

ACOによる多目的要求に対応した旅行計画最適化手法

佐伯 越志^{1,a)} 鮑 思雅^{1,b)} 高山 敏典² 戸川 望^{1,c)}

概要: 観光産業の振興と情報科学技術の発展によって, 旅行計画サービスの開発が進んでいる. 旅行計画サービスが対象とする旅行計画では, 満足度や費用など複数の目的関数を同時に最適化することで, ユーザが満足する経路を生成する必要がある. とりわけ, 過去に多くのユーザが同様な旅程を計画している, あるいは部分的に同様な旅程を計画していることから, いかに過去のユーザの旅行経路を再利用するかが旅行計画の大きな鍵となる. 本稿では, 旅行計画に対するユーザの要求を満足するため, 多目的オリエンテーリング問題をベースに過去のユーザの旅行経路を陽に利用した旅行計画最適化手法を提案する. 提案手法は, 蟻コロニー最適化を利用することで, 過去のユーザの旅行経路を陽に反映した旅行計画を可能とする. その上で, 蟻コロニー最適化において蟻の行動を多様な目的関数に対応して変化させることで, 多目的オリエンテーリング問題を解法する. 評価実験により, 既存手法に対し, 過去の旅行者の旅行経路に近く, よりユーザの要求を満足する旅行経路を生成した.

1. はじめに

1.1 旅行計画問題

近年, 観光産業の振興と情報科学技術の発展によって, ユーザがいくつかの条件を指定するだけで自動で最適な旅程を計画できる, 旅行計画サービスの開発が進んでいる [1], [2]. 旅行計画サービスが対象とする旅行計画では, 満足度や費用など複数の価値を同時に最適化することで, ユーザが満足する経路を生成する必要がある.

例えば, 鮑ら [3] は, 観光地が持つ価値として満足度と費用, 滞在時間を設定している. 一方, 磯田ら [4] は観光地の持つ価値を観光地コンテキストと定義し, 嗜好度や滞在時間を挙げている. 鮑らや磯田らの研究が示すように, 観光地が持つ価値は1種類ではない. 観光地は満足度や費用のように複数の価値を持つため, 旅行計画では, 合計満足度の最大化や合計費用の最小化のようにそれぞれの価値の最適化が不可欠となる. 本稿では, 旅行計画において合計満足度の最大化と合計費用の最小化に焦点を当て, これを旅行計画最適化の第一条件と呼ぶ.

一方, ユーザの行動原理は多岐にわたり, 第一条件だけでユーザの要求を満足することは事実上不可能である. 例えば, 屋外で体力を消耗する観光地の次に比較的近くの屋

内観光地を選択する等, ユーザの要求を数値化することが困難な場合がある. また, 嗜好, 観光地のジャンル, 滞在時間などの数値化可能な目的関数であっても, 目的関数の数が多くなれば, それをすべて満足することは難しい. これらの要求を満足するには, 陽に過去のユーザが辿った旅行経路 (以降, 旅行履歴と呼ぶ) を利用すれば良い.

過去のユーザの旅行履歴を利用した手法として以下のものがある. 笠原ら [5] は過去のユーザの旅行履歴から旅行者の行動をモデル化し, 旅行履歴を考慮しないモデルと比較している. 旅行者が次に訪問する観光地を予測する評価実験において, 笠原らのモデルは比較モデルより 10.4%高い精度で予測している. さらに, 旅行経路の生成において笠原らのモデルを利用した結果, 既存手法と比較して過去のユーザの旅行履歴との類似度の高い経路が生成できている. 評価実験における予測精度の向上や, 生成経路の類似度の高さから, 過去のユーザの旅行履歴の利用が, ユーザの行動原理の反映に有効であると言える. Luら [6] は, 観光経路に過去のユーザの旅行履歴を参照したスコアを付加し, 旅行計画問題を解法している. Luらの手法によって生成された経路は, 旅行履歴を考慮しない手法と比較して, 多様性や満足度が高いことが示されている. これらの事例から, 旅行計画では, 過去のユーザの旅行履歴の利用が不可欠であると言える. これを旅行計画最適化の第二条件と呼ぶ.

ユーザの多様な要求を満足するように旅行計画問題を解法するためには, 第一条件ならびに第二条件を満足することが強く求められる. 文献 [3] や文献 [4] は, 多目的で経路

¹ 早稲田大学
Waseda University

² 株式会社ゼンリンデータコム
ZENRIN DataCom

a) etsushi.saeki@togawa.cs.waseda.ac.jp

b) siya.bao@togawa.cs.waseda.ac.jp

c) togawa@togawa.cs.waseda.ac.jp

を最適化しているため、第一条件を満足している。また、移動履歴を考慮した嗜好などの価値の設定はあるが、旅行履歴の旅行計画への陽な反映はなく、第二条件の満足として不十分である。文献 [5] や文献 [6] は第二条件を満足するが、必ずしも費用の最小化や満足度の最大化などの多様な目的関数を取り入れておらず、第一条件を満足しない。我々の知る限り、第一条件と第二条件を十分に満足する旅行計画最適化手法は知られていない。

1.2 多目的オリエンテーリング問題と蟻コロニー最適化による解法

旅行計画問題は、一般的にオリエンテーリング問題 (Orienteering Problem (OP))[7] に帰着される。すなわち、観光地の集合と観光地間の経路、全体の制限時間と観光地ごとの価値が与えられたとき、制限時間の中で、出発地から目的地まで価値を最大化するよう観光地を巡る経路を求めればよい。

Souffriau ら [8] は、OP を利用して旅行計画問題を定義し、guided local search [9] という手法を応用して最適経路を構築する手法を提案している。しかし、観光地への価値の付加は 1 種類のみであるため、第一条件を満足していない。同時に、ユーザの興味を考慮した観光地の価値決定はあるものの、過去のユーザの旅行履歴を利用した経路などへの価値の付加はないため、第二条件も満たしていない。Souffriau らの研究のように、多くの OP は観光地に 1 種類の価値のみを与えるため、第一条件を満足せず、旅行計画に対するユーザの要求を十分に反映できない。

一方、OP に複数の価値を与える問題として、多目的オリエンテーリング問題 (Multi Objective Orienteering Problem (MOOP)) [10] が提案されている。MOOP は各観光地に複数の独立した価値が与えられ、価値ごとに設定される目的関数をすべて考慮した経路を構築する。複数の目的関数を考慮できるため、第一条件を満たす。

Schilde ら [10] は、MOOP を解く手法として P-ACO と P-VNS の 2 つの手法を提案している。それぞれ、蟻コロニー最適化 (Ant Colony Optimization (ACO))[11] と可変近傍探索法 (Variable Neighborhood Search (VNS))[12] を多目的最適化に適用できるように改良した手法である。これらの手法によって、2 種類の価値をランダムで観光地に付加した MOOP に対し、最適解に近い解が得られている。Schilde らの研究では多目的最適化を実現しており、第一条件を満たしている。しかし、観光地間の経路の価値は距離のみであり、過去のユーザの旅行履歴の利用はなく、第二条件を満足していない。Martin ら [13] は、Multi-Objective Artificial Bee Colony Algorithm (MOABC) という、ミツバチの採餌行動を参考にした最適化手法を用いて複数の目的関数を持つ MOOP を解法する手法を提案している。MOABC を [10] と同一の問題セットに適用し、他手法と比

較した結果、よりユーザの要求を満足する経路を構築できることが示されている。しかしながら、Martin らの手法も MOOP を利用するため、第一条件を満たすが、過去のユーザの旅行履歴の利用による経路などへの価値の付加がなく、第二条件を満たしていない。以上のように、MOOP を利用する手法は第一条件を満足するものの、第二条件である過去のユーザの観光履歴を考慮できず、旅行計画に対するユーザの要求を十分に満足できない。

ここで、第一条件および第二条件を満足するために、ACO を利用して MOOP を解法することを考える。ACO は複数の蟻が同時に経路を構築するため、各蟻の行動を目的関数に応じて変更することで、多目的最適化を達成できると期待できる。また、ACO はフェロモンを媒介して過去の蟻が通った経路を訪問しやすくなり、蟻が経路を逐次的に探索するという特徴がある。この特徴を利用することで、フェロモンの付加によって過去のユーザの旅行履歴を陽に旅行計画に反映可能であると予想できる。フェロモンは任意の経路に付加できるため、出発地から目的地まで連続しない不完全な旅行データや、ある観光地に行った旅行者の多くが次に別の観光地を選択する、などの部分的なデータであっても反映できる。その結果、上述の第一条件および第二条件を同時に満足し、定義 1 の目的関数 (a)~(d) それぞれを最適化する経路の生成が期待できる。

1.3 本稿の提案

本稿では、ACO による多目的要求に対応した旅行計画最適化手法を提案する。提案する旅行計画最適化手法は、

- (1) 各蟻の行動の変更により多目的最適化を実現し (第一条件の満足),
- (2) フェロモンの付加などによって過去のユーザの旅行履歴を陽に反映可能とする (第二条件の満足).

