

障害を考慮したテーマパークのアトラクション経路探索

Theme park attraction pathfinding with obstacle considerations

北村 創拓 (Sotaku Kitamura) *

法政大学情報科学部コンピュータ科学科
sotaku.kitamura.3w@stu.hosei.ac.jp

Abstract

At a theme park, we would like to derive a route that allows us to ride as many attractions as possible. However, it is difficult to find the optimal solution because problems such as temporary suspension of attractions or unusually long waiting times can be expected. Therefore, we propose a new local search algorithm that combines an algorithm for deriving the optimal solution in a short time, which consists of a heuristic construction algorithm and local improvement, with a mandatory visitation orienteering problem that can generate routes including arbitrary attractions. This algorithm can statistically derive which attractions to visit next using historical wait time data. We experiment and discuss the optimal use of data in this algorithm.

1 序論

テーマパークは、人々が楽しみ、思い出に残る体験をする場所として、世界中で人気を博している。

しかし、多くのテーマパークでは、数々のアトラクションやエンターテイメントが存在する。

そのため、閉園時間という限られた時間内で全ての施設への訪問が困難になる。したがって、できるだけ多くのアトラクションを訪問できるよう、効率的な計画が重要となる。

このような地点を訪れる最適な順路を求める問題は主に巡回セールスマン問題で考えられる。

しかし、テーマパークにおいて、全ての地点を訪れるのは時間制限の観点で難しい。そのため、オリエンテーリング問題を使って考える。しかし、オリエンテーリング問題を使用してテーマパークを巡るルートの最適解を求めるのは難しい。

難しくなる理由は2つある。

1つ目は待ち時間が変化し続ける点だ。

一般的に経路探索問題では、経路間の移動時間が一定であると仮定される。移動時間が変わると計算量が増えてしまい、全探索では解を求めるまでに時間がかかってしまう。

2つ目は日によって待ち時間は大きく変わる点だ。休日と平日のアトラクションの待ち時間では、圧倒的に休日のほうが待ち時間が長くなる。また、当日の天気や季節によっても待ち時間は変わる。

3つ目はトラブルが発生し得る点だ。アトラクションは機械のメンテナンスであったり不具合であったりといった要因で、急遽運休となる可能性がある。事前に訪問順番を計画していてもアトラクションを体験できない可能性がある。

これらの要因により、探索空間が指数関数的に大きくなる可能性があり、問題の解決がより困難になる。

そこで、本研究ではオリエンテーリング問題の手法を活用して、テーマパークでの体験を制限時間内で最大化する効率的な戦略の探索を目指していく。

2 オリエンテーリング問題

オリエンテーリング問題 (OP)[3] は、報酬の最大化を目指す計画問題の一つである。無効グラフの特定の出発地点から出発し、与えられた制約条件を満たしつつ、報酬の総和を最大化するルートの発見を目的とする。この問題は、競技者が地図上に示されたチェックポイントを、制限時間内で可能な限り多く訪問してより多くの報酬を獲得する競技オリエンテーリング問題に基づいたものである。この問題は車両ルーティング問題や生産スケジューリング、観光旅行デザイン問題など、様々な分野で利用されている。

