

## 再帰法を用いた簡易型スキー滑走経路ガイドシステム

小柳 文子<sup>\*1</sup>, 近 匡<sup>\*2</sup>, 池本 満<sup>\*3</sup>

### A simple ski trail guide system using backtracking algorithm

Fumiko KOYANAGI<sup>\*1</sup>, Tadashi KON<sup>\*2</sup>, Mitsuru IKEMOTO<sup>\*3</sup>

**ABSTRACT** : Japanese ski resort areas have been widely recognized as superb with excellent snow conditions, facilities and beautiful natural environment. The Naeba and Kagura Ski Areas, have increased the number of foreign skiers. The convenient destination is located only two hours from Tokyo metropolitan. Kagura Ski Area is linked to Mt. Naeba (1,845m high) via the world's longest gondola (5.5km) and has 20 paths and 19 lifts. On the contrary, it is also pointed out that tourist when using these paths have trouble reaching the right destination.

This paper describes a convenient trail guide system which provides a suitable path for each skier. This system was developed as a java applet for the aim of an easy access even from a personal tablet or a smart phone anywhere/anytime. Since the protocol is simply using backtracking algorithm, profitable and applicable for any platform, system can be easily modified and upgraded.

**Keywords** : ski trail path, guide system, Kagura, backtracking

(Received May 27, 2016)

### 1. はじめに

平成 27 年観光白書<sup>[1]</sup>によれば、2014 年（平成 26 年）の訪日外国人は、前年比 29.4%増の 1,341 万人となり、12 月には過去最高の 1,700 万人に達した。北陸、北海道新幹線の開通を機にリゾート地へのアクセスも格段に良くなり、スキーなどのレジャースポーツも気軽に利用されるようになってきた。

複数の滑走経路があるスキー場は、選択肢が多く楽しめる反面、コースの関係性が分からず迷う人も多い。各スキー場や観光サイトでは、グレンデマップや気象情報など、ホームページを通じて情報発信に務めるようになった<sup>[2],[3]</sup>など。これまでグレンデ情報は、経験者のガイドや、情報紙のグレンデマップなどを利用することが一般的である。最近では、個人が携帯するスマートフォンなどを使う機会も増えてきている。

そこで本研究ではこうしたスキーリゾートに向けて、手軽に滑走ルートの選択が行え、迷うことなく効率的にスキーを楽しめるよう、スキーの滑走経路援助システムの構築を試みた。本システムはjavaを使用し、web上で誰でも利用できることを目標に開発を行った。

### 2. システム構築要件

スキー場の滑走経路探索は、基本的に経路探索問題として捉えることが可能であり、多くの効率的な手法が提案されている<sup>[4],[7]</sup>。スキー滑走経路となる場合には、

- ① 滑走には上位から下位への階層性がある
- ② 同じ所を繰り返し滑走する
- ③ リフトなどを利用し、別のルートへ移動する
- ④ 滑走の選択には個人の好みも反映される
- ⑤ 個人の滑走能力によりルートに制限が加わる

など通常の経路問題にない特徴があり、単純には解決出来ない。更に利便性を考慮すると、スマートフォンなどの小さな画面上でも操作可能なコンパクトなシステムであることが望ましい。

\*1 : 物質生命理工学科助教 (koyanagi@st.seikei.ac.jp)

\*2 : 物質生命理工学科教授 (kon@st.seikei.ac.jp)

\*3 : 平成 27 年度物質生命理工学科卒業生

一方、コンピュータを利用したスキー情報提示システムとしては、これまでスキー場管理者向けの事故情報システム<sup>8)</sup>や、コースの表示と共に仮想体験が可能なシステムとして3Dアニメーションによる試みなどがある<sup>9)</sup>。本研究ではこのシステムによって臨場感を得ることは目的としていない。あくまでも手軽に利用できる滑走径路の選択を目的としており、いわゆる電子版グレンデマップと滑走ナビゲーションシステムの簡易版にあたる。