提案手法では、まず対象となる観光エリアをもとにグラフを構成し、過去のユーザの旅行履歴に基づき初期フェロモンをグラフに付加する。その後、複数の蟻により、複数個の経路を構築する。多目的要求に対応した経路を構築するため、各蟻は目的関数に対応して行動原理が異なる。各蟻が出発地から目的地までの経路を構築した後、経路中の観光地の入れ替えやフェロモン情報の更新、目的関数に応じた経路情報の保存などの処理を適用する。最終的に目的関数に応じた経路を選定し、ユーザに提示する。その結果、上述の第一条件および第二条件を満足する旅行計画の生成を実現する。

1.4 本稿の貢献

本稿の貢献は以下の通りである。

- (1) MOOP に ACO を適用し、蟻が複数の目的関数に対応した行動をとることで、多様な要求を満足する旅行経路を同時に求めることができる。

- (2) その際、過去のユーザの旅行履歴に基づきフェロモンを付加することで、ユーザの過去履歴を陽にアルゴリズムに取り入れることができる。
- (3) 京都で 150 ヶ所、札幌で 100 ヶ所の観光地に関するデータセットを用意し、既存手法 [10] との比較実験を実施した。その結果、既存手法に対し、過去の旅行者の旅行経路に近く、よりユーザの要求を満足する旅行経路を生成した。

2. 旅行計画問題の定義

本章では旅行計画問題を定義する。観光エリアから完全グラフ $G = (V, E)$ を構成する。 V はノード集合であり、各ノードは観光地、出発地、目的地を表す。特に出発地と目的地は $v_{st} \in V$ と $v_{dst} \in V$ とする。観光地を表すノードを観光地ノードと呼び、その集合を V_p と書く。 E はエッジ集合であり、2つのノード $v_i, v_j \in V$ 間の経路を表す。観光地ノード $v_i \in V_p$ は、満足度 s_i 、滞在時間 d_i 、費用 c_i が与えられる。各エッジ $(v_i, v_j) \in E$ は移動時間 t_{ij} が与えられる。過去のユーザの旅行履歴を $Tour^{past} = [v_1, \dots, v_k]$ とする。ここで $[v_1, \dots, v_k]$ ($k \geq 2$) は、観光地ノードの列を表し v_1 から順に v_k まで訪問することを表す。

以上のもと、旅行計画問題を次のように定義する。

定義 1 観光エリアを表す完全グラフ $G = (V, E)$ 、制限時間 T 、 l 個の過去のユーザの旅行履歴が与えられたとき、出発地 v_{st} から各観光地ノードを高々 1 回経由して目的地 v_{dst} へ至る経路のうち、滞在時間および移動時間の合計が制限時間 T 以下であり、目的関数 (a) 合計満足度の最大化、(b) 合計費用の最小化、(c) 過去のユーザの旅行履歴との類似度の最大化、(d) (a) から (c) のすべて、を考慮する複数の経路を求める。合計満足度、合計費用とは、構築した経路で訪問した各観光地ノードの満足度および費用をそれぞれ合計したものである。過去のユーザの旅行履歴との類似度は、過去のユーザの旅行履歴と構築した経路の類似度の値である (詳細は 3.5.3 節を参照)。

旅行計画問題の入出力の例を以下に示す。

例 1 表 1 および図 1 に示す観光地ノードを結ぶ完全グラフ、制限時間 $T = 3$ 時間、 $[v_d, v_c, v_{dst}]$ という過去のユーザの旅行履歴が入力として与えられた場合、図 2 のような各目的関数に対応した経路を出力する。 ■

3. ACO による多目的要求に対応した旅行計画最適化手法

本章では、過去のユーザの旅行履歴を陽に考慮した旅行計画最適化手法を提案する。

3.1 戦略

ACO により MOOP を効率よく解法することを考える。この際、(A) フェロモンの付加により過去のユーザの旅行

表 1: 入力例.

| 観光地ノード | 満足度 | 費用 | 滞在時間 |
|--------|-----|-----|------|
| v_a | 1 | 2 | 1 |
| v_b | 3 | 1 | 1 |
| v_c | 3 | 0.5 | 1 |
| v_d | 4 | 1 | 1 |
| v_e | 3 | 0.5 | 1 |
| v_f | 5 | 1 | 1 |
| v_g | 4 | 1.5 | 1 |
| v_h | 4 | 0.5 | 1 |
| v_i | 1 | 1 | 1 |
| v_j | 2 | 2 | 1 |

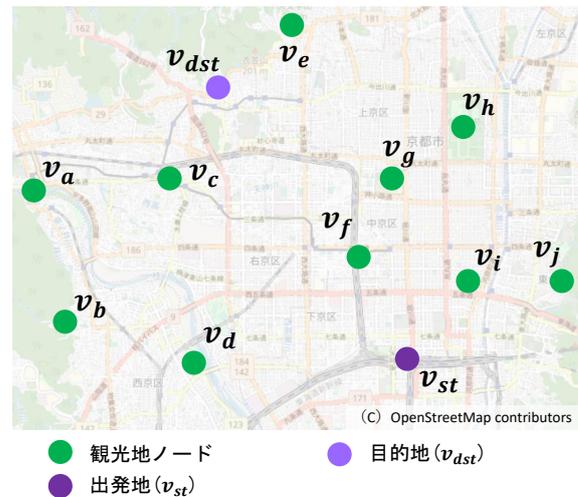


図 1: 入力する観光地ノードの例.

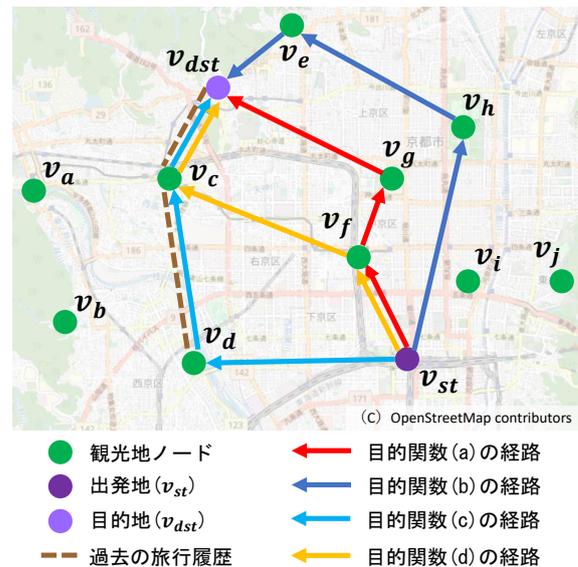


図 2: 出力の例.

履歴と類似度が高い旅行計画を生成すると同時に、(B) 蟻の行動によって複数の目的関数を同時に最適化する。

(A) の実現に最も単純な方法として、過去のユーザの旅行経路と同じ経路に初期フェロモンを付加する手法が考え

られる。しかし、ACO ではフェロモンは時間経過とともに減衰し、フェロモンの更新によって初期フェロモンの影響が小さくなる。そこで、提案手法では初期フェロモンの付加の他に、蟻が経路を構築するごとに構築した経路と過去のユーザの旅行履歴との類似度を計算し、類似度に応じてフェロモンを再度、付加する戦略を取る(詳細は3.5.3節を参照)。経路の構築手順を繰り返すごとに過去のユーザの旅行履歴を考慮した量のフェロモンを付加するため、過去のユーザの旅行履歴を十分に反映できると期待できる。

(B)の実現に対して、一般に複数の目的関数はトレードオフの関係にあり、すべての目的関数を最適化する解を求めることは困難である。そのため、パレート解と呼ばれる目的関数ごとの重みが異なる解を複数求めることとする。この際、各パレート解が局所解に収束せず、パレート解全体ですべての目的関数がバランスよく考慮されることが求められる。一方、ACOによりMOOPを解法すると、観光エリアを表すグラフ全体にフェロモンが共有され、ある解が局所解に陥るとこれがすべての解に影響する。この問題を解決するため、提案手法では局所探索やパス再結合といった手法によって解の多様性を保ち、局所解を回避する戦略をとる(詳細は3.5.1節および3.6.1節を参照)。

その結果、提案手法は上述の第一条件および第二条件を同時に満足し、定義1の目的関数(a)~(d)それぞれを最適化する経路の生成が期待できる。

3.2 提案手法の流れ

以上の議論のもと、提案手法の流れを図3および以下に示す。

(Step 1) 初期化

過去のユーザの旅行履歴に基づきグラフに初期フェロモンを付加する。

(Step 2) 蟻による経路構築

蟻に目的関数を設定する。蟻は、目的関数とフェロモンによって観光地を選択し、出発地 v_{st} から目的地 v_{dst} までの経路を構築する。

(Step 3) 局所探索・フェロモン更新

構築した経路に対し、それぞれの目的関数の局所的最適化のため、局所探索を適用する。その後、ローカル更新・類似度更新によりフェロモンを更新する。経路構築回数が m 回に満たない場合、Step 2の経路構築を繰り返す。

(Step 4) パス再結合・フェロモン更新

パレート解の集合に解を追加するためにパス再結合を適用する。その後、グローバル更新によりフェロモンを更新する。Step 2~Step 4を N 回繰り返し実行する。

(Step 5) 経路選定・提示

各目的関数(a)~(d)に応じた経路を選定し、ユーザに

提示する。

以下ではStep 1~Step 5の各ステップを提案する。

3.3 初期化 (Step 1)

本手順ではフェロモン情報を初期化する。提案手法は第二条件を満足する経路を構築するために、過去のユーザの旅行履歴に基づいて初期フェロモンを付加する。入力されたすべての過去のユーザの旅行履歴 $Tour^{past}$ に対し、履歴中の連続した2つの観光地の組 $v_a, v_b \in V_P$ に対応したエッジ (v_a, v_b) に、 I_{old} の量のフェロモンを付加する。 I_{old} はパラメータであり、値が大きいほど過去のユーザの旅行履歴への依存度が大きくなる。提案手法では、予備実験のもと $I_{old} = 0.6$ とした。この初期化手順によって、経路の構築手順で蟻が過去のユーザの旅行履歴に類似した経路を構築する。