* Supervisor: Prof. Katunobu Itou

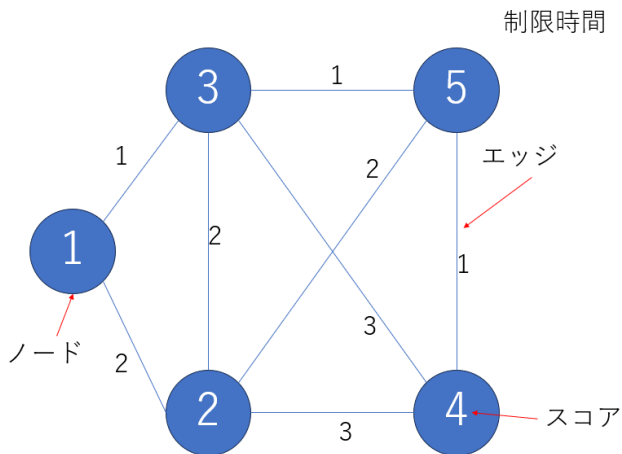


図1 オリエンテーリング問題

オリエンテーリング問題は、 (G, T, R, v_1, v_n, M_3) で定義される。

$G = (V, E)$ は頂点の集合 V と、辺の集合 E を持つグラフである。 $T = v_i \times v_j \rightarrow R^+ \cup 0, \infty$ は頂点 v_i と v_j の間の有限で非負の移動時間を $e_{ij} \in E$ で指定し、それ以外は無限とする。 $R = v_i \rightarrow R^+ \cup 0$ は頂点を訪れたときに獲得できる報酬で、各頂点 $v_i \in V$ に対して有限の非負報酬とする。オリエンテーリング問題の解は開始地点である v_1 と終了地点 v_n を含む頂点を含む部分集合であるハミルトン経路で、その移動時間の合計は H 以下となる。 H は制限時間を表す。

オリエンテーリング問題には一般的にノード、エッジ、スコア、制限時間という要素が含まれる。

ノードは、グラフ上の頂点を表す要素である。つまり、訪れるべき目的地である。今回の研究においてノードは、テーマパーク内の各アトラクションとして表現していく。

エッジはノード間の道である。現在いるノードから隣接しているノードを訪れるためには、このエッジに設定されている移動時間を消費する必要がある。

スコアは各ノードに設定された、訪れたときに獲得できる数値である。スコアは主に重要な場所ほど高い数値が割り当てられる。オリエンテーリング問題はどれだけこのスコアの合計値が高いルートを生成できるかが、問題となる。

制限時間は、テーマパークでの体験を行うために与えられた時間の上限を指す。訪問者は通常、テーマパーク内での滞在時間に制約があり、全てのアトラクションやエンターテイメントを訪れられない。制限時間は、訪問者が体験を最大限に楽しむために合理的な時間内で最適な経路や順序を見つける上で重要な制約条件となる。

本論文ではこれをテーマパークに応用すると、テーマパーク内のアトラクションをノードとみなし、それらを訪れるべき順序を最適化して、制限時間内での最大の満足度を追求する。

また、始点 v_1 及び終点 v_n 、他の頂点 v_i は任意のアトラクションとする。移動時間 $T(v_i, v_j)$ はアトラクション v_j の行列

の待ち時間の和とする。

オリエンテーリング問題の始点と終点と同じ頂点である問題の場合、この問題はオリエンテーリングツアー問題 (OTP) と呼ばれる。しかし、両者の違いは大きくない。OP を OTP とする解決方法としては、始点とい終点の間に移動時間が 0 のダミーのエッジを追加すればよい。

しかしオリエンテーリング問題はアトラクションの待ち時間などの不確実性のある要素を扱う問題のモデル化が困難だ。

その解決策として確率的オリエンテーリング問題 (SOP) が提案された。これは各ノードで獲得できるスコアや移動にかかる時間が確率的に変化する問題で、要素が不確実な問題も扱える。

3 関連研究

関連研究として、動的な移動時間を持つ確率的オリエンテーリング問題のモデルである DSOP と、利用者のリスク許容度を考慮した RS-DSOP アルゴリズム [2] がある。これは 2 つの提案を組み合わせたアルゴリズムである。

しかし SOP にも問題点がある。それはどの時間帯でも移動時間が変わらない点だ。例えば、混雑する昼にあるアトラクションに並ぶ場合、約 30 分かかるとしたら、同じアトラクションを人が少ない夜に並んでも、同じく 30 分かかってしまう。

1 つは、移動時間を確率変数にし、時間帯によって移動時間が動的に変動するようにした。これは従来のオリエンテーリング問題が不確実な要素を扱う問題のモデル化が苦手なためだ。この手法によってアトラクションに訪れる時刻によって待ち時間が変動し、より現実に近いシミュレーションを行える。

2 つ目はリスク許容度を考慮するようにした。リスク許容度とはルートの所要時間が制限時間を超えてもよい可能性を指す。従来の手法ではリスクを承知で混みやすいアトラクションをルートに加えられなかった。そのため利用者がリスク許容度を設定し、そのリスクに応じたルートを生成できるようにした。

この論文では合成データセット、現実世界の遊園地のデータセットのデータで RS-DSOP アルゴリズムを用いて探索している。

また、必須訪問・除外制約付きオリエンテーリング問題 [4] というものもある。これは、オリエンテーリング問題において、最大総時間予算制約を尊重しながら、一組の必須ノードといくつかの任意ノードを訪問する問題だ。

