

スポット推薦を伴う経路推薦手法の提案

Proposal of Route Recommendation Method containing Spot Recommendation

柴田 祐樹¹ 高間 康史¹

Hiroki Shibata¹ and Yasufumi Takama¹

¹ 首都大学東京大学院システムデザイン研究科

¹Department of System design in Tokyo Metropolitan University

概要: 観光案内などでは、訪問すべきスポットだけでなく訪問順序や移動経路も同時に決定する必要がある。この様な、スポット推薦を伴う経路推薦は、観光の分野や日常生活においても需要が高いにも関わらず、問題の定式化の難しさから一般化された手法は十分研究されていない。目的地を巡回する問題については巡回セールスマン問題(TSP)として多くの研究がなされているが、事前に決定したスポット全てを通る経路を発見するためそのまま適用することはできない。本研究ではスポット推薦を伴う経路推薦に対し、確率場による定式化を行い、焼きなまし法等を用いた解法を提案する

1 研究背景

交通手段、情報収集手段の発達に伴い、個人が気軽に未知の土地を訪問できるようになってきている。しかしながら、個人の趣味嗜好というよりは知名度の高い場所が優先的に選択肢へ上がりやすいという昔からの状況は依然として変わらず、当該の観光地が混雑する状況を招き、より個人の趣味に一致した場所や経路の選択機会を喪失している。さらにこのような状況が続けば、観光地としての価値を追求する動きが起こり、その土地本来の良さよりも宣伝性の良さを狙った街作りになる等の弊害も想定され、訪問者、現地民双方にとって良い状況とは言えない。このようなことは旅行等の比較的大きなイベントでなくても、何気ない街や自然の散策、テーマパークの楽しみ方等でも見られる。また、スポットを最短時間でより多く回れば個人の満足度が高くなるというわけでは必ずしもなく、途中通過する街並みや、道の景観等も観光、散策において重要な要素である。従って、個人の嗜好や時間的制約などに基づく、画一的ではない経路推薦手法が必要と考える。

観光経路の推薦では訪れるスポットの選択、訪れる順番、総所要時間等に関する要望、スポット以外の道や景観、混雑状況に対する暗黙的な好みの反映等多様な要求を含んでおり、一般に取り扱うことが難しく、定式化の研究は十分になされていない。目的地を巡回する問題については巡回セールスマン問題 (TSP: Traveling Salesman Problem) [1] として多くの研究がなされているが、事前に決定

したスポット全てを通る経路を発見するため、所要時間のバランスを見ながら訪問スポットを選択するといったような調整を行いにくい。ノードに価値を付加し、総経路負荷が目標値を超えない中で、価値の総和が最大となる経路を選ぶ問題は Selective Traveling Salesman Problem (STSP) [6] [7] として定式化されているが、本問題は所要時間が目標値を超過することが許されず、制限時間の明確でない観光案内へそのまま適用するには不適當である。また、スポットの価値と時間依存性を含む問題を観光経路最適化問題 (ORPS : Optimal Routing Problem for Sightseeing) として定式化することが提案されている[2]。より実用的なシステムとしては CT-Planner [3] が提案されており、各スポット間を結ぶ最短経路をあらかじめ求めておき、スポットをノード集合とする完全グラフの中で、遺伝的アルゴリズムを用いて観光経路を求める手法が用いられている。

これらの既存研究は観光経路の決定を、スポットをノードと見立てたグラフ上の経路問題を解くことに帰着している。文献[2]の定式化はノードの選択や順列の定式化が複雑となっており、一般性に欠ける。また、文献[3]の手法はスポットのみに着目しており、経路に対するユーザの好みに応じた推薦を行うことは想定していない。本稿では、経路を構成する辺に対して目的関数を作り、これをエネルギー関数に含む Boltzmann 分布から経路が生成されるとモデル化する。既存研究ではスポット間の経路は移動負荷を持つものとして幾何学構造と切り離されて考えられているが、本稿では地図上のすべての道にユーザの好みを反映するという

大規模な問題を想定している。また、確率場として定式化することで各種統計、機械学習手法が適用可能となる。本稿では、解法として局所最適化法を用い、確率場による定式化が Simulated Annealing 法の導出を容易にすることを示す。局所最適化法は滑らかな目的関数を持つ問題の効率的解法として知られており、これを用いることは将来的に有用であると考えられる。目的関数の作り方を工夫することで、局所最適化法を用いた場合でも Local Minimum 問題の発生を緩和できる手法を提案する。

2 数学的手法と関連研究について

初めに、本稿で用いる数学的表記について述べる。また、Boltzmann 分布を用いた定式化を行うので、これについて説明し、その後経路問題を扱った関連研究について述べる。

