モンテカルロ木探索を用いた ユーザ個人の嗜好を考慮した経路推薦手法の高速化

石崎 雄太[1,a)] 高山 敏典[2] 戸川 望[1,b)]

概要：スマートフォンやタブレットの普及が進み，経路推薦・案内サービスが幅広く利用されている．従来のインターネット上で提供されてきた経路推薦・案内サービスでは，経路長・所要時間・料金の点で最適化した経路を推薦するサービスが主流だが，これらのサービスでは各ユーザの区別がされない．このため現在地や目的地等の入力情報が同じであれば，各ユーザは安全性や快適性の面から多種多様な嗜好を有しているにもかかわらず，異なるユーザでも同一の経路推薦がされることになる．これまでに我々は，モンテカルロ木探索を用いた経路推薦手法 (P-UCT 手法) を提案し，同手法がユーザ個人の嗜好を考慮した経路を推薦することを，評価実験によって確認している．ところが，目的地点に到達するまでランダムなモンテカルロ・シミュレーションを繰り返して経路を探索するため，探索対象の経路長が大きくなると実行時間が指数的に増大してしまう問題がある．本稿では，モンテカルロ木探索を用いたユーザ個人の嗜好を考慮した経路推薦における高速化手法を提案する．提案手法では，目的地点周辺範囲 (AD) を新たに定義・導入することで，従来の P-UCT 手法のアルゴリズムの高速化を図る．

1. はじめに

1.1 ユーザ個人の嗜好を考慮した経路推薦

近年，スマートフォンやタブレットの普及が進み，経路推薦・案内サービスが幅広く利用されている．従来のインターネット上で提供されてきた経路推薦・案内サービスでは，経路長・所要時間・料金の点で最適化した経路を推薦するサービスが主流であった [1, 2]．これらのサービスでは各ユーザの区別がされないため，現在地や目的地等の入力情報が同じであれば，異なるユーザでも同一の経路推薦がされることになる．しかし，実際の道路には交通量・幅員・勾配・段差等の特徴があるため，各ユーザは安全性や快適性の面から多種多様な嗜好を有している．実際，松田らのアンケート調査 [3] によれば，ユーザには階段等の段差や勾配が大きい坂を避けるような経路，人通りが多く歩道もある経路，右左折回数が少なく道に迷いにくい経路を選択したい等のニーズがあるとされている．ユーザ個人の嗜好を経路推薦に反映させ，そのユーザ好みの経路を推薦することが強く望まれる．

[1] Waseda University
[2] ZENRIN DataCom
[a)] yuta.ishizaki@togawa.cs.waseda.ac.jp
[b)] togawa@togawa.cs.waseda.ac.jp

1.2 先行研究

個別のユーザに特化して当該ユーザの嗜好に合わせた経路推薦をする研究は，現在までにも存在しており，以下のように大別される．

(1) ダイクストラ法を利用した手法 [3]
松田らの研究 [3] では，ダイクストラ法を応用してユーザの嗜好を反映させる手法を提案している．同手法では，2 回のアンケート調査に基づき，歩行者ユーザの嗜好を定量化して経路コストの重み付けに反映する．その後，探索アルゴリズムを従来のダイクストラ法に帰着させている．

(2) 遺伝的アルゴリズムを利用した手法 [4–7]
遺伝的アルゴリズム (Genetic Algorithm; GA) を利用した手法では，経路情報を遺伝子としてコード化する．その後，独自の評価関数により選択された優位な個体について，交叉や突然変異を繰り返すことで経路を推薦する．GA を用いる利点として，独立した目的関数を設計することで性質の異なる経路の同時探索が可能になる，解の収束条件に世代数を設定することで制限時間内に準最適解を導出できることが挙げられる．例えば，原らの研究 [4] では，GA の一種である多目的遺伝的アルゴリズムが採用されている．同手法では，上記の利点に着目して，リアルタイム性が求められるカーナビゲーションシステムを対象とした経路推薦手

法を提案している．

(3) ファジィ測度・積分を利用した手法 [8, 9]

ファジィ測度・積分を利用した手法は，ファジィ測度の非加法性を利用して経路を評価している．経路を評価する要素間の相乗効果・相殺効果を考慮することで，ユーザ個人の嗜好をパラメータとして算出する．例えば，赤坂らの研究 [8] では，事前のアンケート調査により算出したファジィ測度・積分モデルにより，ユーザの嗜好が反映された経路を推薦する手法を提案している．

これらの先行研究はいずれも，アンケート調査に基づき，経路を構成する要素 (経路長，右左折数，勾配等) を個別に重み付けすることで，ユーザに特化した経路推薦を実現している．しかし，ユーザ個人の嗜好を個別の経路要素に反映させたとしても，経路全体がそのユーザの嗜好に適合しない場合がある．例えば，あるユーザがランニングのための「階段と坂道の各距離の合計が同程度になる経路」を希望する場合，経路生成途中で両者の数を比較・評価しながら経路探索することは困難であるが，完成した経路に対しては，両者の数を比較して評価できる．つまり，ユーザ個人の嗜好に適合する経路を推薦するためには，経路要素を個別に評価するのではなく，経路全体を評価しユーザに提示する機構が必要不可欠となる．

