2$ のときの NN の模式図

ここで,

$$\{c_k\}_{1 \leq k \leq K}, \{v_{ki}\}_{1 \leq k \leq K, 1 \leq i \leq I}, \{\alpha_u\}_{1 \leq u \leq U}, \{b_{uj}\}_{1 \leq u \leq U, 1 \leq j \leq I}'$$

$\{w_{uji}\}_{1 \leq u \leq U, 1 \leq j \leq I, 1 \leq i \leq I}$ はパラメータである.

$\text{relu}(\cdot)$ は ReLU 関数であり, 0内が正の値の場合にはその値を, そうでない場合には 0 を返す. 隠れ層は合計 U 個のユニットから構成される. 図 4.1 は, $I = 4, K = 3$ で, $U = 2$ のときの NN の模式図である. 各ユニットの入出力の次元は等しく, 入力 x_1, \dots, x_I であり, 出力は h_{u1}, \dots, h_{uI} である. 各ユニットの h_{ui} を重み α_u で合成した変数を z_i とし, これを線形出力層の入力として y_k の出力を得る. ただし, α_u は正のパラメータとし, 合計値が 1 になるように基準化する. $U = 0$ の場合には隠れ層は存在せず, 政策関数は線形の関数となる. なお, 状態変数 \mathbf{x} は全て 0 以上 1 以下の値に基準化してあるものとする.

各ユニットは $I + I^2$ 個のパラメータを有しており, I が大きい場合には最適化対象となるパラメータ数が多くなる. パラメータ数が多いと過剰適合を起しやすくなるため, 重み w_{uji} を以下のように表すことによりパラメータ数を抑える.

$$w_{uji} = \begin{cases} \beta_{ui} & (\text{if } i = j) \\ \gamma_{uj} \delta_{ui} & (\text{if } i \neq j) \end{cases} \quad (4.9)$$

ここで, $\beta_{ui}, \gamma_{uj}, \delta_{ui}$ は重み w_{uji} を決定するパラメータである. このように w_{uji} を表すことにより, 各ユニットのパラメータ数は $4I$ 個となる.

最適化アルゴリズムを使用する際には, パラメータをいくつかのブロックに分割する必要がある. 本研究は以下のように, 隠れ層のブロックごと, および, 出力層の出力ごとにブロックを設定する.

$$\theta_k = (c_k, \{v_{ki}\}_{1 \leq i \leq I}) \quad (1 \leq k \leq K) \quad (4.10)$$

$$\theta_{K+u} = (\alpha_u, \{b_{ui}, \beta_{ui}, \gamma_{uj}, \delta_{ui}\}_{1 \leq i \leq I}) \quad (1 \leq u \leq U) \quad (4.11)$$

以上の政策関数に対して逐次的最適化アルゴリズムを適用する. 本研究は, 以下のアルゴリズムに従い逐次的最適化アルゴリズムを利用することにより, 隠れ層のユニット数も適切な水準に設定することを試みる. 以下のアルゴリズムでは, 隠れ層のユニット数が U のときのパラメータベクトルを θ^U で表す.

1. $U \leftarrow 0$ (線形の政策関数) とし, 適当なパラメータ θ^0 の初期値を設定する.
2. 逐次的最適化アルゴリズムを適用し, 得られたパラメータの最適値を $\bar{\theta}^0$ とする.
3. $U \leftarrow 1$ とする.
4. パラメータ θ^U の初期値を以下のように設定する. θ^{U-1} と θ^U が共通して含む $\mathbf{c}, \mathbf{v}, \mathbf{b}, \boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\delta}$ の初期値は $\bar{\theta}^{U-1}$ と等しくする. 新たに追加されたパラメータ, および, $\boldsymbol{\alpha}$ の初期値は以下のように設定する.

$$\alpha_u^U \leftarrow 1, \beta_{ui}^U \leftarrow 1, \gamma_{uj}^U \leftarrow 0, \delta_{ui}^U \leftarrow 0 \quad (U = 1) \quad (4.12)$$

$$\alpha_u^U \leftarrow \begin{cases} \frac{\alpha_{u^*}^{U-1}}{2} & (u \in \{u^*, U\}), \\ \alpha_u^{U-1} & (u \notin \{u^*, U\}) \end{cases}$$

$$\beta_{ui}^U \leftarrow \beta_{u^*}^{U-1}, \gamma_{uj}^U \leftarrow \gamma_{u^*}^{U-1}, \delta_{ui}^U \leftarrow \delta_{u^*}^{U-1} \quad (U > 1) \quad (4.13)$$