集合 A とその直積集合 $B_A = \{(x, y) | x \in A, y \in A\}$ の関係を $B_A \subseteq A \times A$ と表記する。集合 A と集合 B の差集合 C_{AB} を $C_{AB} = A \setminus B$ と表記する。確率変数は太字立体で書き、その実現値は斜体であらわす。また、多次元変数 \mathbf{x} に添え字をつけた $x_i \in \mathbf{x}$ はその要素を意味する。多次元離散確率変数 $\mathbf{x} = (x_1, x_2, \dots, x_{|\mathbf{x}|})$ の部分空間 $\mathbf{y} \subseteq \mathbf{x}$ に対して、取りうるすべての状態についての和を $\sum_{\mathbf{y}} f(\mathbf{x})$ と書く。また、特定の、複数の要素 $x_i \in \mathbf{x}, x_j \in \mathbf{x}$ については $\sum_{x_i, x_j} f(\mathbf{x})$ のように書き、 x_i, x_j の状態が互いに独立でない場合は $\sum_{(x_i, x_j)} f(\mathbf{x})$ のように書く。

変数 \mathbf{x} についての目的関数 $\phi(\mathbf{x})$ を最小化する問題を考える。ここで、(1)式で与えられる Boltzmann 分布を考える。

$$p(\mathbf{x}) = \frac{\exp\left\{-\frac{1}{T}\phi(\mathbf{x})\right\}}{Z_p(T)} \quad (1)$$

$Z_p(T)$ は分配関数であり、(2)式で与えられる。

$$Z_p(T) = \sum_{\mathbf{x}} \exp\left\{-\frac{1}{T}\phi(\mathbf{x})\right\} \quad (2)$$

$\phi(\mathbf{x})$ についての最適解集合を \mathcal{X}_t とする時、(1)式は温度 $T \rightarrow 0$ の極限で(3)式となることがわかる。

$$p(\mathbf{x}) = \begin{cases} \frac{1}{|\mathcal{X}_t|} & \text{where } \mathbf{x} \in \mathcal{X}_t \\ 0 & \text{where } \mathbf{x} \notin \mathcal{X}_t \end{cases} \quad (3)$$

(1)式の分布を解析的に求め、直接最適解を得ることは困難であり、一般に焼きなまし (SA: Simulated Annealing) 法 [5] を用いて近似解が求められる。焼きなまし法では(4)式を用いて表される条

件付き確率を用いて、温度 T を徐々に低下させながら Markov Chain Monte Carlo 法を適用し、 $T \rightarrow 0$ の分布を近似する

$$p(\mathbf{y} | \mathbf{x} \setminus \mathbf{y}) = \frac{\exp\left\{-\frac{1}{T}\phi(\mathbf{x})\right\}}{\sum_{\mathbf{y}} \exp\left\{-\frac{1}{T}\phi(\mathbf{x})\right\}} \quad (4)$$

次に、観光案内に関連する経路問題について述べるが、これらを Boltzmann 分布により定式化、解く方法は3節の提案手法で述べる。

与えられたノードをすべて巡回する最短経路を求める問題は巡回セールスマン問題 (TSP) として広く知られている。TSP である必要条件是各ノード間の負荷が定義されていて、すべてのノードを巡回する経路のみを解として認めることである。2-opt 法 (k-opt 法) [4] は TSP を解く局所最適化法の1つであり、Fig. 3 に本手法が動作する様子を示す。2-opt 法は、隣接しない2つの辺を選び、経路が開始点から終止点まで連続している状態を保つ入れ替え方のうち、全体の負荷をより小さくするものへ置き換える操作を繰り返すことで最適化を行う。しかし、観光案内では、すべてのノードを巡回する経路が求められるわけではないので、TSP として定式化することはできない。

1 節で述べた文献[2]では STSP の発展型として、観光経路最適化問題を ORPS として定義している。また、ORPS が NP 困難であることを示し、解析的手法による厳密解法と発見的手法による近似解法についても述べている。しかしながら、定式化はノード集合を基準としたものであり、本稿が提案する辺集合による目的関数の定式化に比べ条件設定が必要であり煩雑となっている。CT-Planner[3] では、最適化手法よりもユーザとのインタラクティブ性を重視している。ユーザの好みを知らない状態から開始し、ごく少数の質問に対する回答に基づき推薦をしながら、段階的にユーザが経路を最適化できるような設計指針を持っている。また、ユーザが観光地情報を拡張可能なインタフェースも提供しており、所定の書式に従った表形式ファイルを用意することで新たなスポットを追加できる。最適化手法としては遺伝的アルゴリズムを用いており、データベース内にある任意の観光スポット間の移動負荷をあらかじめ求めて置き、滞在時間と、ユーザごとに異なる価値が与えられた各観光スポットをノード、ノード間の辺に重みとして移動時間が割り当てられた完全グラフとして目的関数を定義、ユーザが指定した所要時間以下で価値の総和が最大となる経路を求めている。