この課題の解決のため，モンテカルロ木探索アルゴリズム [10–12] に注目する．モンテカルロ木探索では，モンテカルロ・シミュレーション (以下，簡単のためシミュレーションと呼ぶ) を実行して準最適解を求めていくため，経路全体の評価が可能となる．モンテカルロ木探索では，アルゴリズム効率化のため，UCB1 値と呼ばれる探索木中のノードの評価指標を使用している．UCB1 値の計算には，シミュレーションの結果に対して適当な報酬を設定する必要があるため，従来の UCB1 値をそのまま経路推薦に適用することはできない．いかに，UCB1 値に相当するノード評価指標を設計して，効率化を図るかが最大の問題となる．

1.3 本稿の提案

これまでに我々は，ノード評価指標として P-UCB1 値を導入したうえで，モンテカルロ木探索を用いた経路推薦手法 (P-UCT 手法) [13] を提案し，同手法がユーザ個人の嗜好を考慮した経路を推薦することを，評価実験によって確認している．ところが，モンテカルロ木探索では目的地点に到達するまでランダムなシミュレーションを繰り返して経路を探索するため，探索対象の経路長が大きくなると実行時間が指数的に増大してしまう問題がある．

そこで本稿では，モンテカルロ木探索を用いたユーザ個人の嗜好を考慮した経路推薦における高速化手法を提案する．提案手法では，従来の P-UCT 手法に目的地周辺範囲 (Area around the Destination; AD) を新たに導入し，アルゴリズムの高速化を図る．AD とは適当に設定された目的地周辺の範囲である．シミュレーションの成功条件を「目的地点への到達」から「AD への到達」に変更することで，シミュレーションが失敗する確率を抑え，アルゴリズム全体の実行時間減少を図る．

表 1: 有名ランドマークと無名ランドマーク．

有名ランドマーク	官公庁施設，官公庁建物，教育施設，学校，病院，レジャー，交通，宿泊建物，商業建物，目標建物，一般建物，駅舎，駅建物
無名ランドマーク	上記以外

1.4 本稿の貢献

本稿の貢献は以下の通りである．

(1) 従来の P-UCT 手法の高速化実現のため，目的地周辺範囲 (AD) を定義した．AD とは適当に設定された目的地周辺の範囲である．シミュレーションの成功条件を「目的地点への到達」から「AD への到達」に変更することで，シミュレーションが失敗する確率を抑え，アルゴリズム全体の実行時間減少を図る．

(2) AD を P-UCT 手法に適用するため，AD の更新式を設定した．シミュレーションの進行度により，AD は逐次更新される．

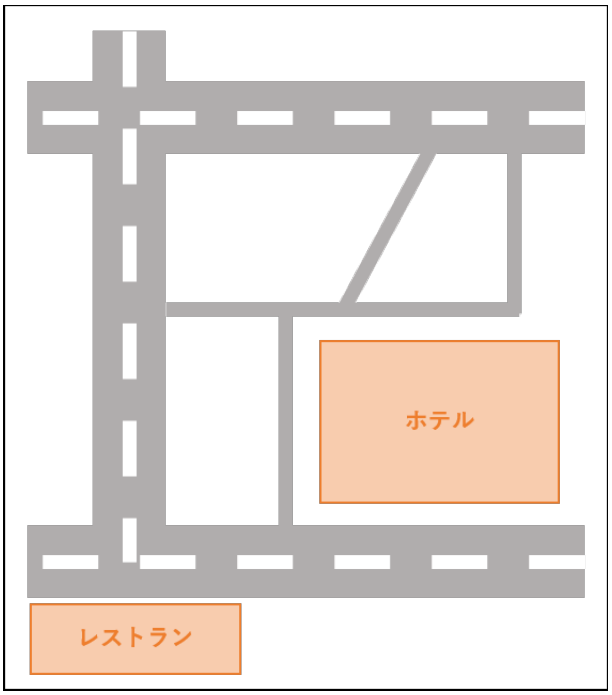
(3) 提案手法の有効性評価のため，AD 更新式のパラメータを変更しながら経路推薦実験を実施した．その結果，適切に AD 更新式中のパラメータを設定することで，提案手法が従来の P-UCT 手法の推薦精度を損なわずに，最大で約 34%程度の実行時間で経路を推薦することを確認した．

2. ユーザ個人の嗜好を考慮した経路推薦問題

2.1 道路ネットワーク

道路ネットワークは，グラフ $G = (V, E)$ で表される．道路ネットワークにおいて，V は交差点，曲がり角，横断歩道や歩道橋の端点の集合，E は道路の集合である．道路ネットワークには，付加的にランドマークの集合 $L = \{l_1, l_2, \ldots, l_k\}$ が与えられている．以降では，$v_s \in V$ を交差点ノード，$e_u \in E$ を道路エッジと呼称する．

交差点ノード $v_s \in V$ は，緯度 lat，経度 lng，高度 z，可視ランドマークの集合 $L_{visible} \subseteq L$ のパラメータを持つ．可視ランドマークとは，当該ノードにおいて目視可能なランドマークを指す．各ランドマーク l_t は，カテゴリ ct，高さ h，座標集合 PP のパラメータを持つ．カテゴリ ct は，当該ランドマークが有名ランドマークか無名ランドマークを表す．有名ランドマークと無名ランドマークを表 1 にまとめる．座標集合 PP は，ランドマークを多角形と見たとき，その多角形の頂点集合を表す．



(a) 地図表現.