制約緩和による実行可能解と実行不可能解の両方を考慮する専用のタブー探索手順と、早期収束を防ぐためのランダム化突然変異を組み合わせた、記憶的アルゴリズムを提案していた。

4 目的

しかし、RS-DSOP には 3 つ問題がある。

1 つ目はアトラクションを指定できないという点だ。従来の方法では、ユーザーの好みを選考基準に反映していない。いか

にアトラクションの評価値が高いかという基準でしかアトラクションを選んでいるため、訪れたいアトラクションを指定できないという問題がある。

2つ目は報酬がランダムである点だ。この論文では合成データ、現実世界データ共にアトラクションに訪れたときに獲得できる報酬は1から100の間でランダムに決められている。しかし報酬をランダムにしてしまうと現実世界をうまく再現できないという問題がある。例えば、人気のアトラクションは平均待ち時間が長くなり、人気のため報酬も高く設定される。しかしランダムに決定してしまうと待ち時間は長い報酬が低く設定されてしまう可能性がある。このように、報酬をランダムに決定してしまうと現実と乖離した状況で探索を行ってしまう可能性がある。

3つ目は実データで評価していない点だ。参考文献には実際のデータを使用したと記載してある。しかし、文献のアルゴリズムにはアトラクションが急なメンテナンスなどで訪れられなかった場合が考慮されていない。実際のテーマパークではそのような事態が起こり得るため、予測が困難になる。よって訪れられなかった場合も考慮すべきである。

本研究は、実際の過去の各アトラクションの待ち時間のデータを活用し、報酬を待ち時間の応じて振り分け、ユーザーが訪れたいアトラクションを指定できるようにし、特定の日に最適なルートを作成できるアプリケーションの開発を目指した。このアプリケーションにより、利用者は自身の嗜好や制約事項を考慮しながら、限られた時間内で最も効率的かつ満足度の高いアトラクション巡りを計画できるだろう。

5 手法

手法は1つ目の従来研究と同様にDSOP、動的確率的オリエンテーリング問題を使用する。

5.1 データについて

確率的オリエンテーリングにおいて、エッジの値を時刻によって変動させる必要がある。そのため、具体的な待ち時間の推移を記録したデータが必要になる。今回は東京ディズニーシーのアトラクションの各時刻ごとの待ち時間のデータ [1] を使用する。このデータは全アトラクションの15分ごとの待ち時間の推移をまとめたものだ。

時間	ソアリン	トイ・ストーリー・マニア	タワー・オブ・テラー
9:00	110	一時運休	5
9:15	130	一時運休	130
9:30	150	115	65
9:45	155	120	75
10:00	165	120	75
10:15	170	120	105

図2 使用データの一部

このデータを使って、極力現実に近いシミュレーションになるようにしていく。

本論文では、探索アルゴリズムを二つのアルゴリズムを組み合わせて作成した。

5.2 ヒューリスティック構築アルゴリズム

そのうちの一つがヒューリスティック構築アルゴリズムである。このアルゴリズムは未訪問の頂点を現在の経路の最適な位置に挿入を行い、解を構築していく方法である。このアルゴリズムは初期解の作成と、解の改善のために使用される。

Algorithm 1 ヒューリスティック構築アルゴリズム

```

Path = S, G
Metric =  $\frac{\Delta R}{\Delta P + 1}$ 
while len(attractionlist) > 0 do
    selectednode = max(Metric(attractionlist))
    attractionlist.remove(selectednode)
    path.append(selectednode)
end while
return Path

```

まず最初に開始地点からゴール地点まで直接移動するパスを生成する。

次に未訪問のアトラクションの中から、報酬や成功確率を元選ばれたアトラクションを、ゴール地点を除いた末尾部分に挿入する。アトラクションを選ぶために、その時刻での各アトラクションの評価値 X を算出する。評価値 X は以下の式で求められる。

$$X = \frac{\Delta R}{\Delta P + 1} \quad (1)$$

ΔR は該当アトラクションに訪れた時に獲得できる報酬を指している。 ΔP は該当アトラクションをルートに追加した際、制限時間を超過しないかの確率である。この確率は全体のサンプルデータの数で、時間内に収まるサンプルデータの数を

割ったものである。アトラクションを追加する前の確率から追加後の確率で引いたものが ΔP となる。未訪問の各アトラクションでこの計算式をあてはめ、得られた中で最も値の高いアトラクションが、次に訪れるアトラクションとなる。