3 提案手法

本稿で対象とする問題は、総移動時間及び目標総移動負荷の制約がある中で、ユーザの経路に対する満足度を最大化するものを推薦することである。推薦スポットは、従来ノードに価値を付加することで表現していた代わりに、本稿では仮想的な辺、あるいはスポット内の実在経路上、例えば博物館の中の通路等に価値を割り当てることで表現する。Fig. 1 にスポット内へ仮想経路を配置した図を示す。スポット内の仮想経路の価値はユーザの好みに応じて変化させることで、喫茶店のような滞在時間の変動が大きいスポットにも冗長な経路を巡回する問題として対応できる。3.1 節では辺に対して定義されるこれらの重みから、目的関数を定義し、これをエネルギー関数とした Boltzmann 分布を用いて、経路は確率的に生成されるとモデル化する。3.2 節では TSP に適用した場合、3.3 節では局所最適化法を適用する場合の目的関数を紹介し、3.4 節で 3.3 節までのものに加え、辺に負荷された価値と総所要時間の制約を考慮した目的関数を示す。

3.1 Boltzmann 分布と経路の定義

地図を構成するノード集合を \mathcal{N} とする。各ノードを結ぶ辺から構成される、終始点を通る有効な経路を含むベクトルを、 $\mathbf{e} = (e_1, e_2, \dots, e_i \dots e_{|\mathbf{e}|})$, $e_i \in \mathcal{N} \times \mathcal{N}$ と表す。Fig. 2 に示す例では、 $\mathcal{N} = \{1, 2, 3, 4, 5, 6\}$, $|\mathbf{e}| = 5$ の場合に以下になる。

$$\mathbf{e} = ((1, 2), (2, 3), (3, 4), (4, 5), (5, 6))$$

3.3 節で示す経路への辺の追加などに対応するため、辺ベクトルは経路に含まれない辺を含む。例えば、 $|\mathbf{e}| = n$ の場合以下になる。

$$\mathbf{e} = ((1, 2), (2, 3), (3, 4), (4, 5), (5, 6), e_6, e_7, \dots, e_n)$$

e_1 から e_5 のみで有効な経路を構成する。辺の方向は第1ノードから、第2ノードへ向かうものとする。 \mathbf{e} の確率変数を \mathbf{e} とし、その分布 $p(\mathbf{e})$ を(5)式で与える。

$$p(\mathbf{e}) = \frac{\exp\left\{-\frac{1}{T}\phi(\mathbf{e})\right\}}{Z_p} \quad (5)$$

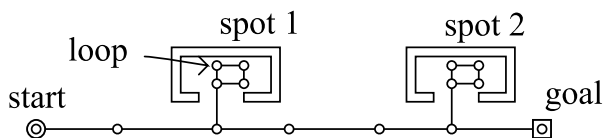


Fig. 1. Virtual redundant loop in spots

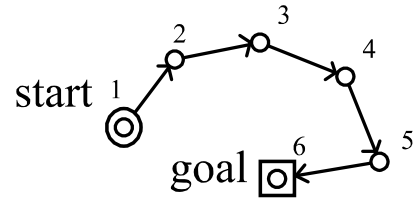


Fig. 2 An example of route.

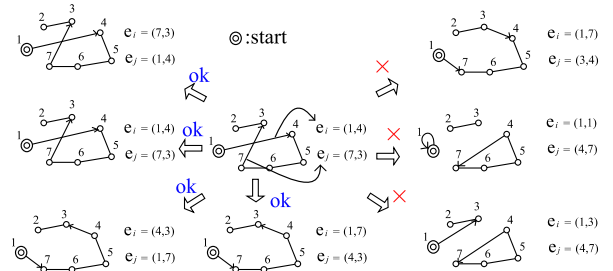


Fig. 3 The restriction to the edge's status.

$\phi(\mathbf{e})$ は目的関数であり、経路 \mathbf{e} に対する制約を定義する。 $\phi(\mathbf{e})$ の値が小さいほど $p(\mathbf{e})$ の値が高まり、そのような経路が選択される確率が高くなることになる。