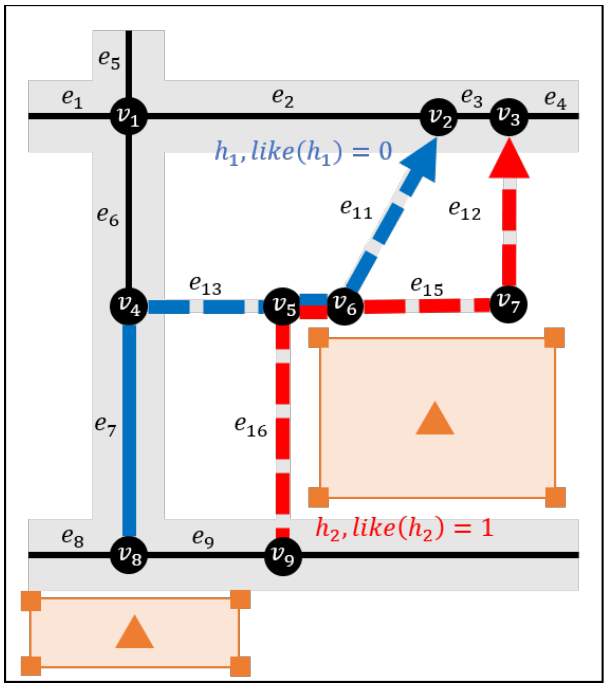
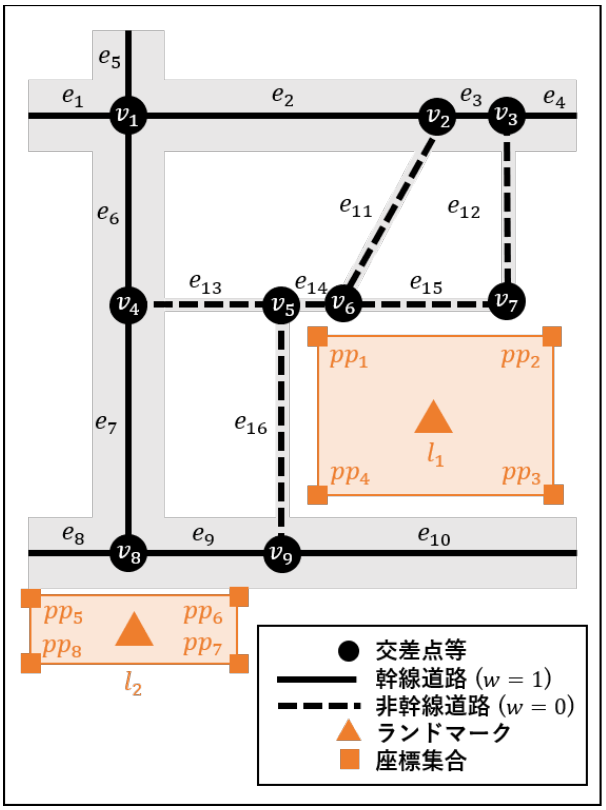


図 2: 経路推薦履歴の例.



(b) 道路ネットワーク表現.

図 1: 道路ネットワークの例.

道路エッジ $e_u \in E$ は，道路種別 et，勾配種別 sl，横断有無 cr，屋根の有無 hr，幹線道路の是非 w のパラメータを持つ[*1]．道路種別 et は，通常道路，歩道，横断歩道，踏切内通路，歩道橋等の道路のタイプを表す．勾配種別 sl は，坂，階段，段差，平坦等の道路の傾斜や高低差を表す．横断有無 cr は，当該エッジが横断歩道等の道路を横断する経路であるかどうかを表す．

例 1. 図 1 に道路ネットワークの例を示す．図 1(a) の地点には，複数の幹線道路と非幹線道路が交差し，ホテルとレストランのランドマークが存在している．このような地点を道路ネットワークで表現すると，図 1(b) のように，9 個の交差点ノード $\{v_1, v_2, \ldots, v_9\}$，10 本の幹線道路エッジ $\{e_1, e_2, \ldots, e_{10}\}$，6 本の非幹線道路エッジ $\{e_{11}, e_{12}, \ldots, e_{16}\}$，2 個のランドマーク $\{l_1, l_2\}$ から構成される．幹線道路の是非は，道路エッジのパラメータ w によって判断される．各ランドマークにはそれぞれ座標集合が紐付き，l_1 には $\{pp_1, pp_2, pp_3, pp_4\}$，$l_2$ には $\{pp_5, pp_6, pp_7, pp_8\}$ が対応している． ■

2.2 経路推薦履歴

経路推薦履歴とは，これまでにユーザ u が推薦されてきた経路集合 $H = \{h_1, h_2, \ldots, h_n\}$ と，各経路 $h_i \in H$ がユーザ u の嗜好に適合したかどうかを表すパラメータ $like(h_i)$ から構成される．経路推薦履歴の経路集合において，出発地点や目的地点は様々であり，パラメータ $like(h_i)$ は，好き/嫌いの 2 段階評価である．

図 2 に経路推薦履歴の例を示す．図 2 では，既に h_1 (青線) と h_2 (赤) の 2 経路が経路推薦されている．経路 h_1 はユーザ u の嗜好に適合せず，経路 h_2 はユーザ u の嗜好に適合したことを表している．