ただし,

$$u^* = \underset{u}{\operatorname{argmax}} \alpha_u^{U-1} \quad (4.14)$$

5. 逐次的最適化アルゴリズムを適用し, 得られたパラメータの最適値を $\bar{\theta}^U$ とする.
6. $\bar{\theta}^{U-1}$ と $\bar{\theta}^U$ の下での目的関数の期待値の差が 0 であるという帰無仮説の検定統計量を 10M 回の乱数シミュレーションにより評価する.
7. 統計量が 1 未満なら $\bar{\theta}^{U-1}$ を採用して終了し, そうでなければ $U \leftarrow U + 1$ として 4 に戻る.

4 番のようにパラメータの初期値を設定するとき, ユニット数 U の政策関数の出力は $\bar{\theta}^{U-1}$ を採用したときのユニット数 $U - 1$ の政策関数の出力と全く同じになる.

これにより、単純な線形の政策関数から始めて、直前の最適化の結果を引き継ぎながら、徐々に政策関数の非線形性を強めていくことができる。ユニット数を増やしても目的関数の改善が見られなくなったら、7番で計算を打ち切る。

5. フライトの条件設定

本研究で提案する解法の有用性を確認するにあたり、2つの仮想的なフライト条件を設定し、それぞれフライト1、フライト2とする。

(1) フライト1の条件設定

$$\begin{aligned} T &= 182 \\ K &= 100 \\ n &= 3 \end{aligned}$$

表-5.1 運賃クラスごとの価格 r_i , キャンセル価格 m_i , 補償費用 u_i

運賃クラス番号 i	r_i (\$)	m_i (\$)	u_i (\$)
1	400	0	675
2	200	0	400
3	100	100	200

表-5.2 運賃クラスごとのキャンセル確率 $p_{i,t}$

運賃クラス番号 i	$p_{i,t} \quad 1 \leq t \leq 179$	$p_{i,t} \quad 180 \leq t \leq 181$
1	0.001	0.05
2	0.001	0.01
3	0.001	0.001

表-5.3 運賃クラスごとのベータ分布の形状パラメータ $\hat{\alpha}_i, \hat{\beta}_i$, 各フライトでの合計平均需要数 \hat{A}_i

運賃クラス番号 i	$\hat{\alpha}_i$	$\hat{\beta}_i$	\hat{A}_i
1	10.0	1.5	30
2	8.0	3.0	70
3	3.0	2.0	100

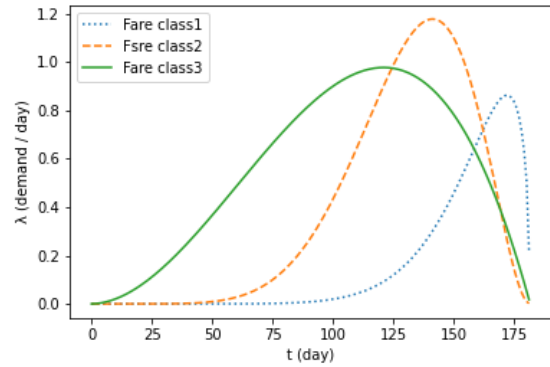


図-5.1 運賃クラスごとの需要分布

(2) フライト2の条件設定

$$\begin{aligned} T &= 182 \\ K &= 250 \\ n &= 10 \end{aligned}$$

表-5.4 運賃クラスごとの価格 r_i , キャンセル価格 m_i , 補償費用 u_i

運賃クラス番号 i	r_i (\$)	m_i (\$)	u_i (\$)
1	5120	0	10240
2	2560	0	5120
3	1280	100	2560
4	640	160	1280
5	320	160	640
6	160	100	320
7	80	60	160
8	40	35	80
9	20	20	40
10	10	10	20

表-5.5 運賃クラスごとのキャンセル確率 $p_{i,t}$

運賃クラス番号 i	$p_{i,t} \quad 1 \leq t \leq 179$	$p_{i,t} \quad 180 \leq t \leq 181$
1	0.001	0.05
2	0.001	0.01
3	0.001	0.001
4	0.001	0.001
5	0.001	0.001
6	0.001	0.001
7	0.001	0.001
8	0.001	0.001
9	0.001	0.001
10	0.001	0.001