3.2 TSP への適用

TSP において目的関数は経路上の辺 $e_i \in \mathbf{e}$ に割り当てられた重み $f_c(e_i)$ の総和で与えられ、(6)式となる。一般的にはグラフ上の辺に割り当てる値のことを重みと呼ぶが、3.4 節で説明するように、本稿ではユーザの好みも辺に割り当てることから重みは多次元量となり、それぞれ負荷、好みのように呼ぶことにする。

$$\phi(\mathbf{e}) = \sum_{e_i \in \mathbf{e}} f_c(e_i) \quad (6)$$

TSP を解くために提案された 2-opt 法を用いた SA 法を導出する。局所最適化対象の 2 辺を確率変数 $\mathbf{e}_i \in \mathbf{e}$, $\mathbf{e}_j \in \mathbf{e}$ で表し、これらを除いた辺ベクトルの部分空間 $\mathbf{e} \setminus \{\mathbf{e}_i, \mathbf{e}_j\}$ により条件つけられた、(5)式に対する条件付確率は(7)式により与えられる。

$$p(\mathbf{e}_i, \mathbf{e}_j | \mathbf{e} \setminus \{\mathbf{e}_i, \mathbf{e}_j\}) = \frac{\exp\left\{-\frac{1}{T}\phi(\mathbf{e})\right\}}{\sum_{(\mathbf{e}_i, \mathbf{e}_j)} \exp\left\{-\frac{1}{T}\phi(\mathbf{e})\right\}} \quad (7)$$

ここで、 $\mathbf{e}_i, \mathbf{e}_j$ の取りうる状態について考える。経路が開始点から終止点まで連続している状態を保つ必要があり、その遷移状態の組み合わせは Fig. 3 に示す“ok”と書かれた 4 つしかない。許されない遷移状態は、開始点を含まない閉じた経路ができてしまうもの、経路の巡回方向と一致しない辺の

方向となるものであり、その一部のみを例示した。ある辺 $e_t \in e$ を構成する 2 つのノードを a_t, b_t と表し、(8)式に e_i, e_j の取りうる状態を示す。

$$(e_i, e_j) \in \left\{ \left((a_i, b_i), (a_j, b_j) \right), \left((a_j, b_j), (a_i, b_i) \right), \left((b_i, b_j), (a_i, a_j) \right), \left((a_i, a_j), (b_i, b_j) \right) \right\} \quad (8)$$

e_i, e_j に対して変更があった場合、経路上のほかの辺に対しても巡回方向が正しくなるように修正を加える。

温度 T を減少させながら、あらゆる辺の組み合わせについて(7), (8)式で定義される確率にしたがい辺を入れ替えることを繰り返すことで、最適解を近似的に得ることができる。

3.3 辺の経路への追加と除外

本稿では局所最適化法により観光経路問題を扱う方法を提案する。局所最適化法では一度に扱える変更幅が小さく、複雑な経路問題に対しては一般的に適用することができない。本稿では辺の、経路への追加および除外過程を導入することでこの課題に対処する。経路を局所最適化法により少しずつ拡張していくためには、途中、地図上に実在しない非存在辺を一時的に追加する必要がある。しかし、計算終了時、つまり $T = 0$ の時には非存在辺がすべて実在辺に置き換わっている必要があるため、非存在辺における罰則 $f_p(e)$ を導入し、負荷 $f_c(e)$ と合わせ目的関数 $\phi(e)$ を(9)式とする。また、非存在辺においては $f_c(e) = 0$ とする。

$$\phi(e) = \sum_{e \in e} f_c(e) + \sum_{e \in e} f_p(e) \quad (9)$$

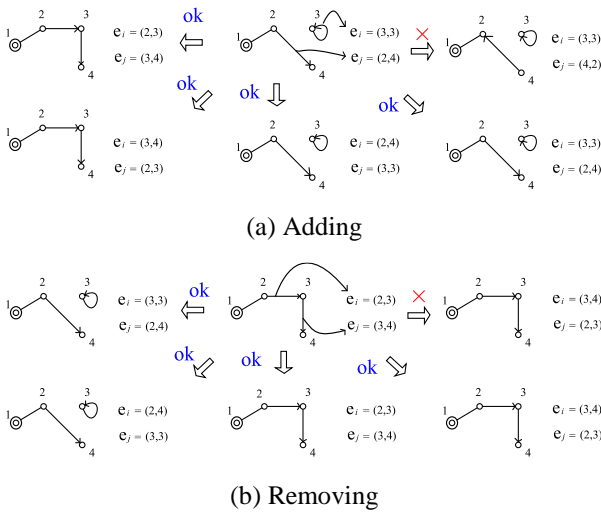


Fig. 4 The restriction of edges' status to add or remove an edge from the route.

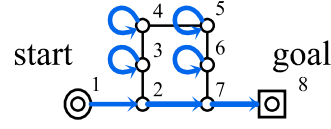


Fig. 5 An example of self loop edges.