[*1] これらのパラメータはゼンリン [14] の道路ネットワークによる．

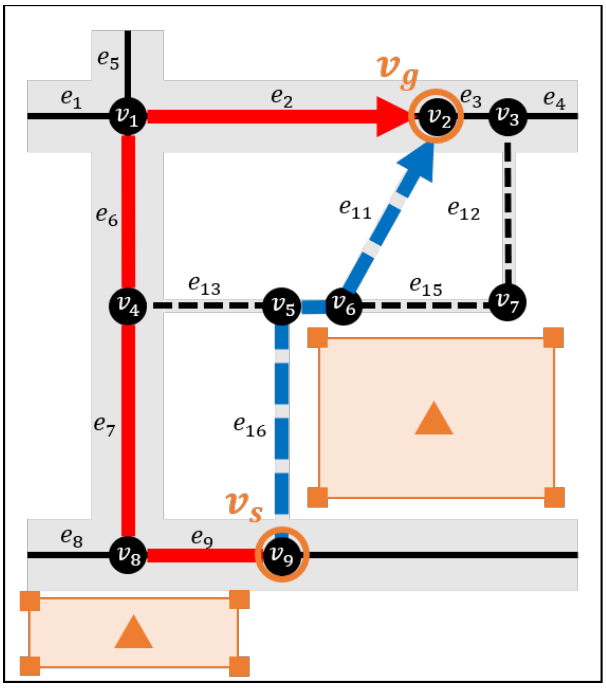


図 3: 最適経路推薦問題の例.

2.3 最適経路推薦問題

本稿で取り扱う最適経路推薦問題を以下に定義する.

定義 1 (最適経路推薦問題). 道路ネットワーク $G = (V, E)$ とユーザの経路推薦履歴が与えられたとき，最適経路推薦問題とは，始点 $v_s \in V$ から終点 $v_g \in V$ に至る道路ネットワーク上の経路を出力するものである．但し，ユーザの経路推薦履歴に基づき，出力される経路はユーザ個人の嗜好に沿ったものとする.

例 2. 例えば，図 3 のような道路ネットワークが与えられ，あるユーザが経路推薦履歴のもと「ある程度の遠回りは許容できるので，大通りを通る方が好ましい」という嗜好を有していたとする．このとき，同ユーザが地点 v_s を出発地点，地点 v_g を目的地点として入力した場合，図中の赤線のような経路を出力する．なお，図中の赤線で示される経路は，青線で示される最短経路よりも距離は長いが，非幹線道路を通行しない経路となり，ユーザの嗜好に適合した経路となっている. ■

3. ユーザ個人の嗜好を考慮した経路推薦手法 (P-UCT 手法) [13]

文献 [13] で我々が提案した，モンテカルロ木探索によるユーザ個人の嗜好を考慮した経路推薦手法 (P-UCT 手法) の概要を以下に示す.

3.1 経路推薦アルゴリズム

経路推薦アルゴリズムは，以下の Phase 1–Phase 3 から構成される.

Phase (1)：評価器の生成

経路探索の実施前に，経路推薦履歴から SVM (Support Vector Machine) [15] を用いて 20 種類の特徴量 (経路長，右左折数，経路中の勾配やランドマークなど) を抽出して評価器を作成する.

Phase (2)：出発地点・目的地点の指定

ユーザが出発地点 $v_s \in V$，目的地点 $v_g \in V$ を指定し，v_s にポインタ $v_{now} \in V$ を置く．探索木中の全ノードのシミュレーション回数を 0 にする.

Phase (3)：経路の探索・推薦

本フェイズは以下の 5 種類のステップから構成される.

Step (3-1)：選択

道路ネットワーク上において，v_{now} と接続されている全てのノードの中から，最も P-UCB1 値が大きいノード v_{select} を一つ選択する.

Step (3-2)：シミュレーション

v_{select} からシミュレーションを実施して，目的地点 v_g までのランダムな経路 (プレイアウト経路) を作成する.

Step (3-3)：プレイアウト経路評価

プレイアウト経路から SVM を用いて 20 種類の特徴量を抽出する．抽出したプレイアウト経路の特徴量から，評価器がその経路がユーザにとって好みである確率 $0 \leq rw_{select} \leq 1$ を算出する (rw_{select} は報酬と呼ばれる).