表-5.6 運賃クラスごとのベータ分布の形状パラメータ

$\hat{\alpha}_i, \hat{\beta}_i$, 各フライトでの合計平均需要数 \hat{A}_i

運賃クラス番号 i	$\hat{\alpha}_i$	$\hat{\beta}_i$	\hat{A}_i
1	10.0	1.5	10
2	10.0	2.0	15
3	8.0	2.5	20
4	6.0	2.5	25
5	4.5	2.5	30
6	4.5	3.0	35
7	6.0	5.0	40
8	5.5	6.0	45
9	4.0	5.5	50
10	2.5	3.5	55

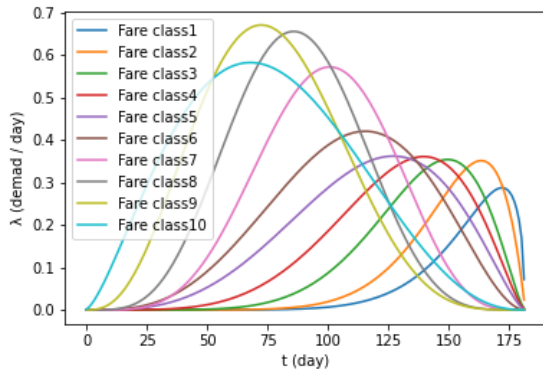


図-5.2 運賃クラスごとの需要分布

6. 分析について

(1) 解法の諸設定

a) 本研究の解法

本研究の解法は、プログラミング言語 Cython を用いてコーディングした。詳しい説明は4節で述べた。NNの最適化を行うにあたり、GAの1世代あたりの個体数を10個、GA内の目的関数の評価回数を500回、目的関数の改善判定に用いる閾値を1.0とした。その設定の下で、以下の試行-aを10回行った。

[試行-a]

4-(5)で説明したアルゴリズムを用いてNNを最適化し、最適化後のNNに対して10000回のテストを行い、目的関数の平均値(平均収益)と標準誤差を算出した。

b) Shihab et al. (2022)⁸⁾の解法

Shihab et al. (2022)⁸⁾の解法は、プログラミング言語 Python を用いてコーディングした。また、最適化には Keras (Chollet et al. (2015)¹⁰⁾ , Keras-rl (Plappert (2016)¹¹⁾ 等のライブラリを用いた。Keras-rlの特性上、NNの最

適化のためのシミュレーション回数を厳密に設定することはできないため、約1000回のシミュレーションが行えるように設定を行った。用いるNNは3層の隠れ層を持ち、隠れ層の各層の活性化関数は ReLU 関数を持つ。

NNの入力は本研究と同じで、状態 (t, \mathbf{x}) であり、出力は座席の販売の仕方としていた。具体的には、運賃クラス数が3(高運賃, 中運賃, 低運賃)のフライトの場合、出力1は高運賃クラスのみ販売、出力2は高・中運賃クラスのみ販売、出力3は全運賃クラスを販売としていた。このような出力では最適な戦略を取れる条件設定に限りがあるが、本研究では Shihab et al. (2022)⁸⁾の解法でも最適戦略を出力することが可能であるよう、フライト1, フライト2の条件を設定している。(本研究ではどのような条件でも対応可能である。)その他の詳細な設定は Shihab et al. (2022)⁸⁾の論文を参照されたい。以上の設定の下で、以下の試行-bを10回行った。

[試行-b]

シミュレーション回数約1000回でNNを最適化し、最適化後のNNに対して10000回のテストを行い、目的関数の平均値(平均収益)と標準誤差を算出した。

(2) フライト1の分析結果

本研究と Shihab et al. (2022)⁸⁾の解法について表にまとめた。

表-6.1 フライト1に対して本研究の解法を用いた結果

	テスト数	平均収益(\$)	標準誤差(\$)	U
1	10000	24143.45	16.05	2
2	10000	24105.57	16.59	2
3	10000	23939.88	16.62	1
4	10000	24113.62	16.65	1
5	10000	24100.51	16.54	1
6	10000	24131.15	16.67	2
7	10000	24124.21	16.38	2
8	10000	24128.88	16.75	2
9	10000	24042.17	16.82	1
10	10000	24121.14	16.28	5
平均		24095.06		