例えば、図3中[Step 1]に示すように、A, B, C, Dという観光地があり、過去のユーザの旅行履歴として $A \rightarrow B \rightarrow C$ という経路のデータがあった場合、図中緑色で示すように経路に対してフェロモンを付加する。

3.4 蟻による経路構築 (Step 2)

本手順では、1匹の蟻が観光地を逐次的に移動することで出発地 v_{st} から目的地 v_{dst} までの経路を構築する。Step 2は、Step 3とともに m 回実行される。今、 k 回目の実行で構築された経路、すなわち k 番目の蟻が構築した経路を $Tour^k$ とする。 $Tour^k$ は出発地 v_{st} から始まり、経路の合計時間 T_k が制限時間 T を超えない限り観光地を移動し、目的地 v_{dst} へ到達することで、経路が構築される。観光地 v_i から観光地 v_j を訪れるごとに移動時間 t_{ij} と滞在時間 d_j が T_k に加算される。同様に、経路の構築時には合計満足度 S_k 、合計費用 C_k が計算される。それぞれ観光地 v_i から観光地 v_j を訪れるごとに満足度 s_j 、費用 c_j が加算される。

蟻 k が現在滞在している観光地から次の観光地を選択する際の確率を状態遷移ルールと呼ぶ。蟻ごとに状態遷移ルールを設定することで、各目的関数(a)~(d)を考慮した多目的最適化を実現する。

今、蟻 k がノード v_i に滞在しているとす。提案手法は、次のように蟻 k が次に訪問する観光地ノード v_j を決定する。

- (1) 確率 q_0 で、 v_i と隣接する未訪問ノードのうち、後述する $p_k(i, j)$ の値が最も高くなる観光地ノード v_j を訪問する。
- (2) 確率 $(1 - q_0)$ で、 v_i と隣接する未訪問ノードのうち、確率 $p_k(i, j)$ で観光地ノード v_j を訪問する。

上記のように、適当な確率 q_0 をベースに確率的に次訪問ノードを決定することで、確率によって選択される新しい経路と、フェロモンや観光地の価値から良いと予測される経路を効率よく探索でき、多様な解の構築に繋がる。予備

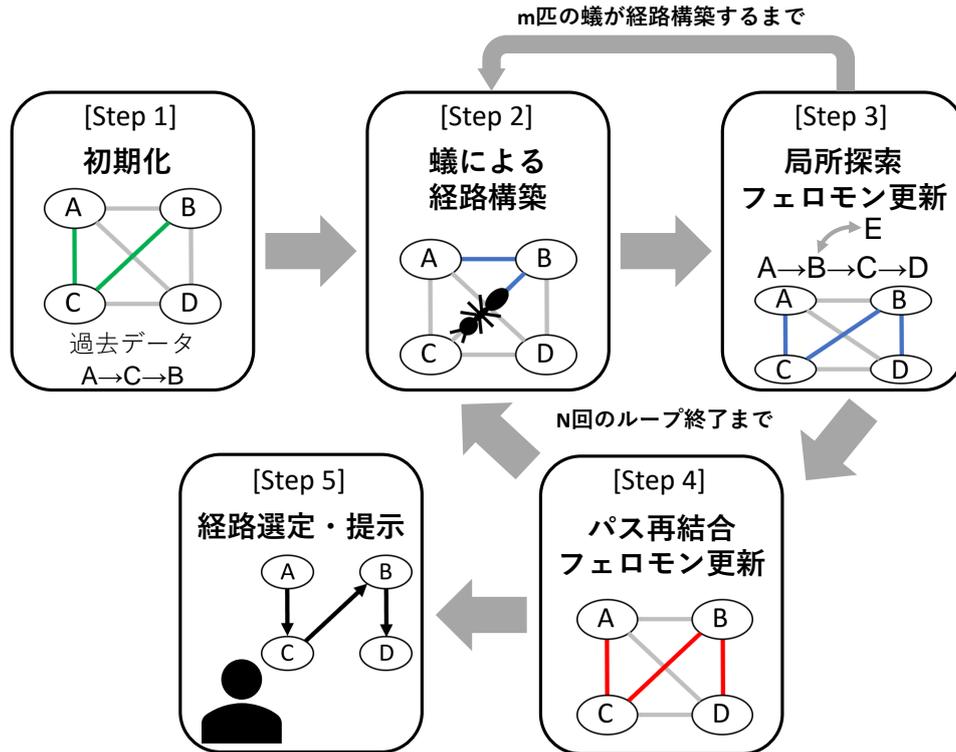


図 3: アルゴリズムの流れ.

実験のもと、提案手法では $q_0 = 0.4$ とした。なお、次訪問ノード v_j が決定した際、現在地 v_i から次訪問ノード v_j を経由し目的地 v_{dst} に至る経路の所要時間が制限時間 T を超える場合、現在地 v_i から直接、目的地 v_{dst} に到着し、経路探索を終了する。

上記 (1) および (2) において、 $p_k(i, j)$ は次のように計算される。グラフ $G = (V, E)$ のエッジ $(v_i, v_j) \in E$ において、 v_i から v_j に至る経路の現在のフェロモンを $\tau(i, j)$ とする。また、 $\eta(i, j) = 1/t_{ij}$ とする。移動時間が小さいほど、 $\eta(i, j)$ の値は大きくなる。さらにノード v_j に対して $\xi(j)$ を以下のように定義する：

$$\xi(j) = 1 - \frac{c_j}{\max_{u \in V} \{c_u\}} \quad (1)$$

この際、蟻 k の現在地 v_i に隣接する未訪問ノード v_j について、 $p_k(i, j)$ は以下のように算出する：

$$p_k(i, j) = \begin{cases} \frac{[\tau(i, j)]^\alpha \cdot [\eta(i, j)]^\beta \cdot [s_j]^\gamma \cdot [\xi(j)]^\mu}{\sum_{u \in J_k(i)} [\tau(i, u)]^\alpha \cdot [\eta(i, u)]^\beta \cdot [s_u]^\gamma \cdot [\xi(u)]^\mu} & \text{if } j \in J_k(i) \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

ここで、 s_j は観光地 v_j の満足度であり、 $J_k(i)$ は v_i に隣接し、蟻 k が未訪問のノード集合である。 $\alpha, \beta, \gamma, \mu$ はパラメータである。

提案手法では、パラメータ $\alpha, \beta, \gamma, \mu$ を適当に設定することで、定義 1 の目的関数 (a)~(d) ごとに状態遷移

ルールを設定する。蟻 k に対して、 $k \equiv 0 \pmod{5}$ のとき、 $\alpha = 0.5, \beta = 1.0, \gamma = 0, \mu = 1.0$ とする。これは目的関数 (a) に対応するものであり、合計費用の最小化を目指す。蟻 k に対して、 $k \equiv 1 \pmod{5}$ のとき、 $\alpha = 0.5, \beta = 1.0, \gamma = 1.0, \mu = 0$ とする。これは目的関数 (b) に対応するものであり、合計満足度の最大化を目指す。蟻 k に対して、 $k \equiv 2 \pmod{5}$ のとき、 $\alpha = 1.0, \beta = 0, \gamma = 0, \mu = 0$ とする。これは目的関数 (c) に対応するものであり、過去のユーザの旅行履歴との類似度の最大化を目指す。蟻 k に対して、 $k \equiv 3 \pmod{5}$ あるいは $k \equiv 4 \pmod{5}$ のとき、 $\alpha = 1.0, \beta = 1.0, \gamma = 1.0, \mu = 1.0$ とする。これは目的関数 (d) に対応するものであり、合計満足度の最大化、合計費用の最小化、過去のユーザの旅行履歴との類似度の最大化を同時に目指す。表 2 に状態遷移ルールのパラメータをまとめる*1。

$p_k(i, j)$ による経路選択の例を以下に示す。

*1 上記の状態遷移ルールの設定において、目的関数 (c) に対応するパラメータ設定では、式 (2) のうちフェロモンを表す項のみを反映したものとなっており、3.5.3 節で議論する類似度を直接考慮していない。これは、予備実験により、(i) 式 (2) において、類似度を陽に計算すると極めて多くの計算時間を要することが予想されること（予備実験によれば 1 回の旅行計画で 1 時間近くの計算時間を要する）、(ii) 類似度を陽に計算することなく、フェロモンのみを考慮した状態遷移ルールにより、類似度を計算した場合とほぼ同等の類似度を持つ旅行経路を生成できること、の 2 点を確認したためである。また、目的関数 (d) に対応するパラメータ設定を持つ蟻の数が、目的関数 (a)~(c) に対応するパラメータ設定を持つ蟻の数の 2 倍となっている。これは、満足度、費用、類似度のそれぞれに特化せず、3 つの価値をバランスよく考慮した解を多く探索するためである。

表 2: 状態遷移ルールのパラメータ.

| $i: k \equiv i \pmod{5}$ | 目的関数 | α | β | γ | μ |
|--------------------------|------|----------|---------|----------|-------|
| 0 | (a) | 0.5 | 1.0 | 0 | 1.0 |
| 1 | (b) | 0.5 | 1.0 | 1.0 | 0 |
| 2 | (c) | 1.0 | 0 | 0 | 0 |
| 3, 4 | (d) | 1.0 | 1.0 | 1.0 | 1.0 |