Step (3-4)：更新

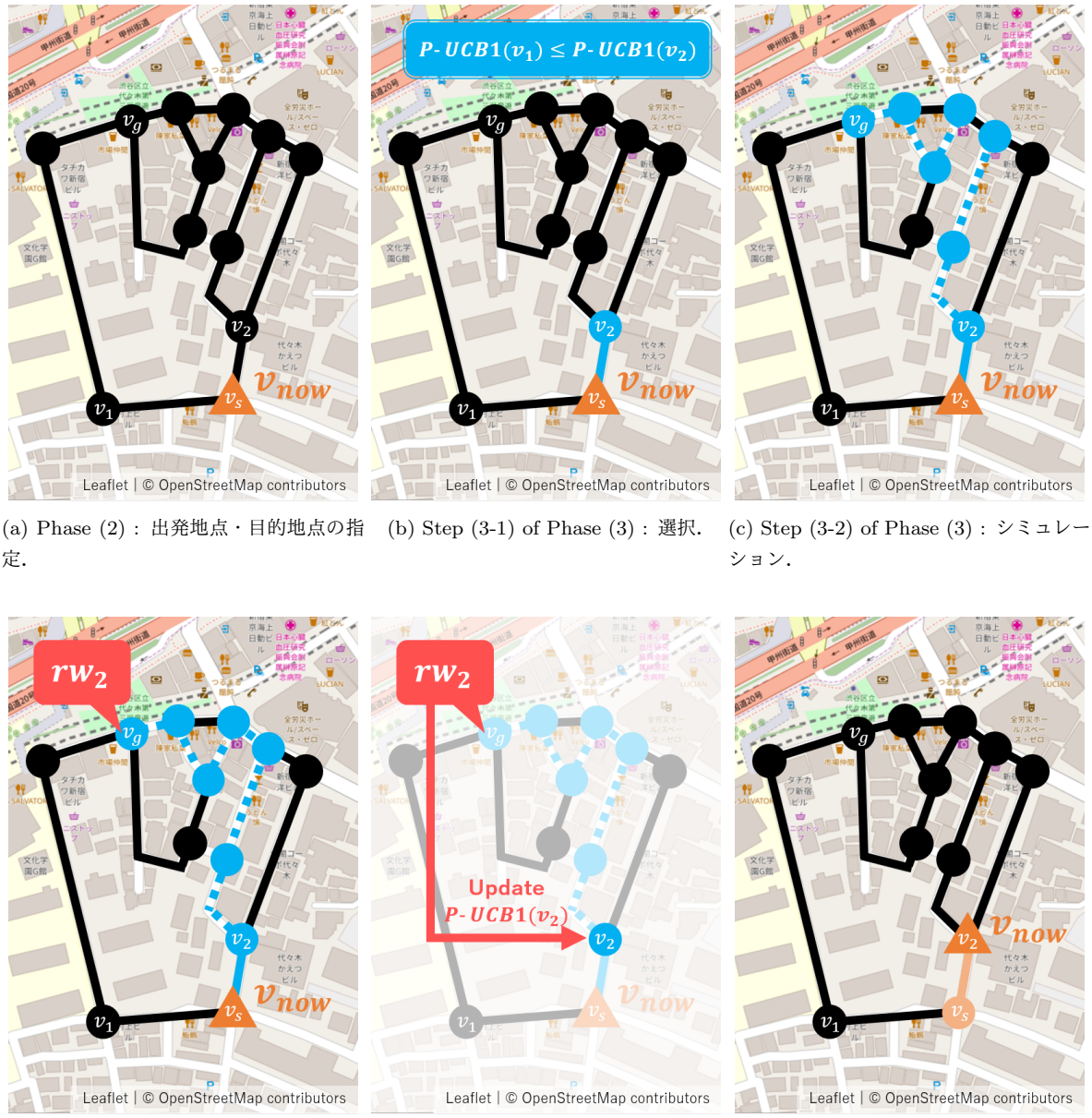
rw_{select} の値から v_{select} の P-UCB1 値を更新する．v_{select} のシミュレーション回数を 1 増加させる.

Step (3-5)：経路確定

Step (3-1) から Step (3-4) までを繰り返し処理する．Step (3-4) 終了時に v_{select} に対するシミュレーション回数が閾値 N[*2]に達している場合，Step (3-1) に戻る前に v_{select} を v_{now} とする．v_{now} が v_g に到達したとき，v_{now} が移動した経路を出力しアルゴリズムを終了する.

例 3. 図 4 に，経路推薦アルゴリズム実行例を示す．経路推薦履歴から評価器を作成する (Phase (1))．ユーザが出発地点と目的地点を指定したら，ポインタ v_{now} を出発地点に置く (Phase (2)，図 4(a))．v_{now} と接続されている 2 交差点ノードでの P-UCB1 値を比較し，値が大きい交差点ノードを選択する (Step (3-1) of Phase (3)，図 4(b))．選択した交差点からシミュレーションを実施して，目的地点までのプレイアウト経路を作成する (Step (3-2) of Phase

[*2] 通常，$N = 100$ に設定されている.



(a) Phase (2)：出発地点・目的地点の指定.　(b) Step (3-1) of Phase (3)：選択.　(c) Step (3-2) of Phase (3)：シミュレーション.

(d) Step (3-3) of Phase (3)：プレイアウト経路評価.　(e) Step (3-4) of Phase (3)：更新.　(f) Step (3-5) of Phase (3)：経路確定.

図 4: 経路推薦アルゴリズム実行例.