表-6.2 フライト 1 に対して Shihab et al. (2022)⁹⁾ の解法を用いた結果

	シミュレーション数	テスト数	平均収益 (\$)	標準誤差 (\$)
1	1008	10000	22067.15	10.26
2	1006	10000	21845.71	10.00
3	1003	10000	22094.82	11.05
4	1006	10000	23466.20	19.08
5	1005	10000	21767.30	11.06
6	1006	10000	19480.85	12.48
7	1007	10000	23301.54	20.74
8	1008	10000	18805.07	27.70
9	1006	10000	21938.72	11.82
10	1008	10000	23459.48	17.94
平均			21822.68	

a) 本研究の解法と Shihab et al. (2022)⁹⁾ の解法の結果の比較

本研究の解法を用いた場合、最も高い平均収益を記録できたのは 1 回目の試行で、\$24143.45 であった。Shihab et al. (2022)⁹⁾ の解法を用いた場合、最も高い収益を記録できたのは 4 回目の試行で、\$23466.20 であった。この結果は単に、Shihab et al. (2022)⁹⁾ の解法に比べて、本研究の解法が $24143.45 \div 23466.20 = 1.029$ 倍の収益を記録することができたことを示す。また、各回の試行において最適化された NN を用いて得られる収益を無限母集団として捉え、標本 10000 個の平均値がそれぞれ 24143.45, 23466.20 としてウェルチの t 検定を行った。その結果、有意水準 1% の片側検定で有意と判断され、母平均が等しいという帰無仮説は棄却された。よって、本研究の解法の優位性が示された。

また、10 回の試行の平均値は、本研究の解法では \$24095.06、Shihab et al. (2022)⁹⁾ の解法では \$21822.68 となった。 $24095.06 \div 21822.68 = 1.104$ であり、本研究の解法の方が安定して高い収益を生み出すことができると分かった。

b) 本研究の解法における NN の隠れ層ユニット数 U について

本研究の解法を用いた結果、 $U = 1, 2, 5$ となる場合があった。これらはいずれの場合も、Shihab et al. (2022)⁹⁾ の解法を用いた場合の 4 回目の試行での収益よりも高い収益を記録できた。よって、最適化時に入力された顧客の行動に基づき、隠れ層のユニット数も適切に最適化されたと考えられる。

(3) フライト 2 の分析結果

本研究と Shihab et al. (2022)⁹⁾ の解法について表にまとめた。

表-6.3 フライト 2 に対して本研究の解法を用いた結果

	テスト数	平均収益 (\$)	標準誤差 (\$)	U
1	10000	142837.67	193.11	1
2	10000	142812.05	189.70	2
3	10000	142580.24	192.05	4
4	10000	142120.11	189.88	0
5	10000	142449.92	191.01	1
6	10000	142667.24	190.65	4
7	10000	142507.43	191.87	2
8	10000	142516.50	191.37	0
9	10000	142351.87	191.81	1
10	10000	143028.37	193.35	3
平均		142587.14		

表-6.4 フライト 2 に対して Shihab et al. (2022)⁹⁾ の解法を用いた結果

	シミュレーション数	テスト数	平均収益 (\$)	標準誤差 (\$)
1	987	10000	141083.53	189.57
2	989	10000	45213.18	152.31
3	989	10000	137151.24	193.61
4	989	10000	136704.53	192.25
5	989	10000	142384.12	189.82
6	984	10000	141536.01	191.34
7	988	10000	140110.76	189.10
8	987	10000	137369.97	193.63
9	989	10000	141673.20	194.04
10	988	10000	139707.49	196.93
平均			130293.40	

a) 本研究の解法と Shihab et al. (2022)⁹⁾ の解法の結果の比較

本研究の解法を用いた場合、最も高い平均収益を記録できたのは 10 回目の試行で、\$143028.37 であった。Shihab et al. (2022)⁹⁾ の解法を用いた場合、最も高い収益を記録できたのは 5 回目の試行で、\$142384.12 であった。この結果は単に、Shihab et al. (2022)⁹⁾ の解法に比べて、本研究の解法が $143028.37 \div 142384.12 = 1.005$ 倍の収

益を記録することができたことを示す。また、ウェルチの t 検定で有意水準 1% で片側検定を行った結果、有意と判断され、母平均が等しいという帰無仮説は棄却された。よって、本研究の解法の優位性が示された。

また、10 回の試行の平均値は、本研究の解法では \$142587.14\$、Shihab et al. (2022)⁹⁾ の解法では \$130293.40\$ となった。 $142587.14 \div 130293.40 = 1.094$ であり、本研究の解法の方が安定して高い収益を生み出すことができると分かった。

b) 本研究の解法における NN の隠れ層ユニット数 U について

本研究の解法を用いた結果、 $U = 0, 1, 2, 3, 4$ となる場合があった。これらのうち、Shihab et al. (2022)⁹⁾ の解法を用いた場合の 5 回目の試行での収益よりも高い収益を記録できたのは $U = 2, 3, 4$ の時であり、隠れ層ユニット数が増加することで、より高い収益を記録することができたと考えられる。