システム構築要件として利便性を優先するならば、利用者が起動するOSやプラットフォームに依存しないアプリが好ましい。また入力パラメータは限りなく抑え、操作性が高いものが望まれる。更に表示もごく限られた範囲で良く、ここでは地図、ルートと所要時間程度とした。

### 3. スキー・ルート探索手法

2章で述べたようにスキー経路の探索では、選定の基準は多様である。例えばスキーヤーの技量であったり、時間的な制約であったり、または好きなコースを徹底的に滑りたいなど、一定の評価値を与えるのは難しい。通常、最適経路探索では目的関数が与えられて、いくつかの制約条件を満たす解の最大、または最小値を求めるものである。制約を多用すれば、本来沢山ある選択肢を狭め、気ままなスキーヤーの要望を満たすことができない可能性がある。従ってここでの「最適な滑走径路」とは、個人の好みが反映され、且つ多少の選択肢が示されることであるとする。

このようなことから、初期画面では制約時間内に滑走可能なルートを逐次表示し、より広い条件を含むものとした。また他方、希望度の高い昇順ソート機能を用意し、繰り返しの表示がほとんど示されない合理的ルートの表示も行えるようにした。以降で、これらの問題解決のために適用した2つのアルゴリズムについて述べる。一つ目はナップザックアルゴリズムで、選定基準となるスキーヤーの希望度を反映することを目的に用いている。二つ目は再帰法であり、選択の自由度と探索の省力化のために利用している。

#### 3-1 ナップザック問題アルゴリズム

ナップザック問題とは、 $n$ 個のアイテムから成る有限集合 $N$ 、各々のアイテム $I \in N$ の重量 $S_i$ と価値 $V_i$ 、及びナップザックの重量制限 $b$ が与えられたとき、アイテムの合計重量が $b$ を超えないようなアイテムの詰め合わせで、価値の合計が最大のものを求める問題をいう。

#### 3-2 ナップザック問題の定式化

アイテム $I ( \in N )$  をナップザックに詰めるときに1、詰めない場合は0になる、 $\{0, 1\}$ の2進数変数 $x_i$ を使うと、以下の様に定式化することが出来る。

$$\text{最大化} \quad \sum_{i \in N} V_i x_i$$

$$\text{条件} \quad \sum_{i \in N} S_i x_i \leq b \quad x_i \in \{0, 1\} \quad \forall i \in N$$

このようにして、ある制約を満たす整数の組の線形関数を最大化（もしくは最小化）するものを見つけることが出来る。

ナップザック問題のアルゴリズムは、基本的には価値の高いと思われるものから順に取り入れていくアルゴリズムである。もっとも簡単な方法では、単位重量あたりの価値 $V_i/S_i$ の大きいもの順にソートし、総重量が制限重量 $b$ を超えない限り詰めていく。ただし、この解法では最適解を得られる保証はないが、ナップザック問題では問題の制約を満たした解を常に算出する。

ここでは、コースとリフトの関係をナップザック問題に当てはめて考える。今、重量 $S_i$ を所要時間、価値 $V_i$ を希望度とした場合、経路に要する時間はコースとリフトの所要時間の合計としているが、この場合リフトの所要時間が掛かるほど価値が高くなってしまふ。そのため希望したコースの希望度の合計を先に昇順に並べ替え、その希望度が最も高い経路をユーザーに明示的に示すようにした。ユーザーはその順位を元に、候補の中から最も希望に沿う経路を選択するものとする。

次に経路の組み合わせの中から、繰り返しを許容しつつ、希望度を満たす経路探索を行うアルゴリズムについて述べる。繰り返しを許すことは、滑走径路の組み合わせは膨大なものになる。従って多様な組み合わせが考えられる一方、適切な組み合わせを開示できなくなる恐れがある。こういった組み合わせが膨大な問題の解法には、ヒューリスティックな手法による近似解を求めるアルゴリズム<sup>10)</sup>等が利用される。一方、スキーの滑走という設定を考えると、ここでは階層性を飛び越えることがない条件を加味し、探索手法として再帰法を用いて不要な探索手続きを省くものとした。

