

1. はじめに
2. 既存研究
3. GA の適用
4. 実験
5. まとめ

観光スケジュール探索アルゴリズムへの 遺伝的アルゴリズムの適用

川口 晏璃

June 11, 2021

1.1 本研究の背景

2/12

- 1. はじめに
- 2. 既存研究
- 3. GA の適用
- 4. 実験
- 5. まとめ

背景

観光するユーザは、様々な要求を持っており、それらを満たすことは困難である。そのために開発された探索アルゴリズムは選択肢が多岐にわたるため、現実的な時間で最適解を求めることができない。

目的

本研究では、天気を考慮した観光スケジュール群の探索アルゴリズムに遺伝的アルゴリズムを適用することによって、現実的な計算時間で準最適解を求める。

2.1 既存研究

3/12

既存研究

事前に入手した天気予報情報に基づき、天気に応じて分岐する目的地の列からなるスケジュール木の木構造を求めている。また、ある天気における観光スポットの重要度とその時の巡回確率を設定し、スケジュールごとの満足度の期待値を計算している。

データベースには道路地図および目的地のデータ、天気予報データが保持されている。

ユーザ側の入力として、滞在日数や出発・帰着時間、観光スポットの重要度、観光スポットにおける時間制約を与える。

既存研究のアルゴリズム

- 1 欲張り法
- 2 帰着スケジュール生成
- 3 局所探索法

2.2 アルゴリズム

4/12

- 1. はじめに
- 2. 既存研究
- 3. GA の適用
- 4. 実験
- 5. まとめ

欲張り法

入力に欲張り法を適用し，初期のスケジュール木を生成する．帰着時刻を考慮せず，完全木を生成する．スケジュール木の各ノードからの天気の分岐先のノードそれぞれに対し，最も満足度が高いと予測される目的地を割り当てる．

帰着スケジュール生成

欲張り法で生成されたスケジュール木から，帰着時刻までに帰着地点に帰るスケジュール木を生成する．

局所探索法

帰着スケジュールを考慮したスケジュール木に局所探索を適用し改善する．

3.1 遺伝的アルゴリズム

5/12

1. はじめに
2. 既存研究
3. GA の適用
4. 実験
5. まとめ

遺伝的アルゴリズム

- 1 初期個体の生成
- 2 各個体の期待値の計算
- 3 エリート個体の情報を e 個分保存
- 4 交叉
- 5 突然変異
- 6 生成された各個体の期待値の計算
- 7 手順 2 に戻る

手順 2 では既存研究と同じ計算式を用いる.

3.2 期待値の計算

6/12

期待値を求める式は以下のように与えられる.

$$exp(T_s) = \sum_{j \in L_s} pro(j) \cdot satt(j)$$

L_s はスケジュール木 T_s の葉ノードの集合である.

$pro(i)$ はスケジュール木 T_s におけるノード i への到達確率を返す関数.

$$pro(i) = \begin{cases} 1 & \text{if } i = \text{root} \\ pro(par(i)) \cdot pw(w'_i, dep(par(i))) & \text{otherwise} \end{cases}$$

$satt(j)$ はスケジュール木 T_s において根ノード $root$ から葉ノード j に到達するスケジュール S_j の満足度を与える関数.

$$satt(j) = \alpha \cdot desp(S_j) + \beta \cdot disp(S_j) + \gamma \cdot timp(S_j)$$

3.2 期待値の計算

7/12

1. はじめに
2. 既存研究
3. GA の適用
4. 実験
5. まとめ

$dep(i)$ はノード i の出発時間を返す関数

$$dep(i) = \begin{cases} pt_s & \text{if } i = \text{root} \\ dep(par(i)) + dur'_i + dist(d'_i, d'_{par(i)}) / speed & \text{otherwise} \end{cases}$$

$desp(S_j)$ は、スケジュール S_j において制約条件を満たす目的地における重要度の合計値を返す関数

$$desp(S_j) = \sum_{i \in nod(S_j)} satd(i)$$

$$satd(i) = \begin{cases} pre(d'_i, w'_i, tima(i)) & \text{if } tima(i) \leq rst_i \\ 0 & \text{otherwise} \end{cases}$$