すべてのアトラクションを挿入する、またはどのアトラクションも挿入できなくなったらアルゴリズムは終了する。

以上がヒューリスティック構築アルゴリズムの手順である。

このアルゴリズムのメリットとして解を導き出すまでの時間が早いという点がある。しかしデメリットとして、解が最適でない可能性がある。このプログラムは計算時の評価値が高いアトラクションを選び続けるというグリーディ法に近い方法である。したがって、効率的なルートを作り出しているとは言いがたい。そのため、このアルゴリズムで作出した解を局所改善で改良する。

5.3 局所改善

局所改善は現在の解から近傍解を生成し、その中で最もよい解を選択して探索を続け、最適解を導き出す方法である。局所改善の手順は以下のとおりである。

Algorithm 2 局所改善

```

Path = HC()
bestPath = tentativePath
searchnum = 0
Metric = randommetric
T = startingtemperature
for 1tosearchnumdo do
  T = T · Δ T
  Z = count/(2 · searchnum)
  tentativePath = 2opt(tentativePath)
  while random ≤ Z do
    removelastrandomtentativePath
  end while
  neighborPath.append(HC(tentativePath))
  Δ R = neighborPath.reward − tentativePath.reward
  if Δ R > 0 then
    tentativePath = neighborPath
  end if
  if tentativePath : reward > bestPath : reward then
    bestPath = currentPath
    count = 0
  else
    count = count + 1
    if count > searchnum then
      Metric = newrandommetric
      count = 0
    end if
  end if
end for
returnbestPath

```

次に、このアルゴリズムの流れを解説していく。まず、初期解を生成する。ここでは先ほどのヒューリスティック構築アルゴリズムを使用して初期解を生成する。次に近傍解を生成する。2-opt 操作を行って近傍解を生成する。2-opt 操作は現在の解からランダムに2つのアトラクションを選択し、それぞれの場所を入れ替える操作である。

次に頂点を確率で削除する。削除する確率は、解の改善が見られなかった回数が多いほど高くなる。具体的には、ランダムな値が、解が改善されなかった回数を、探索回数の2倍の値より下だった場合に頂点の削除を行う。削除する頂点はゴール地点を除いたルートの末尾のアトラクションである。頂点の削除はランダムな値が、計算で出た値を上回るまで行われる。

次に頂点を挿入する。これもヒューリスティック構築アルゴリズムを使用して挿入していく。未訪問のアトラクションの中から制限時間内に収まるように選び、現在の解の末尾に追加していく。

最後に現在の解と生成された近傍解を比較し、よりよい解を現在の解とする。スコアが高い、また終了時間が短い解をよりよい解とする。

近傍解の生成と評価を繰り返し、よりよいルートを発見するものが局所改善である。

また、局所解に陥らないようにするため、解が一定回数改善されなかった場合、メトリックを変更する。メトリックはヒューリスティック構築アルゴリズムで使われる、各アトラクションの評価値を求める式を表している。選ばれるメトリックは従来ヒューリスティック構築アルゴリズムで使われている(1)式の外に4つの式がある。

$$X = \frac{1}{\Delta P + 1} \quad (2)$$

$$X = \Delta R \quad (3)$$

$$X = \frac{(\Delta R)^2}{\Delta P + 1} \quad (4)$$

$$X = \frac{\Delta R}{\sqrt{\Delta P + 1}} \quad (5)$$

解が一定回数改善されなかった場合、5つの中からランダムにメトリックを変更する。

5.4 移動時間

経路 v_k, v_j の移動時間を計算する際、ガンマ分布を使用している。

また、この局所改善の際に使用する待ち時間は待ち時間データのガンマ分布に基づいたものを使用する。正規分布ではなくガンマ分布を使用した理由は、負の数が出ないようにするためである。正規分布は負の方向にも値が分布されるため、負の値が含まれる。しかし、アトラクションの行列の待ち時間に負の値が含まれることはあり得ない。そのため、今回はガンマ分布