罰則 $f_p(e)$ に関して、 s を非存在辺 e の始点、終点を結ぶ実在最短経路、これに含まれる実在辺を $s_j \in s$ とし、(10)式を満たすようにすることで、 $T = 0$ となったときに非存在辺はすべて実在辺に入れ替わる。ただし、実在最短経路が存在しない場合、 $f_p(e) = \infty$ とする。

$$f_p(e) > \begin{cases} \sum_{s_j \in s} f_c(s_j) & \text{where } |s| \neq 0 \\ \infty & \text{where } |s| = 0 \end{cases} \quad (10)$$

例えば、Fig. 2 のにおいて、 $e = (1,4)$ の時、対応する実在最短経路は、 $s = ((1,2), (2,3), (3,4))$ となる。

(10)式を満たす具体的な例として(11)式があり、本稿ではこれを用いた。 C_{pb}, C_{pc} はともに定数である。

$$f_p(e) = C_{pb} \left(\sum_{s_j \in s} f_c(s_j) \right) + C_{pc} \quad (11)$$

また、本稿では 2-opt 法により辺の追加と除外を考慮するが、この場合、辺の取りうる状態として Fig. 3 に加え、Fig. 4 も許可される。(a)は経路に含まれないノード 3 上に自己ループとなる辺があらかじめ配置されている下で、この辺と経路に含まれる辺について状態の変化を考え、経路にとって新しいノード 3 に対して辺が 1 つ追加される過程を表現している。結果として経路に含まれないままの状態も許可される。また、(b)は経路内の隣接する 2 つの辺の状態を、一方の辺が自己ループとなるように変化させることで、1 つのノードが経路から外れる除外過程を表現する。3.2 節と同様にここでも許可される遷移状態を“ok”と表し、許可されないものについては一部の例を示した。この(a), (b)の方法は、初期状態として経路に含まれないノードに自己ループ辺を与えておけば、経路の拡張縮小を繰り返してもノード上に自己ループ辺がある状態を保つことができる。例えば、Fig. 5 は地図上に青で辺ベクトルに含まれる要素を示している、自己ループ辺がノードにそれぞれ 1 つずつ配置されていた場合の例で、この図において以下のようになる。

$$e = ((1,2), (2,7), (7,8), (3,3), (4,4), (5,5), (6,6))$$

各ノードを複数回通る経路を生成可能とする場合には、この自己ループ辺はノードごとに複数個用意しておく。

Fig. 4 (a), (b)に描かれている状態の変化は、同じ規則に従っており、 $\mathbf{e}_i, \mathbf{e}_j$ の取りうる状態を整理すると、(12)式の様になる。

$$(\mathbf{e}_i, \mathbf{e}_j) \in \left\{ \left((a_i, b_i), (a_j, b_j) \right), \left((a_j, b_j), (a_i, b_i) \right), \left((a_j, b_i), (a_i, b_j) \right), \left((a_i, b_j), (a_j, b_i) \right) \right\} \quad (12)$$

(12)式を見れば、 $a_j = b_j$ もしくは $a_i = b_i$ の場合に自己ループ辺となり、辺の追加による経路の拡張に対応する。そうでない場合、 $\mathbf{e}_i, \mathbf{e}_j$ は隣接するので、 $a_j = b_i$ もしくは、 $a_i = b_j$ となる場合は辺の削除に対応し、経路の縮小を考慮できていることがわかる。自己ループ辺を配置しておくことで、経路に対して辺が追加、除外されることはあっても、 \mathbf{e} を構成する要素数は固定されることから、辺ベクトルとして扱うことができ、経路を含む辺の集合に対する確率場を、経路の拡張縮小を考慮したうえでも定式化することができる。なお、自己ループ辺を配置しておくことは、これら定式化における利便性のために導入したもので、実装上必ずしも必要なものではない。

なお、(10)式を満たさない場合、非存在辺が最終的に残る可能性がある。 $f_p(\mathbf{e})$ を幾何学距離の関数で近似することも考えられるが、(10)式を常に満たすために、 $f_p(\mathbf{e})$ を過大に設定する必要がある。この時、非存在辺への遷移確率を低下させ、経路探索を行いつらくなる。すなわち、 $f_p(\mathbf{e})$ が(10)式の右辺に近いほど経路探索を行いやすくなる。 $f_p(\mathbf{e})$ はすべてのノード対に対して計算しておく必要があり、その記憶量、計算量は $O(|N|^2)$ となる。しかしながら、通常計算開始時に $f_p(\mathbf{e})$ は一度だけ計算しておけばよいので、この計算負荷は大きな問題とならない。

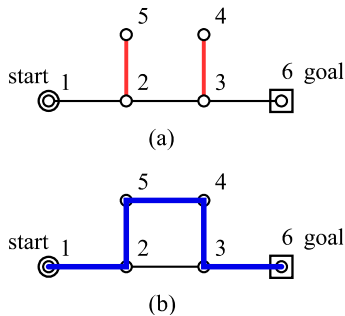


Fig. 6 An example where we have to add $f_d(s_j)$

3.4 緩和 STSP

本項ではユーザの辺に対する好みを $f_d(\mathbf{e})$ で与える。既存研究ではノードに対してスポットを表現する値が付加されていたが、本研究では経路負荷を辺に対する関数 $f_c(\mathbf{e})$ として与えており、これとの一貫性を取るために辺に対してスポットを割り当てる。同一ノードを複製した2ノード間に距離が0の辺を考えることで、ノードに対して割り当てたものと等価な問題を考えることができる。ユーザの好み $f_d(\mathbf{e})$ を取り入れ、目標移動負荷値 C_c を導入したエネルギー関数を(13)式で与える。