3.2 期待値の計算

8/12

$tima(i)$ はノード $dist(d'_i, d'_{par(i)})/speed$ スケジュール T_s においてノード i の目的地に到着する時間を返す関数

$$tima(i) = \begin{cases} pt_s & \text{if } i = root \\ dep(par(i)) + dist(d'_i, d'_{par(i)}) / speed & \text{otherwise} \end{cases}$$

$disp(S_j)$ は、スケジュール S_j の経路長さによるペナルティ値（負の値）を返す関数

$$disp(S_j) = \sum_{i \in nod(S_j)} dist(i, par(i))$$

$timp(S_j)$ は、スケジュール S_j において帰着時間を超過した時のペナルティ値（負の値）を返す関数

$$timp(S_j) = \begin{cases} 0 & \text{if } i = root \\ tima(j) - pt_g & \text{otherwise} \end{cases}$$

4.1 実験

9/12

- 1. はじめに
- 2. 既存研究
- 3. GA の適用
- 4. 実験
- 5. まとめ

実験方法

実験では、目的地 6 地点から 4 地点を巡回する．ただし、各個体の遺伝子長は 6 とする．巡回しない地点も個体情報に加えることによって、世代交代の際に、巡回路となる可能性を残す．生成する個体数は 20、エリート選択は 2 個体分の情報を保持、手順を 10 回繰り返すとする．

この実験では遺伝子長が 6 と短いため、突然変異が及ぼす影響が非常に大きいものとなる可能性が高い．よって、突然変異は適用しないものとする．

4.1 実験

10/12

表 1 観光スポットリスト

目的地番号	観光スポット	基本重要度	滞在時間(h)	属性
1	東大寺大仏殿	91	2	その他
2	平城京跡	78	1.5	屋外型
3	春日大社	74	1.5	屋外型
4	飛鳥寺	50	1	その他
5	興福寺	79	1.5	室内型
6	法隆寺	100	2	室内型

表 2 目的地属性による重要度係数

天気	屋外型	室内型	その他
晴	1	0.4	0.8
雨	0.2	1	0.8

表 3 各地点の情報

地点名	緯度	経度
奈良ホテル	34.679855	135.834031
東大寺大仏殿	34.6888126	135.8399158
平城京跡	34.6910562	135.7953815
春日大社	34.6812433	135.8482926
飛鳥寺	34.4787657	135.8202946
興福寺	34.6829464	135.8310724
法隆寺	34.6015645	135.7388892

1. はじめに
2. 既存研究
3. GA の適用
4. 実験
5. まとめ

1. はじめに
2. 既存研究
3. GA の適用
4. 実験
5. まとめ

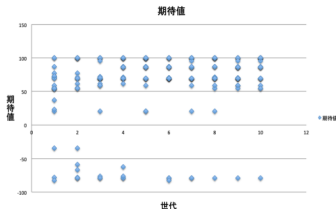


表 4 最も高い期待値とその値を持つ個体数

世代	期待値	個体数
1	100.49	1
2	100.52	2
3	100.52	2
4	100.52	2
5	100.52	3
6	100.52	4
7	100.52	3
8	100.52	5
9	100.52	2
10	100.52	5

期待値が 50 以下の個体の数が，第 1 世代のほうが第 10 世代より多いことが分かる。

表 4 より，第 2 世代以降，期待値は変化していないが，期待値をもつ個体の数が変化していることが分かる。

まとめ

天気を考慮した観光スケジュール群の探索アルゴリズムへ遺伝的アルゴリズムを適用した.

期待値を用いて遺伝的アルゴリズムを実行することにより, より優良な個体情報を取得することができることが確認された. また, エリート個体を残しているため, 優良個体の情報を失われることがない.

交叉させる親をランダムに選んでいるため, 多様性が保たれている.

課題

より多くの目的地に対する実験が必要である.

1. はじめに
2. 既存研究
3. GA の適用
4. 実験
5. まとめ