表 3: 観光地の満足度と費用, $p_k(i, j)$ の値.

| 観光地 | 満足度 | 費用 | (a) の $p_k(i, j)$ | (b) の $p_k(i, j)$ |
|-------|-----|-----|-------------------|-------------------|
| v_a | 1.0 | 1.0 | 29.3 | 36.6 |
| v_b | 5.0 | 1.0 | 56.4 | 73.8 |
| v_c | 3.0 | 0.5 | 59.0 | 70.5 |
| v_d | 3.0 | 1.0 | 35.9 | 44.9 |
| v_e | 3.0 | 1.0 | 32.2 | 40.2 |
| v_f | 4.0 | 1.5 | 38.0 | 47.5 |
| v_g | 2.0 | 0.5 | 24.7 | 30.9 |

例 2 図 4 のように観光地として $v_a, v_b, v_c, v_d, v_e, v_f, v_g$ が与えられ, 各観光地の満足度と費用が表 3 のように与えられたとする. 今, 蟻 k が出発地 v_{st} に滞在しているとす. 蟻 k が目的関数 (a) に対応して行動する場合, 表 2 の通り, $\alpha = 0.5, \beta = 1.0, \gamma = 0, \mu = 1.0$ のパラメータをとり, 各観光地の $p_k(i, j)$ の値は表 3 の 4 カラム目の値をとる. このとき, v_{st} の次に $p_k(i, j)$ の値が高い観光地 v_c を訪問する可能性が高い. 同様に, 蟻 k が目的関数 (b) に対応して行動する場合, 表 2 の通り, $\alpha = 0.5, \beta = 1.0, \gamma = 1.0, \mu = 0$ のパラメータをとり, 各観光地の $p_k(i, j)$ の値は表 3 の 5 カラム目の値をとる. このとき, v_{st} の次に $p_k(i, j)$ の値が高い観光地 v_b を訪問する可能性が高い. 目的関数 (c) や目的関数 (d) の場合も同様である. 実際の実行結果として, 目的関数 (a), (b) に対応する蟻は図 4 に示すように, $p_k(i, j)$ の値が高い観光地を選択した. ■

3.5 局所探索・フェロモン更新 (Step 3)

Step 3 では, 局所探索と 2 つのフェロモン更新 (ローカル更新と類似度更新) を実行する.

3.5.1 局所探索

局所探索では, 蟻 k によって構築された経路について, 観光地ノードの交換を行うことで, 経路の改良を図る. 今, Step 2 で蟻 k が目的関数 (a) によって経路を構築したとする.

経路に含まれている観光地ノードの 1 つを v_i とする. 経路に含まれていない観光地ノードの集合からランダムに 1 つの観光地ノード v_j を選択し, v_i と v_j とを交換する. 交換の結果, 制限時間 T を超えずに総満足度が高くなれば交換を採用する. 高くならなければ交換を不採用とする. 上記の交換の手順を, v_i に対し経路に含まれていないすべての観光地ノードについて実行する. さらに, 経路に含まれているすべての観光地ノードについて上記手順を繰り返し

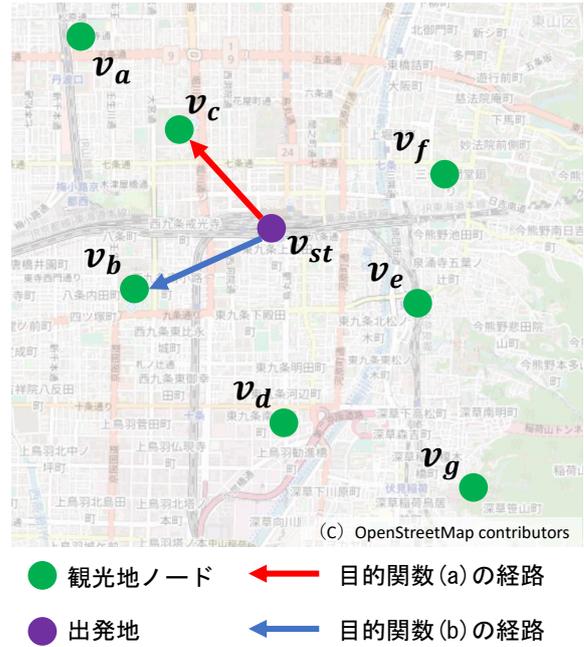


図 4: $p_k(i, j)$ による経路選択の例.

実行する.

目的関数 (b)~(c) に対応する蟻についても, 同様な手順で経路の改良を図る. 目的関数 (d) については, 蟻が構築した経路を保存するために局所探索は適用しない.

3.5.2 フェロモン更新 1: ローカル更新

Step 2 の経路構築ならびに Step 3 の局所探索によって, 蟻 k が出発地 v_{st} から目的地 v_{dst} に至る経路を生成した. その後, 蟻 k が構築した経路のフェロモン量を調整する. このフェロモン調整をローカル更新と呼ぶ. 構築された経路中の隣接する 2 つのノード v_i と v_j について, エッジ (v_i, v_j) のフェロモン $\tau(i, j)$ を以下のように更新する:

$$\tau(i, j) \leftarrow (1 - \rho_l) \cdot \tau(i, j) + \rho_l \Delta \tau(i, j) \quad (3)$$

ここで, ρ_l はパラメータで, フェロモンが減衰する量を調整する. 提案手法では, $\rho_l = 0.01$ とした. $\Delta \tau(i, j)$ は, 蟻 k によってエッジ (v_i, v_j) に追加されるフェロモン量を表す定数で, [14] における設定を参考に, 提案手法では $1/(nT)$ とした. ただし, n は観光地ノードの数である.

ローカル更新は, 蟻 k が構築した経路のフェロモン量を減衰させることで, Step 2 および Step 3 を繰り返す際に, 蟻 k が構築した経路を次以降に経路を構築する蟻が通りにくくし, 多様な経路の構築を促す.

3.5.3 フェロモン更新 2: 類似度更新

前節のローカル更新に加え, さらに, 蟻 k によって構築された経路と過去のユーザの旅行履歴との類似度に基づき, 構築された経路のフェロモンを更新する. この手順により, 蟻 $k+1$ あるいはそれ以降の蟻が, 過去のユーザの旅行履歴を考慮した経路を構築しやすくなる.

類似度の導入

今、蟻 k が構築した経路を $Tour^k$ とする。 T_k を構築した経路の所要時間とする。その上で、 $Tour^k$ 中に含まれるノードを文字列とみなし、 str_k とする。例えば $Tour^k = [a, b, c]$ であれば、 $str_k = "abc"$ とする。 str_k の長さを l_k とする。同様に、過去のユーザの旅行履歴 $Tour^{past}$ に対応する文字列を str_o 、その長さを l_o とする。 $lsd(str_k, str_o)$ を2つの文字列のレーベンシュタイン距離 [15]、 $lcs(str_k, str_o)$ を2つの文字列の最長共通部分列の長さとする。このとき、2つの経路の類似度 $Sim(str_k, str_o)$ を以下のように定義する:

$$Sim(str_k, str_o) = \left[\rho_o \cdot \left(1 - \frac{lsd(str_k, str_o)}{\max(l_k, l_o)} \right) + (1 - \rho_o) \cdot \frac{lcs(str_k, str_o)}{\min(l_k, l_o)} \right] \cdot \frac{T_k}{T} \quad (4)$$

ただし、 $\rho_o = l_k / (l_k + l_o)$ とする。 ρ_o は構築した経路と過去の旅行データの長さの差にかかわらずに、レーベンシュタイン距離と最長共通部分列の双方にバランスよく重みを付加するためのパラメータである。 ρ_o によって、過去の旅行データの長さが極端に短い場合にレーベンシュタイン距離の影響で $Sim(str_k, str_o)$ が極端に小さくなることを防ぐ。

式 (4) の通り、提案手法が採用する経路の類似度は、レーベンシュタイン距離と最長共通部分列による。まず最長共通部分列を取り入れることで、 $Tour^k$ 中の観光地が $Tour^{past}$ の観光地を含む場合、類似度が大きくなる。さらにレーベンシュタイン距離を取り入れることで、 $Tour^{past}$ に含まれる観光地の順序と $Tour^{past}$ の観光地の順序が一致する場合、類似度が大きくなる。

例 3 $Tour^k = [a, b, c, d, e]$ 、 $Tour^{past} = [b, c, e]$ の場合を考える。簡略化のため、 $T_k = T$ とする。このとき、 $lsd(str_k, str_o) = 2$ 、 $lcs(str_k, str_o) = 3$ となる。結果として、 $Sim(str_k, str_o) = 0.75$ のように、類似度が計算される。式 (4) で表す類似度は、 $Tour^k$ と $Tour^{past}$ が一致するとき、1.00 となる。また、 $Tour^k$ と $Tour^{past}$ で1つも一致する観光地ノードがないとき、0.00 となる。 ■

類似度更新

構築した経路 $Tour^k$ とすべての過去の旅行履歴に対して式 (4) を適用し平均をとった値を Sim_k と書き、これを $Tour^k$ の類似度と呼ぶ。そして、前節のローカル更新に加え、 $Tour^k$ を構成する各エッジに対して、 $Sim_k \cdot \rho_{old}$ の量のフェロモンを付加する。提案手法では、 $\rho_{old} = 3$ とした。