$$\phi(\mathbf{e}) = \frac{\beta}{2\sigma^2} \{F_c(\mathbf{e})\}^2 + (1 - \beta) \left\{ F_c(\mathbf{e}) - \frac{\sigma^2}{2} \right\} - \sum_{\mathbf{e}_i \in \mathbf{e}} f_d(\mathbf{e}_i) + \sum_{\mathbf{e}_i \in \mathbf{e}} f_p(\mathbf{e}_i) \quad (13)$$

ただし、条件変数 β 、総移動負荷 $F_c(\mathbf{e})$ は(14)式で与えられる。

$$F_c(\mathbf{e}) = \left| C_c - \sum_{\mathbf{e}_i \in \mathbf{e}} f_c(\mathbf{e}_i) \right|, \quad \beta = \begin{cases} 1 & \text{where } \sigma^2 > F_c(\mathbf{e}) \\ 0 & \text{where } \sigma^2 \leq F_c(\mathbf{e}) \end{cases} \quad (14)$$

STSP では $\phi(\mathbf{e})$ が C_c で表される目標移動時間を少しでも上回ることを認めないが、観光における経路推薦では妥当な近傍解は許容されることを考え、(13)式のように総移動負荷 $F_c(\mathbf{e})$ に対して最小値を持つエネルギー関数を導入する。また、(13)式は移動負荷の増加に見合う $f_d(\mathbf{e}_i)$ の値が得られるスポットに対し経路の拡張を許可する。 σ^2 は拡張規模の許容範囲を表現する。また、 $f_d(\mathbf{e}_i)$ を導入したことにより、 $f_p(\mathbf{e})$ の定義式(10)式を(15)式に変更する。また、(10)式を満たすための近似式として(11)式の代わりに(16)式を用いる。

$$f_p(\mathbf{e}) > \sum_{s_j \in \mathbf{e}} \{f_c(s_j) + f_d(s_j)\} \quad (15)$$

$$f_p(\mathbf{e}) = C_{pb} \left(\sum_{s_j \in \mathbf{e}} \{f_c(s_j) + f_d(s_j)\} \right) + C_{pc} \quad (16)$$

$f_d(s_j)$ は目的関数に対して減算されているので、罰則関数に加算することは奇妙に思える。Fig. 6に $f_d(s_j)$ を加算しなければならない状況の例を示す。(a)において、赤く示されている辺は高い価値が設定されている辺であり、 $f_d((2,5)) = f_d((5,2)) = f_d((3,4)) = f_d((4,3)) = 3$ ほかの辺 \mathbf{e} については

$f_d(e) = 0$ とする. 実在辺の負荷は 1 とする. (13)式において, $\sigma^2 \rightarrow 0, C_c = 0$, (付録に詳細を記載) また (11)式に $C_{pb} = 1, C_{pc} = 1$ を用いるとして, Fig. 6 (b) に青で描かれた, 非存在辺を含む経路 $\mathbf{e}^{(*)} = ((1,2), (2,5), (5,4), (4,3), (3,6))$ の (13) 式の値は $\phi(\mathbf{e}^{(*)}) = 2$, 望まれる実在辺のみの経路 $\mathbf{e}^{(t)} = ((1,2), (2,3), (3,6))$ に対する (13) 式の値は $\phi(\mathbf{e}^{(t)}) = 3$ となり, $\phi(\mathbf{e}^{(*)}) < \phi(\mathbf{e}^{(t)})$ であることから $\mathbf{e}^{(*)}$ が選ばれてしまうことがわかる. 同じ条件で (16)式を用いた場合は, $\phi(\mathbf{e}^{(*)}) = 8, \phi(\mathbf{e}^{(t)}) = 3$ であり, $\phi(\mathbf{e}^{(*)}) > \phi(\mathbf{e}^{(t)})$ となるので, $\mathbf{e}^{(t)}$ が選ばれることがわかる. また, (11)式の罰則定数 $C_{pb} = 1, C_{pc} = 1$ をいくら大きくしても, $f_d(e)$ の大きな場所があれば, 非存在辺が残る可能性をなくすることはできない. 定性的には, 非存在辺を通してでも, 好みの経路を通った方が目的関数の値を小さくできるような状況避ける必要がある, ということと言える

4 評価実験

3.2 節で説明した TSP へ適用した結果を Fig. 7 に示す. (a)は $T = 0$ で計算した結果で, 2-opt 法に一致するものであり, (b)は $T = 0.093$ で計算した結果で焼きなまし法となる. 20 行 20 列の幅 1 で置かれた等間隔配置ノードからなる, ノード数 400 の完全グラフに対し, 全ノードを通り負荷最小の経路を求めている. この時, すべてのノード対に幾何学距離に等しい負荷が設定されており, 斜めの辺が 1 つもない経路が最適解となる. (b)の焼きなまし法を用いた結果がより最適解に近いことがわかる.