を使用した。下のグラフは「ソアリン」というアトラクションの9時45分の待ち時間のガンマ分布である。

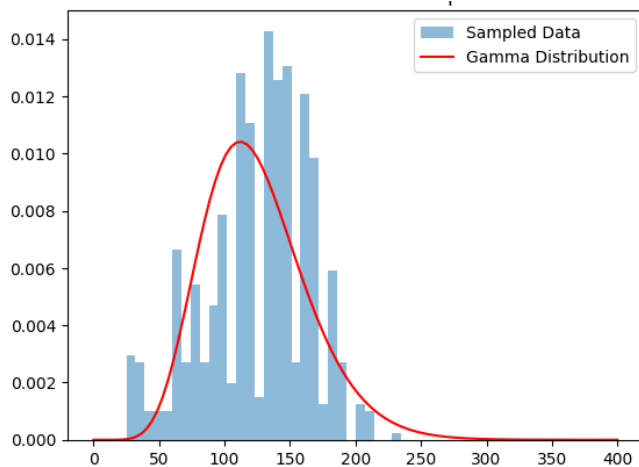


図3 報酬の分布

各時間、各アトラクションごとに待ち時間の各時間ごとに待ち時間のガンマ分布を作成し、1000個のサンプルの中から選んだ1つを待ち時間とする。

5.5 報酬

オリエンテーリング問題において報酬は、次にどのアトラクションに訪れるかを定めるための手法として重要な役割を果たす。

従来の報酬評価方法では、各アトラクションに訪れたときに獲得できる報酬は、ある値の中でランダムに決定していた。

しかし、この報酬決定方法には問題がある。一つ目はシミュレーションが現実に近いものでなくなってしまうからだ。アトラクションの待ち時間は該当アトラクションの人気とともに増えていく傾向にある。従って、アトラクションの報酬をランダムに決定してしまう手法では、より現実に近いシミュレーションを行えない。

二つ目は体験の向上に結び付かないからだ。ランダムに報酬を決定すると、利用者の嗜好を無視してしまう。利用者が行きたいと思っていないアトラクションに訪れてしまうと、利用者の体験を向上する役割を果たしていない。

そのため、人気のアトラクションには高いスコアを設定するようにした。ここで報酬を全て1にしてしまうとただ待ち時間が少ないアトラクションを回り続けるだけになってしまう。また、報酬を平均待ち時間の逆数にしてしまうと、順位が下のアトラクションは最終スコアへの貢献度が低すぎるため、訪れられなくなってしまう。したがって、具体的な決定方法は待ち時間の平均にした。これであれば上位のアトラクションと差が付きづらく、報酬が低めのアトラクションでも選ばれる可能性が残される。

5.6 必須訪問オリエンテーリング問題

これは、オリエンテーリング問題において、時間制限を超過しないようにしながら、一組の必須ノードといくつかの任意ノードを訪問する問題だ。

これを今回のDSOPに組み込む。必須訪問オリエンテーリング問題は以下のように実装できる。

まず、ノードを必須ノードと任意ノードに分ける。必須ノードは必ず訪れるべきアトラクション、任意ノードはどちらでもよいアトラクションとなる。

次に必須ノードを追加する。未訪問の必須ノードの集合からランダムにノードを選択肢、現在の解の最適な位置に挿入する。これを全ての必須ノードを追加するまで繰り返す。

次に任意ノードの追加を行う。これも未訪問の任意ノードの集合からランダムに、現在の解の最適な位置に挿入する。このプロセスは制限時間を超えずに、任意ノードが追加できなくなるまで繰り返される。

以上が必須訪問オリエンテーリング問題の手順である。

これをDSOPに組み込んでいく。具体的にはヒューリスティック構築アルゴリズムのノードを追加する際のプロセスを少し変更する。「全ての未訪問アトラクションから挿入する」のではなく、「まず必須ノードの集合からランダムに追加、その次に任意ノードの集合から追加する」といった変更を加える。

この変更によりユーザーが訪れたいと考えているアトラクションを含んだ解の生成が可能となった。

6 実験

6.1 実験概要

本論文では、求められた最適解と、提案手法によって得られた解を比較を行い、提案手法の有効性を評価する。探索を行う日付は2023年10月1日～10月7日までの一週間と、13日、14日、20日、21日である。アトラクション数は32個で、スコアは2023年の各アトラクションの平均待ち時間とする。HCで探索のシミュレーションを行う日付を変更し、1500回改善するプログラムを25回動かした平均を比較する。このアルゴリズムでは初期解が結果に大きな影響を与える。そのため、どのようにすればより良い初期解を導き出せるのか調査するため、サンプルデータを2種類用意した。一つは全データである2020年12月22日から2023年9月30日までの待ち時間データ、二つ目はコロナの影響で入場者数を制限していた時期を除いた、2022年3月から9月30日までのデータだ。また、各日程の尤度と運休率は以下の表のようになっている。