3.5.4 パレート解の更新

最後に、蟻 k によって構築された経路 $Tour^k$ とパレート解集合 $\{Tour^i\}$ の各解を比較し、パレート解を更新する。 $Tour^k$ の合計満足度を S_k 、合計費用を C_k とする。同様に、1つのパレート解 $Tour^i$ の合計満足度を S_i 、合計費

用を C_i とする。このとき、 $S_k > S_i$ 、 $C_k < C_i$ のいずれかを満足するパレート解 $Tour^i$ がパレート解集合に存在するとき、 $Tour^k$ をパレート解集合に含めるものとする。その後、パレート解集合の任意の2つの解 $Tour^i$ 、 $Tour^j$ において、 $S_i \geq S_j$ 、 $C_i \leq C_j$ の両方が成立する場合、 $Tour^j$ をパレート解集合から削除する。

3.6 パス再結合・フェロモン更新 (Step 4)

Step 4 では、パス再結合とグローバル更新と呼ばれるフェロモン更新を実行する。

3.6.1 パス再結合

Step 3・Step 4 を繰り返し実行することで複数のパレート解を得る。これらのパレート解に対して、パス再結合 [16] により複数の経路を同時に参照することで、複数の目的関数をバランスよく最適化することを目指す。パス再構築は以下の4ステップより構成される:

(パス再結合ステップ1) $Tour^{init}$ と $Tour^{guide}$ の選択

パレート解を構成する各解 $Tour^i$ について以下を繰り返す。

(パス再結合ステップ2) 挿入と削除

(2-1) $Tour^{init} \leftarrow Tour^i$ とする。 $Tour^{init}$ の所要時間を T_{init} とし、パレート解の集合の中からランダムにガイド解 $Tour^{guide}$ を選択する。

(2-2) (パス再結合ステップ3)~(パス再結合ステップ4) を $Tour^{guide}$ に含まれるが $Tour^{init}$ に含まれないすべての観光地ノードが $Tour^{init}$ に挿入されるまで繰り返す。

(パス再結合ステップ3) 観光地ノード挿入

制限時間 T の制約によって $Tour^{init}$ に1つ以上の観光地ノードが追加可能な場合、以下の処理を実行する。今、 $Tour^{guide}$ に含まれるが $Tour^{init}$ に含まれない観光地ノードの1つを v_w とし、 $Tour^{init}$ を構成するエッジ (v_i, v_j) について、 v_i と v_j の間に v_w を挿入する。挿入後の $Tour^{init}$ の所要時間を T'_{init} とする。 T'_{init} が制限時間 T を超えないとき、

$$\delta t(i, w, j) = T'_{init} - T_{init} \quad (5)$$

とする。 $Tour^{init}$ を構成するすべてのエッジについて、式 (5) を計算し、その最小値を δt_{min} とする。また δt_{min} を与える $Tour^{init}$ のエッジ (v_i, v_j) を最適な挿入位置と呼ぶ。このとき、 $Tour^{guide}$ に含まれるが $Tour^{init}$ に含まれない観光地ノード v_w すべてについて、以下の値 $b(w)$ を算出する:

$$b(w) = \frac{[\tau(i, w)]^\alpha \cdot [\eta(i, w)]^\beta \cdot [s_w]^\gamma \cdot [\xi(w)]^\mu}{\delta t_{min}} \quad (6)$$

式 (6) において、 α 、 β 、 γ 、 μ の値は、表2において

目的関数 (d) のものを採用する。 $b(w)$ の値が最大となる観光地ノード v_w を実際に $Tour^{init}$ の最適な挿入位置に挿入する。その後、3.5.4 節と同様にパレート解集合を更新する。

(パス再結合ステップ 4) 観光地ノード削除

一方、制限時間 T の制約によって $Tour^{init}$ に 1 つも追加の観光地ノードを挿入できない場合、 $Tour^{init}$ に含まれるが $Tour^{guide}$ に含まれない観光地ノード v_j を次のように削除する。観光地ノード v_j の削除コスト r_j を以下のように計算する:

$$r_j = \frac{[\tau(i, j)]^\alpha \cdot [\eta(i, j)]^\beta \cdot [s_j]^\gamma \cdot [\xi(j)]^\mu}{d_j} \quad (7)$$

式 (7) において、 α , β , γ , μ の値は、表 2 において目的関数 (d) のものを採用する。また式 (7) において、 v_i は、 $Tour^{init}$ において v_j の 1 つ前の観光地ノードである。 r_j の値が最小となる観光地ノード v_j を $Tour^{init}$ から削除する。

3.6.2 フェロモン更新 3: グローバル更新

パス再結合を実行後、パレート解集合のすべてパレート解に対して、フェロモン量を更新する。このフェロモン調整をグローバル更新と呼ぶ。更新にあたって、パレート解のうち総満足度上位 2 つの経路について、経路 $Tour^k$ の総満足度を S_k としたとき、

$$\Delta\tau = \rho_s \cdot S_k \quad (8)$$

とする。さらに、総費用最小の 2 つの経路について、経路 $Tour^k$ の合計費用を C_k としたとき、

$$\Delta\tau = \rho_c \cdot \frac{1}{C_k} \quad (9)$$

とする。上記 $\Delta\tau$ の値によって、各経路を構成するエッジのフェロモンを更新すれば、目的関数 (a) および (b) の値が高い解の近傍を蟻が探索しやすくなる。目的関数 (c) については、3.5.3 節で述べた類似度更新によってフェロモンが更新されるため、グローバル更新では特別な処理は適用しない。

また、残ったパレート解について、経路 $Tour^k$ の所要時間を T_k としたとき、

$$\Delta\tau = \rho_p \cdot \frac{S_k}{C_k \cdot T_k} \quad (10)$$

とする。式 (8)~式 (10) において、 ρ_s , ρ_c , ρ_p はそれぞれ付加するフェロモン量を調整するパラメータであり、 $\rho_s = 0.002$, $\rho_c = 0.0008$, $\rho_p = 0.02$ とした。

以上の設定のもと、パレート解の経路中の隣接する 2 つのノード v_i と v_j について、エッジ (v_i, v_j) のフェロモン $\tau(i, j)$ を以下のように更新する:

$$\tau(i, j) \leftarrow (1 - \rho_g) \cdot \tau(i, j) + \Delta\tau \quad (11)$$

ρ_g はパラメータで、フェロモンが減衰する量を調整する。提案手法では、 $\rho_g = 0.1$ とした。

表 4: データセットの詳細。

| 観光エリア | 観光地数 | 過去データ数 | 出発地 | 目的地 |
|-------|------|--------|-----|-----------|
| 京都 | 150 | 315 | 京都駅 | 蓮花の湯 |
| 札幌 | 100 | 146 | 札幌駅 | 札幌グランドホテル |

3.7 経路選定・提示 (Step 5)

経路の選定では目的関数に応じた経路を選定し、選定された経路が最終的にユーザに提示される。経路は N 回の反復終了時のパレート解の集合の中から選定される。目的関数 (a), (b) を対象に、パレート解の中で最も合計満足度が高い経路と、最も合計費用が低い経路が選定される。目的関数 (c) を対象に、すべてのパレート解 k に対して Sim_k を計算し、最も値の高い経路が選定される。

目的関数 (d) を対象とした経路選択では、3 つの目的関数のバランスの良い経路をユークリッド距離によって求める。すべてのパレート解において、合計満足度、合計費用、 Sim_k を最小値 0、最大値 1 で正規化した値を、それぞれ (\hat{s}) , (\hat{c}) , (\hat{Sim}_k) としたとき、

$$\sqrt{(1 - (\hat{s}))^2 + (\hat{c})^2 + (1 - (\hat{Sim}_k))^2} \quad (12)$$

の値が最も小さいものを目的関数 (d) の経路として選定する。

4. 評価実験

本章では、提案手法を実観光地図データに適用し、提案手法を評価する。

4.1 実験方法

4.1.1 データセット

実験で利用したデータセットを表 4 に示す。

各データセットにおいて、出発地を旅行者が最も利用する駅とし、目的地を旅行サイト [17] のランキングでその地域で一番人気のホテルとした。探索における制限時間 T は、1 日の旅行を想定して 10 時間とした。それぞれの観光地に対する満足度、費用、滞在時間と、過去のユーザの旅行履歴は [17] から取得した。

満足度は、0.1 刻みで最低が 1、最高が 5 の値である。一方、費用は観光地でかかる費用を 100 円単位で表した値であり、0 円から最大で 22000 円までの費用を含む。[17] に費用の記載がない飲食店は、グルメサイト [18] の費用を採用した。また、[18] にショッピングの平均費用の記載がない場合や、市場やショッピングセンターなど複数のショップが存在して費用設定が困難である場合は、[17] による旅行中の飲食にかかる平均費用を利用して一律で 2456 円に設定した。滞在時間は、旅行者の平均滞在時間を 1 時間単位で表した値である。また、過去のユーザの旅行履歴は、[17] に投稿されている、対象の観光エリアにおける実際の旅行者の 1 日の経路を表したデータである。例えば、京都の過

表 5: 提案手法のパラメータ.

| N | m | q_0 | I_{old} | ρ_l | ρ_{old} | ρ_g | ρ_p | ρ_s | ρ_e |
|-----|-----|-------|-----------|----------|--------------|----------|----------|----------|----------|
| 50 | 20 | 0.4 | 0.6 | 0.01 | 3 | 0.1 | 0.02 | 0.002 | 0.0008 |