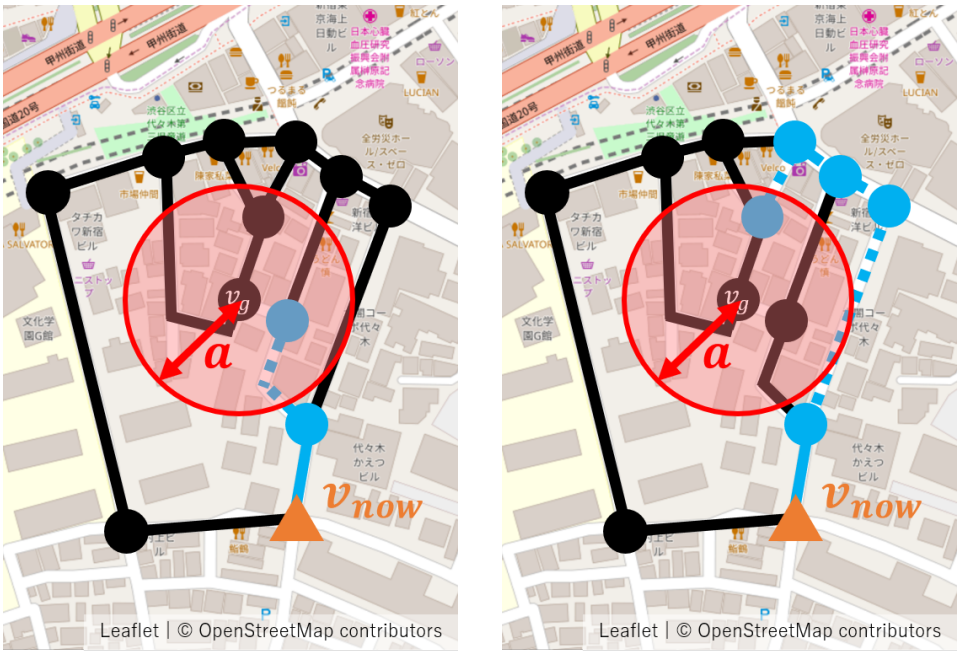
(3)，図 4(c))．プレイアウト経路から SVM を用いて 20 種類の特徴量を抽出し，抽出したプレイアウト経路の特徴量から，評価器がその経路がユーザにとって好みである確率を $0 \leq rw_j \leq 1$ 算出する (Step (3-3) of Phase (3)，図 4(d))．rw_j の値から選択した交差点ノードの P-UCB1 値を更新し，選択した交差点ノード v_1 のシミュレーション回数を 1 増加させ，交差点ノードの選択作業に戻る (Step (3-4) of Phase (3)，図 4(e))．以上の作業を繰り返すと，選択した交差点ノードでのシミュレーション回数が閾値 N に達する場合がある．この場合，出発地点から選択した交差点までの経路が確定して，経路探索が次段階へ進む (Step (3-5) of Phase (3)，図 4(f))．■

3.2 P-UCB1 値

P-UCB1 値とは，UCB1 値 [10] をベースにした各交差点ノード $v_j \in V$ の有望さを表す評価指標であり，3.1 節で提案した経路推薦アルゴリズムの挙動を制御する役割を持つ．初期値は探索木中の全ノードで 0 とする．P-UCB1 値を式 1 に示す．

$$\text{P-UCB1}(v_j) = \overline{X}_j + \sqrt{\frac{2\log n}{n_j}} \qquad (1)$$

$\overline{X}_j$ は当該交差点ノードを選択した際に返される報酬の期待値，n_j は当該交差点ノードを選択した回数，n は全交差点ノードの選択回数の総和である．但し，n や n_j は v_{now}



(a) シミュレーション成功例 (1). (b) シミュレーション成功例 (2).

図 5: AD を利用したシミュレーションの例.

が更新されるごとに 0 にリセットされる.

報酬の期待値 $\overline{X}_j$ は，式 2 によって更新される.

$$\overline{X}_j = \frac{\sum_{k=1}^{n_j} rw_j(k)}{n_j} \tag{2}$$

$rw_j(k)$ は，v_j にとって k 番目のプレイアウト経路に対する報酬である.

4. 目的地点周辺範囲を利用した高速化手法

従来の P-UCT 手法には，経路推薦アルゴリズム自体の実行時間が非常に長い問題が存在する．考えられる原因として，以下の 2 点が挙げられる.

(i) シミュレーションの必要回数が多い.

(ii) シミュレーションが失敗する確率が高い.

上記のシミュレーションとは，3.1 節に示した経路推薦アルゴリズムの「**Step (3-2) of Phase (3)：シミュレーション**」における，目的地点までのプレイアウト経路をランダムに作成していくことである.

(i) に関して，シミュレーションの必要回数は「**Step (3-5) of Phase (3)：経路確定**」における閾値 N に依存する．この N の値を小さくすることでシミュレーションの必要回数は減少するが，推薦精度 (推薦された経路の評価) も悪化する．ユーザ個人の嗜好をに沿った経路が推薦できなくなるため，閾値 N を変更する手法は望ましくない.

(ii) に関して，従来の P-UCT 手法におけるシミュレーションは，目的地点の交差点ノードまで到達しなければ成功と見なされない．目的地点に到達しないまま一定距離を超過すると，当該シミュレーションは失敗と見なされ，シミュレーションはやり直しとなる．失敗と見なされたシミュレーションは回数にカウントされないため，シミュレーションの失敗確率が高い場合，経路確定までに相当の時間を要する．基点となる交差点ノード v_{now} と目的地点 v_g 間の経路長が増大すると，シミュレーションの失敗確率も高くなる傾向にあり，対策を講じる必要がある.

以上の議論のもと，以下のように高速化手法を提案する.

4.1 目的地点周辺範囲の設定

適当に設定された目的地点周辺の範囲を，AD (Area around the Destination) と呼称する．従来は，目的地点の交差点ノード v_g に到着した場合のみシミュレーションが成功したと見なしていたが，本手法では，AD 内に存在する任意の交差点ノードに到着した場合でもシミュレーションが成功したものと見なす．ただし，AD はシミュレーションの進行度によって，逐次更新される.