日付	平均待ち時間	運休率
10/1	27.89	0.21
10/2	30.92	0.21
10/3	30.03	0.23
10/4	26.19	0.20
10/5	27.61	0.21
10/6	39.39	0.27
10/7	40.62	0.22
10/13	27.86	0.29
10/14	39.89	0.21
10/20	38.50	0.21
10/21	38.00	0.20

表1 各日程での平均待ち時間と運休率

この日のデータは運休率が他の日程よりも高い。この日に探索を行うとスコアがどのようになるのか実験を行う。

報酬には2023年の各アトラクションの待ち時間の平均を使用する。

暫定の最適解は25回プログラムを実行した中で最も値が高かったものとする。

これによりどのような結果になるか調査を行った。

6.2 実験結果

結果は以下の表のようになった。

表2: 全サンプルで探索を行ったスコアの平均

日付	スコア平均	暫定最適解
10/1(日)	584.21	662.71
10/2(月)	551.09	604.41
10/3(火)	519.32	600.59
10/4(水)	588.43	68651
10/5(木)	587.98	587.98
10/6(金)	437.41	489.65
10/7(土)	445.21	512.92
.	.	.
10/13(金)	548.02	651.59
10/14(土)	445.21	512.92
10/20(金)	440.1676	488.48
10/21(土)	454.4932	511.0

表3: コロナのみのサンプルで探索を行ったスコアの平均

日付	スコア平均	暫定最適解
10/6	450.46	507.21
10/7	441/26	517.09

平均スコアが、当日の最高スコアの8割ほどの結果を出している。また、金曜日、土曜日のスコアは他の曜日よりも少ない。運休率が高い日は他の人比スコアが減少する傾向にあ

る。サンプルに関する実験だが、コロナ期のみのデータで探索したものは若干スコアが増加する結果となった。

6.3 考察

金曜、土曜日のスコアが減少した理由として、平均待ち時間、運休率が高いことが原因なのではないかと考えられる。この日のプログラムの実行履歴を見ると、予定していたアトラクションを訪れることができないまま制限時間を迎えてしまい、そこで探索が終了している。これは待ち時間が長かったり、運休率が高いため、予定していたアトラクションを訪問することができず、スコアが低くなっていると考えられる。

サンプルに関する実験だが、コロナ禍のデータを除いたデータをサンプルとすると暫定最適解やスコアが全体のデータで探索したものより増加した。これはガンマ分布の精度が上がったためではないかと考えられる。コロナ禍の待ち時間は現在ではありえないほど待ち時間が少ない。そのデータを探索の際に考慮しなくてよくなったため、スコアが上がったのではないかと考えられる。

7 結論

任意のアトラクションを含め、かつ10個前後のアトラクションを訪問できるルートを生成できるアルゴリズムを作成した。その中には人気のアトラクションも含まれている。しかし、改善点が2つ考えられる。1つ目はデータの不足である。今回使用したデータはコロナの影響を受けたものが多い。しかし実験ではコロナ期のデータを含まないで探索を行ったものの方がよいスコアを出すことができている。そのため、最近の傾向に近いデータがさらに必要とされる。2つ目は運休の傾向に応じたルートを生成できるようにすることである。今回のアルゴリズムでは訪問した際アトラクションが運休していた場合、飛ばして次のアトラクションを訪問している。しかしこの処理では一つアトラクションを訪問できなくなってしまい、スコアが減ってしまう。解決策として、データから運休しやすい時間を予測し、それを評価値の計算に加えるなどの方法が考えられる。

参考文献

- [1] deuluxe. ユアトリップ. page <https://urtrip.jp/>, (参照 2024-01-31).
- [2] P. V. D. T. N. H. C. Hoong Chuin Lau, William Yeoh. Dynamic stochastic orienteering problems for risk-aware applications. pages UAI-P-2012-PG-448-458, 2012.
- [3] T. Tsiligrides. Heuristic methods applied to orienteering. pages Journal of Operation Research Society,35(9):797 809, 1984.
- [4] Q. W. Yongliang Lu, Una Benlic. A memetic algorithm for the orienteering problem with mandatory visits and exclusionary constraintss. pages Eur. J. Oper. Res. 268-PG-4-69, 2018.