表 6: 平均実行時間 (標準偏差).

| 手法 | 場所 | 実行時間 [s] |
|------------|----|-------------|
| 提案手法 | 京都 | 29.4 (0.80) |
| | 札幌 | 15.3 (0.18) |
| P-VNS [10] | 京都 | 28.9 (9.68) |
| | 札幌 | 15.2 (2.90) |

去のユーザの旅行履歴は、最大で 13 個、最小で 2 個、平均すると 4.96 個の観光地を訪問する経路が含まれ、過去のユーザの旅行履歴の総数は 315 個となる。

移動時間は各観光地の座標から観光地間の距離を算出し、日本の一般道路の平均旅行速度 [19] に近い時速 30km で走行した場合の移動時間を適用した。

4.1.2 比較手法

提案手法の有効性を評価するため、以下の 2 種類の手法を比較した。

提案手法： 表 5 に提案手法のパラメータをまとめた。蟻による探索の反復回数 N は 50 とし、1 回の反復当たりの蟻の数 m は 20 とした。提案手法は、目的関数 (a) ~ (d) に沿った 4 つの解を出力するが、途中解を含めて解全体の分布を評価するため、提案手法が内部に持つパレート解の集合を評価対象とした。

比較手法 P-VNS [10]： 比較手法として P-VNS [10] を採用した。我々の知る限り、1 章で議論した第一条件と第二条件を十分に満足する手法は存在しない。一方、P-VNS は複数の目的関数を持つ手法であり、主に 1 章で議論した第一条件を満足するものである。P-VNS は初期解に含まれるノードを別なノードに置き換えることで近傍の解を探索する局所探索法の一つである。多目的最適化に対応するため、解を評価する際、目的関数の重みを変化させることで、トレードオフの関係にある目的関数をバランスよく考慮し、複数のパレート解を保持する。また、解の更新時に常に現在の解よりも良い値の解を採用するのではなく、一定の確率で値の劣る解も採用することで局所解を回避する。P-VNS でも途中解を含めて解全体の分布を評価するため、P-VNS が内部に持つパレート解の集合を評価対象とした。

提案手法および P-VNS を Python3 で実装し、計算機 (CPU: Intel Xeon Gold 6130 2.10GHz, メモリ: 768GB) 上で 10 回実行し、結果を取得した。性能比較のために P-VNS は提案手法と同程度の実行時間になるように調整した。提案手法および P-VNS の平均実行時間を表 6 に示す。

4.2 経路探索結果

表 7 に京都のデータセットに対し提案手法と P-VNS を適用した結果を示す。表 8 に札幌のデータセットに対し適用した結果を示す。表中、「最大満足度」、「最小費用」、「最大類似度」は、それぞれ目的関数 (a), (b), (c) に対応して選定された経路の最大満足度、最小費用、最大類似度を表す。「満足度 (d)、費用 (d)、類似度 (d)」は目的関数 (d) に対して選定された経路の値を表す。「所要時間」は経路選定前のすべてのパレート解の平均所要時間を表し、「解の個数」はパレート解の個数を表す。図 5 と図 6 に、京都のデータセットと札幌のデータセットに対する提案手法と P-VNS の解の分布を示す。図 5, 6 では、10 回実験した際のすべてのパレート解の分布を表示している。

- 目的関数 (a), (b) の経路：

目的関数 (a) のもとに最適化された経路の「最大満足度」の平均値に注目すると、京都および札幌の両方のデータセットで提案手法が P-VNS より高い満足度の経路を構築できている。提案手法の「最大満足度」の標準偏差は、P-VNS の「最大満足度」の標準偏差の 2% 程度 (京都) あるいは 28% 程度 (札幌) であり、ばらつきが少なく安定した目的関数 (a) の経路を構築できている。

次に、目的関数 (b) のもとに最適化された経路の「最小費用」の平均値に注目すると、データセットにかかわらずどちらの手法でも 0 円となっている。費用が 0 円の経路を分析すると、京都では伏見稲荷大社や京都御所など拝観料 0 円の有名寺社が多く含まれており、札幌では大通公園や北海道日本庁舎などの料金が 0 円の有名観光地が多く含まれているからであると判明した。提案手法は第一条件について、P-VNS よりも良い経路を構築できている。

- 目的関数 (c) の経路：

次に目的関数 (c) に注目する。提案手法の「最大類似度」の平均値は、P-VNS の「最大類似度」の平均値に比べて、14% (京都) ならびに 43% (札幌) 高い。提案手法は第二条件について、P-VNS よりも良い経路を構築できている。

- 目的関数 (d) の経路：

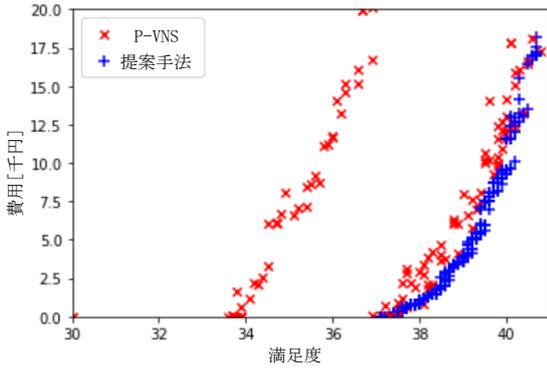
提案手法および P-VNS について、目的関数 (d) のもとに最適化された経路の「満足度 (d)、費用 (d)、類似度 (d)」の平均値を比較する。京都では、提案手法の満足度が 7% 高く、類似度が 4% 高いが、費用が 239% 高い経路を選定している。一方、札幌では、提案手法の満足度が 9% 高く、費用が 56% 低く、類似度が 40% 高い経路を選定している。目的関数 (d) はすべての目的関数を考慮する経路の構築が目的であり、トレードオフの関係の目的関数があるため、他の目的関数を対象とした経路と比較して評価が困難である。そこで、標

表 7: 京都のデータセットに対する平均値 (標準偏差).

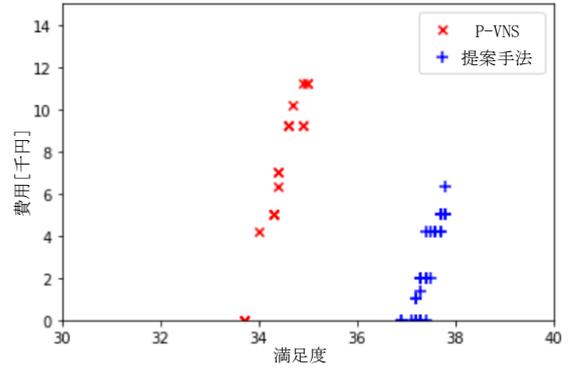
| 手法 | 最大満足度 | 最小費用 | 最大類似度 | 満足度 (d) | 費用 (d) | 類似度 (d) | 所要時間 | 解の個数 |
|-------|---------------|-------|---------------|---------------|-------------|---------------|--------------|--------------|
| 提案手法 | 40.09 (0.030) | 0 (0) | 0.065 (0.006) | 39.08 (0.140) | 4.24 (0.60) | 0.054 (0.007) | 9.85 (0.057) | 25.8 (2.482) |
| P-VNS | 39.65 (1.229) | 0 (0) | 0.057 (0.006) | 36.64 (1.772) | 1.25 (1.36) | 0.052 (0.007) | 9.76 (0.124) | 11.9 (4.763) |

表 8: 札幌のデータセットに対する平均値 (標準偏差).

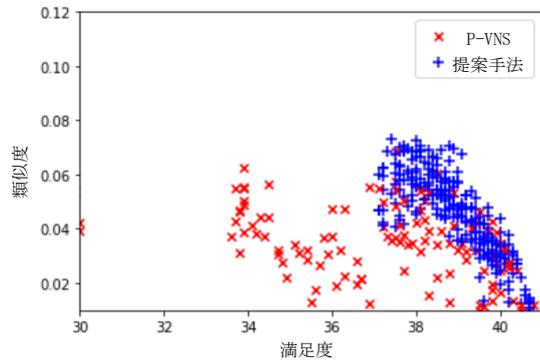
| 手法 | 最大満足度 | 最小費用 | 最大類似度 | 満足度 (d) | 費用 (d) | 類似度 (d) | 所要時間 | 解の個数 |
|-------|---------------|-------|---------------|---------------|-------------|---------------|--------------|-------------|
| 提案手法 | 37.75 (0.063) | 0 (0) | 0.110 (0.005) | 37.44 (0.133) | 2.20 (1.81) | 0.108 (0.008) | 9.90 (0.041) | 3.8 (0.943) |
| P-VNS | 34.67 (0.226) | 0 (0) | 0.077 (0.020) | 34.25 (0.350) | 4.98 (2.92) | 0.077 (0.019) | 9.68 (0.063) | 3.2 (0.831) |



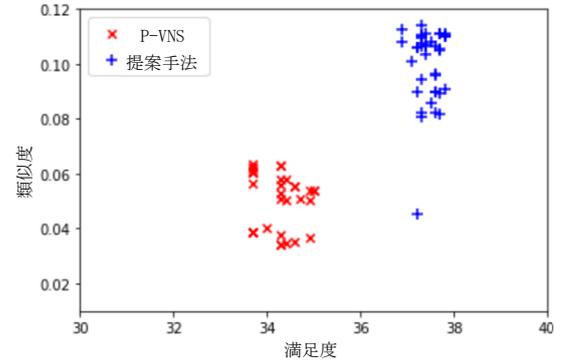
(a) 満足度と費用の分布.