例 4. 図 5 に AD を利用したシミュレーションの例を示す．AD を目的地点を中心とした半径 a の円とした場合，AD は図中の赤円のようになる．図 5(a), (b) ではいずれも，AD 内の交差点ノードに到達しているため，シミュレーションは成功したと見なされ，それまでの経路がプレイアウト経路として評価される. ■

4.2 目的地点周辺範囲の更新

AD を目的地点を中心とした半径 a の円とする．基点となる交差点ノード v_{now} と目的地点の交差点ノード v_g 間の直線距離を $D(v_{now}, v_g)$，c を正の定数とするとき，AD の

表 2: 実行環境.

OS	Windows 10 Pro
CPU	Intel Core i7 2.90 GHz
RAM	16.0 GB
Python	Python 3.7.0

表 3: 実験結果.

経路推薦手法	提案手法					従来手法 [13] (AD なし)
	$c=2$	$c=3$	$c=5$	$c=7$	$c=10$	
平均評価点	1.6	1.5	2.6	3.1	2.9	3.2
平均実行時間	82.8 s	306.2 s	1377.2 s	1713.9 s	2665.5 s	5090.7 s

円の半径 a は式 3 で更新される.

$$a = \begin{cases} \frac{D(v_{now},v_g)-100[\mathrm{m}]}{c} & (D(v_{now},v_g) \geq 100[\mathrm{m}]) \\ 0 & (D(v_{now},v_g) < 100[\mathrm{m}]) \end{cases} \quad (3)$$

ただし，$a=0$ のときは AD は存在せず，従来通り目的地点の交差点ノードに到達したときのみシミュレーション成功と見なす.

5. 評価実験

提案手法を Python3 で実装し，経路推薦実験を実施した．実行環境を表 2 に示す．実験に使用する地図は，被験者が慣れているエリアである高田馬場駅周辺の地図 (交差点ノード数: 10614，エッジ数: 11936，ランドマーク数: 130) を選んだ.

5.1 実験方法

提案手法について，AD 更新式中の定数 c を変えながら，アルゴリズムの実行時間を計測した．その後，推薦された経路を 0，1，2，3，4 の 5 段階で評価する (最低評価が 0，最高評価が 4)．また比較のため，従来の P-UCT 手法 [13] でも同様に計測する．計測は 1 条件に対して，出発地点と目的地点を変更しながら各 10 回実施し，実行時間および評価点の平均をとる．出発地点と目的地点は，いずれも直線距離にして約 1km になるペアを選んだ.

5.2 実験結果

実験結果を表 3 に示す．また，一部の推薦された経路を図 6 に示す.

表 3 より，定数 $c=7,10$ のときにおいて，提案手法は評価点に大きな影響を与えずに実行時間の削減に寄与できることが確認できた．特に $c=7$ のとき，平均実行時間は従来手法の約 34%程度にまで抑えられた．シミュレーションの成功条件が緩和されたことで，シミュレーション失敗による試行ロスが減少したためだと推測される．図 6 を見ても，定数 c の値が適切であれば，概ね従来手法と同じ経路が作成できていることが分かる.

6. おわりに

本稿では， 目的地点周辺範囲 (AD) を定義・導入することによって，モンテカルロ木探索を用いた経路推薦手法 (P-UCT 手法) [13] の高速化手法を提案した．提案手法では，目的地点の交差点ノードに到達せずとも，その周辺範囲である AD 内に存在する任意の交差点ノードに到達すればシミュレーションが成功と見なされる．このため，従来に比べて高確率でシミュレーションを成功させることでき，より効率的なアルゴリズムが実現できる．評価実験の結果，AD 更新式中の定数が $c=7,10$ である場合に，提案手法は評価点に大きな影響を与えずに実行時間の減少に寄与できることが確認できた．特に $c=7$ のとき，平均実行時間は従来手法の約 34%程度にまで抑えられた.

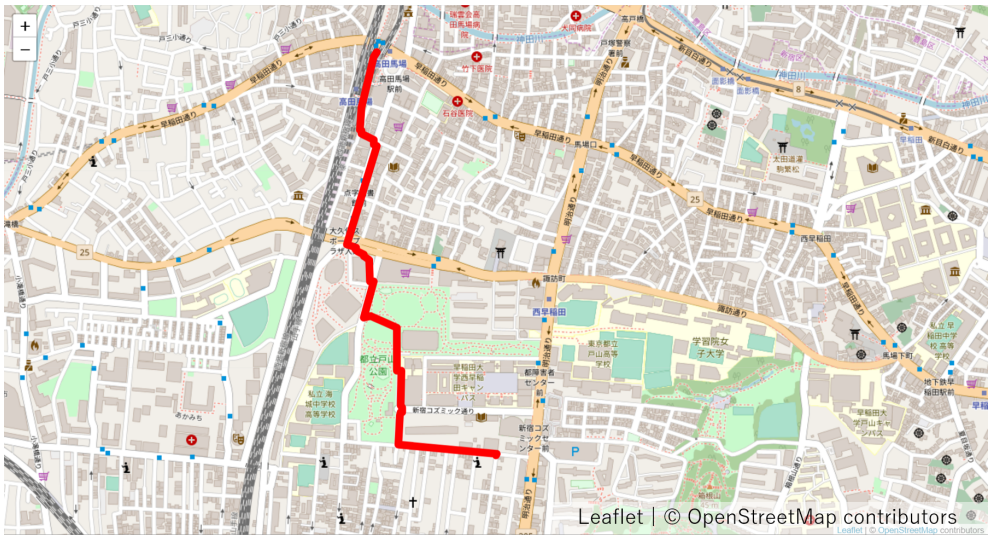
今後の課題は，より大規模な実験を実施して提案手法の有効性を確認することである.