7. おわりに

(1) 本研究のまとめ

本研究では、不確実性下における座席管理問題において、座席販売による収益最大化を目的として、NN のパラメータを GA を用いて最適化する解法を提案した。この解法では、簡単な条件（フライト 1）と複雑な条件（フライト 2）のどちらとも Shihab et al. (2022)⁹⁾ の解法より優れた結果を出せた。また、10 回の試行の収益の平均値では、いずれの条件でも本研究の解法の方が高くなっており、本研究の解法の方が安定して高い収益を生み出すことができることを示した。

以上から、本研究の解法は座席管理問題に対して有用であり、解の安定性の観点から、NN のパラメータ数を減らすことで過学習を防ぐことができると言える。また、NN のパラメータを GA を用いて最適化したことで、解の精度が安定し、局所的最適解への収束も克服することができると言える。

(2) 今後の課題

今後の課題として、複雑な条件でもより高い収益を記録できるよう、適切に NN のパラメータ数を設定することが挙げられる。また、今回は本研究の解法と既存の DRL を用いた解法の精度を比較したが、本研究の NN では座席の販売上限数を出力としていたのに対し、Shihab et al. (2022)⁹⁾ が用いた NN では座席の販売の仕方を出力としていたため、この出力の仕方をそろえて分析を行う必要性も考えられる。

参考文献

- 1) Littlewood, K.: Forecasting and control of passenger bookings, Airline Group International Federation of Operational Research Societies Proceedings, 12 th, pp. 95-117, 1972.
- 2) Belobaba, P. P.: Airline Yield Management An Overview of Seat Inventory Control, Transportation Science, Vol. 21, pp. 63-73, 1987.
- 3) Lee, C.K., and Hersh, M.: A Model for Dynamic Airline Seat Inventory Control with Multiple Seat Bookings, Transportation Science, Vol. 27, pp. 252-265, 1993.
- 4) Subramanian, J., Stidham Jr, S., and Lautenbacher, J.C.: Airline yield management with overbooking, cancellations, and no-shows. Transportation science, Vol. 33, pp. 147-167, 1999.
- 5) Gosavi, A., Bandla, N., and Das, K.T.: A reinforcement learning approach to a single leg airline revenue management problem with multiple fare classes and overbooking, Institute of Industrial Engineers (IIE) Transactions, Vol. 34, pp. 729-742, 2002.
- 6) Gosavi, A., Ozkaya, E. and Kahraman, F.A.: Simulation optimization for revenue management of airlines with cancellations and overbooking, OR Spectrum, Vol. 29, pp. 21-38, 2007.
- 7) Lawhead, J.R., and Gosavi, A.: A bounded actor-critic reinforcement learning algorithm applied to airline revenue management, Engineering Applications of Artificial Intelligence, Vol. 82, pp. 252-262, 2019.
- 8) Shihab, S., and Wei, P.: A deep reinforcement learning approach to seat inventory control for airline revenue management, Journal of Revenue and Pricing Management Vol. 21, pp. 183-199, 2022.
- 9) Weatherford, Lawrence, R., Samuel, E.B., and Phillip, E.P.: Modeling the customer arrival process and comparing decision rules in perishable asset revenue management situations, Transportation Science, Vol. 27, pp. 239-251, 1993.
- 10) Chollet, F. et al.: Keras, 2015.
- 11) Plappert, M. et al.: Keras-rl, 2016.
- 12) Yoon, Y. and Kim, Y.: The Roles of Crossover and Mutation in Real-Coded Genetic Algorithms, 2011.

Using neural networks to develop a new solution method for the airline revenue management problem

Takumi Furuta, Shunsuke SEGI

Generally, airlines differentiate multiple seats in different fare classes by cancellation restrictions and amenities. Therefore, airlines need to optimize the number of seats sold per fare class to maximize revenue. This optimization problem is called the airline revenue management problem. It is difficult to find an exact solution because of the uncertainties in customer demand and cancellations. Therefore, various approximate solution methods have been proposed, but each method has its own problems, such as adopting unrealistic assumptions or having poor performance. In this study, we propose a new solution method using a neural network to represent the policy function of the problem. The proposed method uses simulation optimization to tune the parameters of the neural network. It utilizes genetic algorithm as the optimization routine to prevent convergence to a poor local optimal solution and over-learning, thereby increasing solution stability. The results show that the proposed solution method generates better revenues than existing methods and improves the stability of the solution.