Fig. 8 に辺の追加と除外の仕組みが機能している様子を示す. 実在辺は薄い灰色で表示されており, これら以外のノード対を結ぶ辺はすべて非存在辺である. 開始点と終止点が設定されており, 開始点と終止点がつながる経路のうち目的関数(9)式が最小となる経路を求める. この問題の最適解は, 37 ノードすべてを実在辺のみで通る経路である. (a)に初期状態を示しており, 初期経路として開始

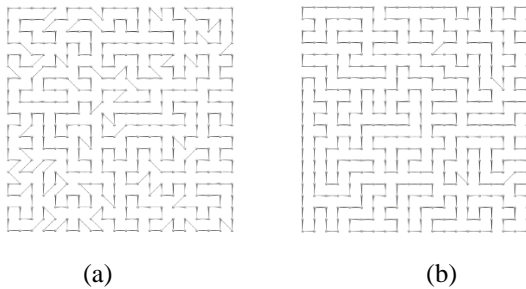


Fig. 7 The result of TSP with SA and Markov Chain Monte Carlo.

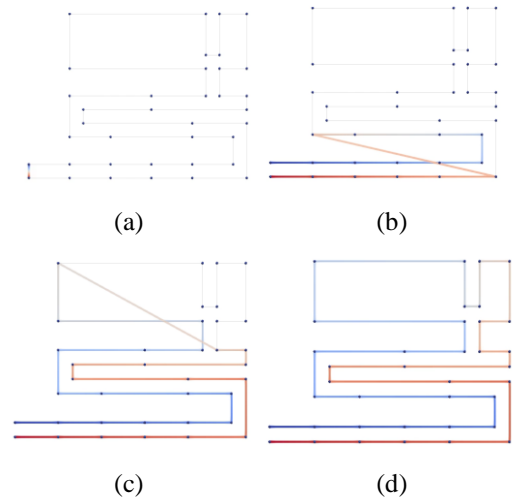


Fig. 8 The result of edge adding and removing. $T = 0$

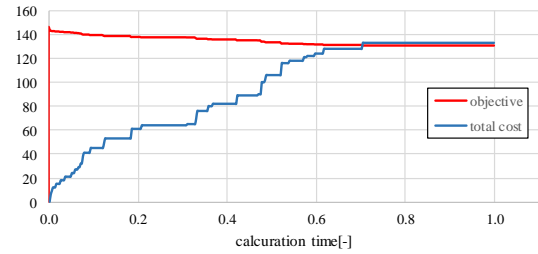


Fig. 9 The graph of total value of $f_c(e)$ in the route and objective function. Red is the objective function while blue denotes the total value of $f_c(e)$.

点と終止点を直接結ぶ非存在辺からなる経路を与えている. $f_p(e)$ を定義する(11)式のパラメータとして, $C_{pb} = 1.1, C_{pc} = 0.1$ を用い, $T = 0$ とした.

この問題は, 罰則関数 $f_p(e)$ を幾何学距離で近似した場合にうまくいかない複雑な経路上において, 計算終了時に非存在辺が残らないことを検証することを目的としたものである.

Fig. 8 の (b), (c), (d) はそれぞれ 40, 200, 400 ステップにおける経路である. 開始点に近い辺を赤, 終止点に近い点を青で, グラデーションを用いて可視化している. 局所最適化を繰り返すうちに非存在辺と実在辺の追加と除外を繰り返し, 最終的に実在辺のみが残っていることがわかる. また, Fig. 9 にエネルギー関数と総コストの変化過程を示す. 総コストの上昇に対して, 非存在辺が消滅することでエネルギー関数は常に減少していることがわかる.

Fig. 10 に(13)式のエネルギー関数で表される緩和 STSP を解いた結果を示す. ここでは問題を簡潔にするため, 18 行 18 列の等間隔におかれたノードからなるマス目上の地図を想定した. ノード数は

終始点を含め 326 である。 $\sigma = 5, C_c = 60$ とし、赤く表示されている部分は $f_d(e) = 2, f_c(e) = 1$ の辺であり、そのほかの薄い灰色で描画された辺は $f_d(e) = 0, f_c(e) = 1$ の辺である。これら以外のノード間をつなぐ辺はすべて非存在辺である。(a)は計算開始直後の状態であり、温度 T が高いことにより発見的な経路探索を行っていて、非存在辺も含んでいる。斜めの辺はすべて非存在辺である。(b)は計算終了時の状態であり、非存在辺はなくなっており、負荷の目標値として妥当な経路でスポットを巡回する経路を生成していることがわかる。