謝辞

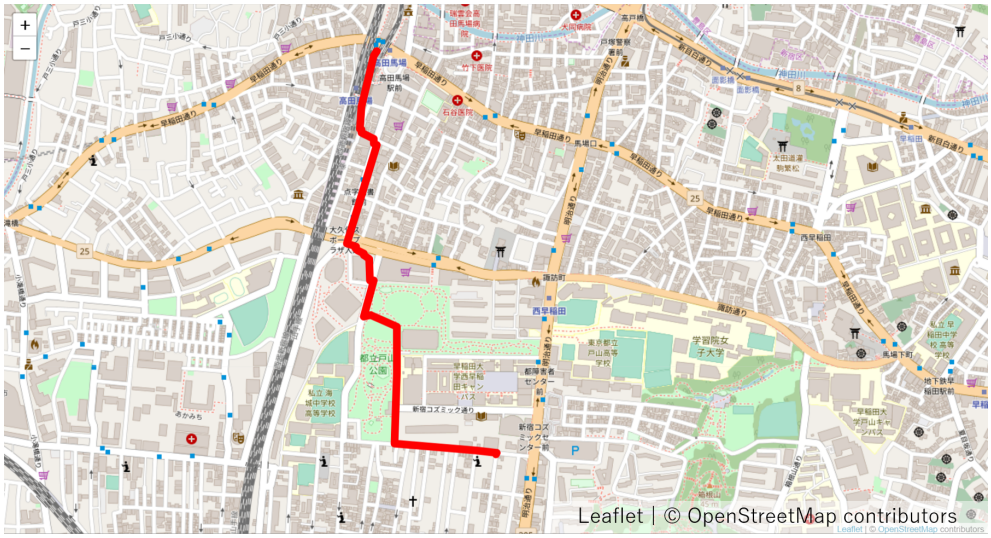
本研究は, 一部, JST CREST (Grant No. JPMJCR19K4) の支援を受けたものである.

参考文献

[1] Yahoo Japan Corporation, “Yahoo! MAP.”https://map.yahoo.co.jp/.
[2] NAVITIME JAPAN, https://www.navitime.co.jp/.
[3] M. Matsuda, H. Sugiyama, and M. Doi, “A personalized route guidance system for pedestrians (in japanese),” *IEICE Transactions*, vol. 87, no. 1, pp. 132–139, 2004.
[4] K. Hara, and H. Kanoh, “Hybrid genetic algorithm for dynamic multi-objective route planning with predicted traffic in a real-world road network,” in *Proceedings of the 10th Annual Conference on Genetic and Evolutionary Computation*, pp. 657–664, 2008.
[5] Y. Kurata, Y. Shinagawa, “CT-Planner5: a computer-aided tour planning service which profits both tourists and destinations,” in *Proceedings of Workshop on Tourism Recommender Systems,* vol. 15, pp. 35–42, 2015.
[6] Y. Sawayanagi, and R. Hamakawa, “Calculation of the user optimum path in mobile navigation with ambiguous destinations,” *IPSJ Technical Report, MPS-73*, pp.



(a) 提案手法 ($c = 7$ のとき).



(b) 従来手法 [13].

図 6: 経路推薦結果例.

25–28, 2009.

[7] H. Kanoh, N. Nakamura, and T. Nakamura. “Route selection with unspecified sites using knowledge based genetic algorithm,” *Transactions of the Japanese Society for Artificial Intelligence 17*, pp. 145–152, 2002.

[8] Y. Akasaka, and T. Onisawa, “Individualized pedestrian navigation using fuzzy measures and integrals,” in *Proceedings of 2005 IEEE International Conference on Systems, Man and Cybernetics*, vol. 2, pp. 1461–1466, 2005.

[9] L. Teng, T. Izumi, X. Lu, and F. Wakui. “A Method of the Optimum Route Search by Fuzzy-AHP.” *IEEJ Transactions on Electronics, Information and Systems 133*, pp. 1269–1276, 2013.

[10] C.B. Browne, E. Powley, D. Whitehouse, S.M. Lucas, P.I. Cowling, P. Rohlfshagen, S. Tavener, D. Perez, S. Samothrakis, and S. Colton, “A survey of monte carlo tree search methods,” *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 4, no. 1, pp. 1–43, 2012.

[11] S. Gelly, L. Kocsis, M. Schoenauer, M. Sebag, D. Silver, C. Szepesvri, and O. Teytaud, “The grand challenge of computer Go: Monte Carlo tree search and extensions,” *Communications of the ACM*, vol. 55, no. 3, pp. 106–113, 2012.

[12] R. Coulom, “Efficient selectivity and backup operators in monte-carlo tree search,” in *Proceedings of International Conference on Computers and Games*, pp. 72–83, 2006.

[13] Y. Ishizaki, T. Takayama, and N. Togawa, “A route recommendation method based on personal preferences by Monte-Carlo tree search,” in *Proceedings of IEEE International Conference on Consumer Electronics in Berlin (ICCE-Berlin)*, pp. 404–409, 2019.

[14] Zenrin, `https://www.zenrin.co.jp/english/index.html`.

[15] M.M. Adankon and M. Cheriet, “Support vector machine,” pp.1303–1308, Springer, 2009.