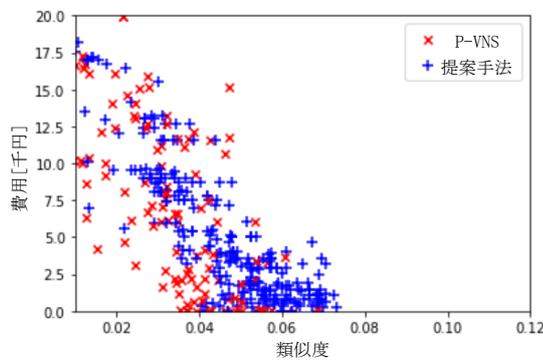
(a) 満足度と費用の分布.



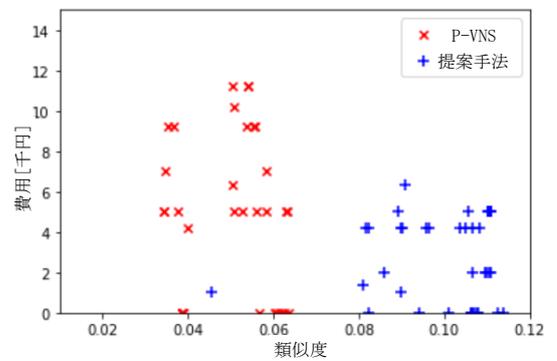
(b) 満足度と類似度の分布.



(b) 満足度と類似度の分布.



(c) 類似度と費用の分布.



(c) 類似度と費用の分布.

図 5: 提案手法と P-VNS の解の分布の比較 (京都).

図 6: 提案手法と P-VNS の解の分布の比較 (札幌).

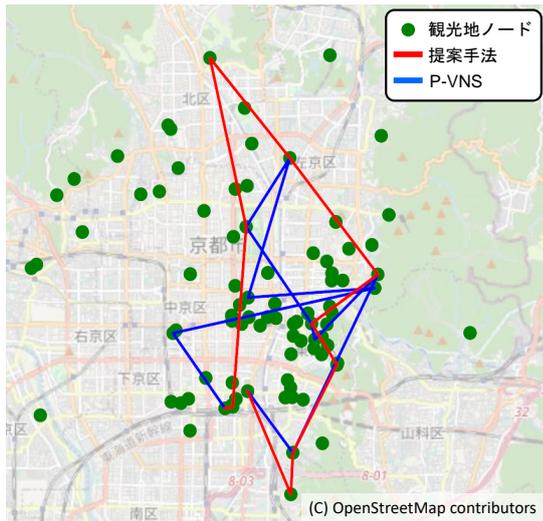


図 7: 目的関数 (d) の経路の例.

標準偏差や解の分布から、安定して全体的に良い解を選定できるかを評価する。表 7, 8 より、提案手法において目的関数 (d) のもとに最適化された経路の「満足度 (d), 費用 (d), 類似度 (d)」の標準偏差は、いずれも P-VNS で得られた標準偏差以下となっている。すなわち、提案手法の方が安定して、目的関数 (a)~目的関数 (c) を最適化できている。同時に、目的関数 (d) の経路はすべての目的関数を考慮して解を選定するため、標準偏差が小さいことは探索ごとのばらつきが小さいことも意味する。また、「解の個数」も提案手法の方が多くことから、目的関数 (d) の経路の選択肢が多く、よりバランスの良い経路を選定できる可能性が高い。さらに、図 5, 6 の解の分布より、提案手法の方が良い解を多く出力できている。

実際に実験によって取得された目的関数 (d) の経路の例を図 7 および表 9 に示す。

表 9 より、「満足度、費用、類似度」の値に注目すると、提案手法が満足度が 1% 高く、費用が 33% 低く、類似度が 60% 高い解を構築できている。よって、第一条件である多目的最適化を実現した解を構築できている。第二条件について、類似度の有効性と合わせて検証するために、過去のユーザの旅行履歴に多く出現した観光地および経路の部分列を表 10 に示す。

表 10 を基に、表 9 の提案手法および P-VNS の経路に含まれる過去のユーザの旅行履歴の数を表 11 に示す。表 11 より、提案手法は、過去のユーザの旅行履歴に出現する観光地および経路の部分列をより多く含むことが確認できる。実際に、提案手法の類似度は、P-VNS の類似度より 60% 高いことから、類似度の値に過去のユーザの旅行履歴を反映できている。したがって、第二条件の満足には満足度や費用のみでは不十分であり、類似度の計算によって、提案手法では第二条件を

満足し、ユーザの多様な要求に対応できる経路を構築している。

- その他の値：

京都と札幌のどちらのデータセットでも、提案手法の「所要時間」は、P-VNS の「所要時間」よりも制限時間に近い。制限時間はユーザが設定するため、所要時間が制限時間に近いほどユーザが望む時間で旅行できる。よって、提案手法の方がユーザの要求に近い経路を構築できている。提案手法の「解の個数」も P-VNS の「解の個数」よりも多く、提案手法がより多様な経路の中から目的関数に沿った経路を選定できている、ユーザの要求に合った経路を提示できる可能性が高い。

4.3 さまざまな評価指標による評価

多目的最適化問題は複数の解が出力される特徴があり、トレードオフの関係の目的関数もあるため、解の評価が単目的最適化より困難である。そのため、個々の解の評価だけでなく、解全体の収束度や多様性、分布の広がりなどを評価すべきである。そこで本節では、解の収束度を評価する指標である Generational Distance (GD)[20] と、解の収束度と多様性を評価する指標である Inverted Generational Distance (IGD)[21]、解の分布の広がり进行评估する Maximum Spread (MS)[22] を利用して評価実験の解を評価する（各指標の定義は付録 A を参照）。

表 12 に、提案手法と P-VNS の GD, IGD, MS の値を示す。本稿では既存のデータセットではなく現実の京都と札幌のデータセットを利用するため、GD, IGD で利用する最適解は未知である。そのため、本実験では提案手法と P-VNS を独立して 50 回実行し、得られた解をもとにパレート解の集合を算出した。そして、これらのパレート解集合を擬似最適解集合とした。なお、満足度、費用、類似度を等価に評価するために、最大値を 1、最小値を 0 として正規化している。

- GD の値：

表 12 から GD に注目すると、提案手法の GD の平均値は、P-VNS の GD の平均値に比較して、京都で 8%、札幌で 59% 程度と小さい。すなわち、提案手法の方がどちらのデータセットに対しても収束度が高く、良い解を構築できている。京都ならびに札幌のデータセットそれぞれに注目すると、提案手法、P-VNS のどちらも京都の方が収束度が高い。同様に、京都のデータセットに注目すると、提案手法の GD の標準偏差が他の値より 100 倍以上も低く、常に安定して収束度の高い解を構築できている。図 5 に注目すると、P-VNS の解では明らかに収束度が低い解が出力されているのに対し、提案手法はばらつきの少ない安定した解が出力されていることがわかる。さらに、表 7 および表 8 の解の個数に注目すると、提案手法、P-VNS のどちら

表 9: 目的関数 (d) の経路の例.

| 手法 | 満足度 | 費用 | 類似度 | 構築された経路 |
|-------|------|-----|-------|---|
| 提案手法 | 38.7 | 1.4 | 0.094 | 京都駅→京都駅ビル→京都御所→上賀茂神社→下鴨神社→永観堂→八坂神社→清水寺→東福寺→伏見稲荷大社→蓮花の湯 |
| P-VNS | 38.5 | 2.1 | 0.057 | 京都駅→壬生寺→南禅寺→京都府京都文化博物館→下鴨神社→京都御所→八坂神社→石堀小路→永観堂→東福寺→蓮花の湯 |

表 10: 過去のユーザの旅行履歴のデータ.

| 出現数の順位 | 観光地 | 経路の部分列 |
|--------|--------|------------------------|
| 1 | 清水寺 | 渡月橋→天龍寺→野宮神社 |
| 2 | 伏見稲荷大社 | 南禅寺→永観堂 |
| 3 | 鹿苑寺金閣 | 貴船神社→鞍馬寺 |
| 4 | 渡月橋 | 北野天満宮→鹿苑寺金閣→龍安寺→仁和寺 |
| 5 | 南禅寺 | 八坂神社→清水寺→伏見稲荷大社→平等院鳳凰堂 |
| 6 | 京都駅ビル | 西本願寺→東本願寺 |
| 7 | 八坂神社 | 野宮神社→竹林の道 |
| 8 | 天龍寺 | 慈照寺銀閣→哲学の道 |
| 9 | 東寺 | 上賀茂神社→下鴨神社 |
| 10 | 北野天満宮 | 宇治橋→東福寺 |

表 11: 経路に含まれる過去のユーザの旅行履歴の数.

| 手法 | 観光地 | 経路の部分列 |
|-------|-----|--------|
| 提案手法 | 4 | 2 |
| P-VNS | 1 | 0 |

も京都のデータセットに対する解の個数が多い。京都のデータセットと札幌のデータセットの大きな差異は観光地の個数であり、京都は札幌の 1.5 倍の数の観光地を含む。したがって、解の個数や分布の差は主に観光地の数に依存すると推測できる。