5 結論

観光経路推薦に要求される、スポットの選択性、目標移動負荷および時間、経路の推薦を含む問題に対して、辺の集合に対する確率場を Boltzmann 分布を用いてモデル化し、局所最適化法により経路を生成する方法を示した。また、本手法は、辺に対するユーザの好みに応じた経路生成が可能なこと、ノードであらわされるスポットに対しても仮想的な辺を配置することで経路推薦が行えることも示した。局所最適化法により効率的に解の探索が行える本提案手法は、スケーラビリティに優れるため、広域を対象とした観光案内や、経路上のユーザの好みまで反映させる大規模な問題にも適用可能と考える。今後の展望としては棄却、重点サンプリング[8]を導入し、近傍経路に対する確率場の評価を重点的に行い、遠方への辺検索に要する無駄な探索を抑制することを検討している。また、現在 2 辺の組み合わせによる最適化だけを考慮しているが、より多くの組み合わせを用いた場合について検証する。さらに、スポットの価値が時間依存性を持つ場合にも対応できるようにする。最適化手法だけではなく、実際の経路推薦に適用された場合のユーザの満足度等、ユーザとのインタラクティブ性を考慮し、経路推薦手法としての精度評価指標も検討する予定である。

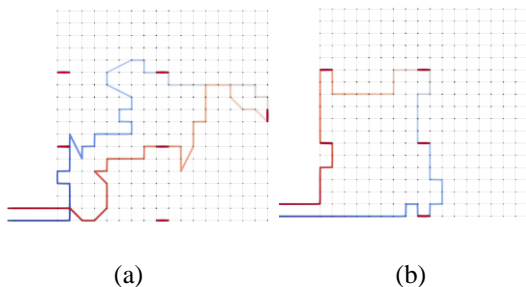


Fig. 10 The result of reduced STSP. The red edges have the weight of $f_d(e) = 2, f_c(e) = 1$ while graies have $f_d(e) = 2, f_c(e) = 1$ of it.

参考文献

- [1] G. Gutin, A. Punnen, "The traveling salesman problem and its variations," Springer Science & Business Media, 2006.
- [2] 松田善臣, 名嘉村 盛和, 姜 東植, 宮城 隼夫, 最適観光経路問題とその解法, 電気学会論文誌 C (電子・情報・システム部門誌), 2004, Vol. 124, No. 7: pp. 1507-1514.
- [3] 倉田陽平, 有馬貴之, 対話的旅行計画作成支援システムの実装と評価, 第 25 回日本観光研究学会全国大会, 日本観光研究学会全国大会学術論文集, 2010, pp. 173-176.
- [4] K. Helsgaun, "General k-opt submoves for the Lin-Kernighan TSP heuristic," Mathematical Programming Computation, 2009, Vol. 1, No. 2-3: pp. 119-163.
- [5] S. Kirkpatrick, C. D. Gelatt, Jr., M. P Vecchi, "Optimization by simulated annealing," Science, 1983, Vol. 220, No. 4598: pp. 671-680.
- [6] G. Laporte, S. Martello, "The selective travelling salesman problem," Discrete applied mathematics, 1990, Vol. 26, No. 2-3: pp. 193-207.
- [7] D. Feillet, P. Dejax, M. Gendreau, "Traveling salesman problems with profits," Transportation science, 2005, Vol. 39, No. 2: pp. 188-205.
- [8] C. M. Bishop, "Pattern Recognition and Machine Learning," Springer, 2006.
- [9] 倉田陽平, 原辰徳, インターネット上での対話的旅行プラン作成支援サービス とその展開可能性, サービス学会第 2 回国内大会, 2014, pp. 191-194.

付録

(14)式の条件のもと、 $\sigma^2 \rightarrow 0$ の極限を(13)式に適用すると、(17)式となる。

$$\lim_{\sigma^2 \rightarrow 0} \phi(\mathbf{e}) = F_c(\mathbf{e}) - \sum_{\mathbf{e}_i \in \mathbf{e}} f_d(\mathbf{e}_i) + \sum_{\mathbf{e}_i \in \mathbf{e}} f_p(\mathbf{e}_i) \quad (17)$$

また、 $F_c(\mathbf{e})$ 中の定数 C_c を 0 とすれば、(18)式となる。

$$\phi(\mathbf{e}) = \sum_{\mathbf{e}_i \in \mathbf{e}} f_c(\mathbf{e}_i) - \sum_{\mathbf{e}_i \in \mathbf{e}} f_d(\mathbf{e}_i) + \sum_{\mathbf{e}_i \in \mathbf{e}} f_p(\mathbf{e}_i) \quad (18)$$