● IGD の値：

IGD に注目すると、提案手法の IGD の平均値は、P-VNS の IGD の平均値に比較して、京都で 4%、札幌で 30% 程度と小さい。標準偏差も提案手法の方が小さく、特に京都において、提案手法の IGD の標準偏差が大幅に小さい。よって、P-VNS より提案手法の方が多様性の高い解を構築できており、その傾向は京都のデータセットの方が強い。

● MS の値：

MS に注目すると、京都において、P-VNS の MS の平均値が提案手法の MS の平均値より高く、札幌では、提案手法の MS の平均値が P-VNS の MS の平均値より高い。一方、京都において、提案手法の標準偏差は、P-VNS の標準偏差に比較して、1/10 程度となっている。よって京都のデータセットに対して、P-VNS はまれに提案手法より MS の値が高い経路を構築する可能性があるが、安定して良い分布の解を得られていない。

札幌のデータセットに対する結果でも提案手法の標準偏差が低いため、提案手法は安定して MS の高い解を構築できている。よって、京都と札幌のデータセットについて、提案手法の方が安定して分布の広がった多様な解を構築できている。

以上より、提案手法は 1 章で述べた第一条件および第二条件を満足し、既存手法と比較して分布も優れた解を構築できている、と言える。

5. おわりに

本稿では、ACO による多目的要求に対応した旅行計画最適化手法を提案した。提案手法では ACO を利用することで、過去のユーザの旅行経路を陽に反映し、多目的最適化を実現した旅行計画を可能とした。評価実験により、既存手法に対し、過去の旅行者の旅行経路に近く、よりユーザの要求を満足する旅行経路を生成した。

今後の課題として、データセットによる値の偏りの解消や、必ず経路に含める観光地の設定、あるカテゴリの観光地は経路に含めないようにするなど、より細かなユーザの要求を反映できる制約の設定が考えられる。

謝辞

本研究は一部、JST CREST (Grant No. JPMJCR19K4) の支援を受けた。

参考文献

- [1] AVA Intelligence, AVA Travel, <https://travel.ava-intel.com/>.
- [2] まちづくりプラットフォーム, ルート最適化・AIによる観光プラン自動作成, <https://www.machi-pla.com/concierge>.
- [3] S. Bao, M. Tawada, S. Tanaka, and N. Togawa, "Multi-day travel planning using ising machines for real-world applications," in *Proceedings of 2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, pp. 3704–3709. IEEE, 2021.
- [4] 磯田祥吾, 日高真人, 松田裕貴, 諏訪博彦, 安本慶一 *et al.*, "観光ナビにおいて必見スポットをよりよい時間帯に訪問可能にするオンサイトプランニング手法," in *Multimedia, Distributed, Cooperative, and Mobile Symposium (DICOMO 2020)*, 2020.
- [5] 笠原秀一, 田村和範, 飯山将晃, 棕木雅之, 美濃導彦 *et al.*, "行動履歴に基づく地域の環境要因を考慮した観光行動モデルの構築とその応用," *情報処理学会論文誌*, vol. 57, no. 5, pp. 1411–1420, 2016.
- [6] X. Lu, C. Wang, J.-M. Yang, Y. Pang, and L. Zhang,

表 12: 評価指標の平均値 (標準偏差).

| 手法 | 場所 | GD | IGD | MS |
|-------|----|-----------------|-----------------|-----------------|
| 提案手法 | 京都 | 0.0167 (0.0008) | 0.0133 (0.0042) | 1.6733 (0.0347) |
| | 札幌 | 0.3632 (0.1155) | 0.3779 (0.1840) | 1.5359 (0.1467) |
| P-VNS | 京都 | 0.2083 (0.1620) | 0.3455 (0.3777) | 1.7401 (0.3202) |
| | 札幌 | 0.6147 (0.1016) | 1.2494 (0.2844) | 1.1300 (0.3539) |

“Photo2trip: generating travel routes from geo-tagged photos for trip planning,” in *Proceedings of the 18th ACM international conference on Multimedia*, pp. 143–152, 2010.

- [7] B. L. Golden, L. Levy, and R. Vohra, “The orienteering problem,” *Naval Research Logistics (NRL)*, vol. 34, no. 3, pp. 307–318, 1987.
- [8] W. Souffriau, P. Vansteenwegen, J. Vertommen, G. V. Berghe, and D. V. Oudheusden, “A personalized tourist trip design algorithm for mobile tourist guides,” *Applied Artificial Intelligence*, vol. 22, no. 10, pp. 964–985, 2008.
- [9] C. Voudouris and E. Tsang, “Guided local search and its application to the traveling salesman problem,” *European Journal of Operational Research*, vol. 113, no. 2, pp. 469–499, 1999.
- [10] M. Schilde, K. F. Doerner, R. F. Hartl, and G. Kiechle, “Metaheuristics for the bi-objective orienteering problem,” *Swarm Intelligence*, vol. 3, no. 3, pp. 179–201, 2009.
- [11] M. Dorigo, V. Maniezzo, and A. Coloni, “Ant system: optimization by a colony of cooperating agents,” *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 26, no. 1, pp. 29–41, 1996.
- [12] N. Mladenović and P. Hansen, “Variable neighborhood search,” *Computers & Operations Research*, vol. 24, no. 11, pp. 1097–1100, 1997.
- [13] R. Martin-Moreno and M. A. Vega-Rodriguez, “Multi-objective artificial bee colony algorithm applied to the bi-objective orienteering problem,” *Knowledge-Based Systems*, vol. 154, pp. 93–101, 2018.
- [14] M. Dorigo and L. M. Gambardella, “Ant colony system: a cooperative learning approach to the traveling salesman problem,” *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 53–66, 1997.
- [15] V. I. Levenshtein, “Binary codes capable of correcting deletions, insertions, and reversals,” in *Soviet Physics Doklady*, vol. 10, no. 8, pp. 707–710. Soviet Union, 1966.
- [16] R. Montemanni and L. M. Gambardella, “An ant colony system for team orienteering problems with time windows,” *Foundation of Computing and Decision Sciences*, vol. 34, no. 4, p. 287, 2009.
- [17] Recruit Co., じゃらん, <https://www.jalan.net/>.
- [18] Kakaku.com, Inc., 食べログ, <https://tabelog.com/>.
- [19] 国土交通省, 平成 27 年度 全国道路・街路交通情勢調査 一般交通量調査 集計表, <https://www.mlit.go.jp/road/census/h27/data/pdf/syuukei05.pdf>.
- [20] D. A. Van Veldhuizen and G. B. Lamont, “Evolutionary computation and convergence to a pareto front,” in *Late breaking papers at the Genetic Programming 1998 Conference*, pp. 221–228, 1998.
- [21] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. G. Da Fonseca, “Performance assessment of multi-objective optimizers: An analysis and review,” *IEEE*

Transactions on Evolutionary Computation, vol. 7, no. 2, pp. 117–132, 2003.

- [22] E. Zitzler, *Evolutionary algorithms for multiobjective optimization: Methods and applications*. Shaker Verlag GmbH, Germany, 1999.

付 録

A 評価指標の定義

• Generational Distance

GD はパレート解の集合と最適解の集合との距離を計算することで解の収束度を評価できる評価指標である [20]. 式 (A.1) により GD を計算する.

$$GD = \frac{\sqrt{\sum_{i=1}^n d(PF_i, OP)}}{n}$$

$$d(PF_i, OP) = \min_{o \in OP} \| PF_i - o \|^2 \quad (\text{A.1})$$

式 (A.1) において, n はパレート解の個数であり, PF_i は i 番目のパレート解, OP は最適解の集合を表す. よって, $d(PF_i, OP)$ は i 番目のパレート解と最も近い最適解とのユークリッド距離を表す. GD の値が小さいほどパレート解が最適解に収束しており, アルゴリズムによって求めた近似解の精度が高い.

• Inverted Generational Distance

IGD は GD 同様にパレート解と最適解を比較することで, 解の収束度と多様性を評価できる評価指標である [21]. 式 (A.2) により IGD を計算する.

$$IGD = \frac{\sqrt{\sum_{i=1}^n d(OP_i, PF)}}{n}$$

$$d(OP_i, PF) = \min_{p \in PF} \| OP_i - p \|^2 \quad (\text{A.2})$$

式 (A.2) において, n は最適解の個数であり, OP_i は i 番目の最適解, PF はパレート解の集合を表す. $d(OP_i, PF)$ は i 番目の最適解と最も近いパレート解とのユークリッド距離である. GD と異なりすべての最適解について距離を計算するため, パレート解の多様性を評価できる. ユークリッド距離を計算するため, GD と同様に, IGD は値が小さいほど多様で最適解に収束した解を解法できていることを示す.

- **Maximum Spread**

MS はパレート解の各目的関数における最大値と最小値を比較することで、解の分布の広がり进行评估できる評価指標である [22]. 式 (A.3) により MS を計算する.

$$MS = \sqrt{\sum_{i=1}^k \max \{ \| a_i - b_i \| \mid \mathbf{a}, \mathbf{b} \in PF \}} \quad (\text{A.3})$$

式 (A.3) において, k は目的関数の個数であり, a_i, b_i はパレート解 \mathbf{a}, \mathbf{b} の i 番目の目的関数の値である. それぞれの目的関数における最大値と最小値の距離を計算し, 分布の広がり进行评估する. MS の値が大きいほど分布が広がった多様な解を得られている.