#### 3-3 再帰法による階乗の定義

一般に再帰法とは、再帰呼び出しと呼ばれる繰り返しを行うもので、関数の中で自身の関数を呼び出す方法である。最も簡単な例としては、階乗を求めるアルゴリズムが挙げられるだろう。 $N$ の階乗は、

$$N! = N * (N - 1)!$$

で定義され、空積である  $0! = 1$  と定義される。階乗は自分自身を使って再帰的に定義するものであり、 $N!$ を求めるには、 $(N-1)!$ を求める必要があり、 $(N-1)!$ を求めるためには  $(N-2)!$ の計算を求め、結局繰り返しながら最後に  $0!$ を求めることになる。再帰法はこの計算の重複を省くことができる。これをjavaで関数定義した例が以下のようである。

```
void fact(int n){
    if(n == 0){
        return 1;
    } else {
        return n * fact(n - 1);
    }
}
```

ここで定義した関数 fact は引数 n が 0 であれば 1 を返し、そうでなければ  $n * \text{fact}(n - 1)$  の計算結果を返す。つまり、fact の定義の中で fact 自身を呼び出していることが分かる。

### 3-4 バックトラック法による経路探索

再帰呼び出しの1つで、経路探索に良く用いられるアルゴリズムがバックトラックである。図1にバックトラックの経路探索の挙動の例を示した。この図では、A地点が最も上位であり、各地点(B~C)へは図のような接続関係があるとする。この場合のバックトラック法による経路探索のアルゴリズムを考える。

まず通常的全探索では、最上位地点 Aから進み、B地点へ向かう。B地点では経路が左右に分かれており、左の経路を選んで先へ進むと行き止まりになる。この時点で、探索をA 地点までいったん戻し、再びB地点に到達後右側の経路へ進むことになる。しかしながら、バックトラ

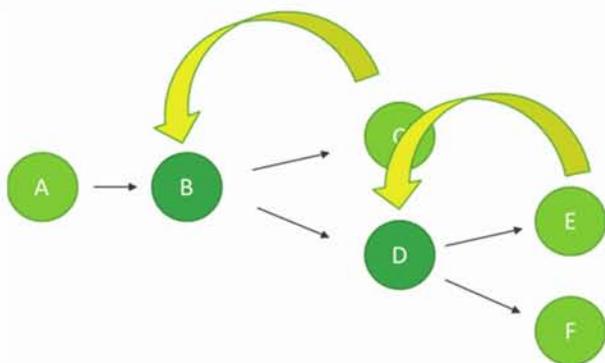


図1 バックトラックの経路探索

ック法では、C地点に到達後は一つ前のB地点に戻ること、別の選択肢である右側の経路を選び直す。

このように試行錯誤を繰り返してゴールにたどり着く方法がバックトラックである。本システムにおいても、バックトラック法による再帰呼び出しを行うことで、経路探索の重複を避け、効率よく探索を行うアルゴリズムとして利用している。

## 4. 対象モデルの特徴

対象としたかぐらスキー場のマップを図2に示した。かぐらスキー場は、新潟県南魚沼郡湯沢町にあり、上越新幹線湯沢駅からおよそ10km、また車でも関越自動車道湯沢ICから8kmと、都心からは2時間程度で利用できるアクセスの良い場所として知られている。標高1,845m、初心者から上級者まで楽しめる全20コースが設定されており、リフトも19コース設置されている。かぐらスキー場は、かぐら、田代、みつまたの3つのエリアに別れており、隣接する苗場スキー場にもゴンドラで行き来できるなど、広大なスケールとバラエティ豊かな景観の中で滑走が楽しめることから、スキー場人気ランキングに常に上位に位置している。

ここで滑走経路の全組み合わせを考えると、それぞれの層におけるノード(節)の数と滑走経路の組み合わせとなる。かぐらスキー場の場合、このノードは最高層から最低層まで13層ある。ここで第4層までを1度だけ滑走すると仮定すると、経路は14ルートあり、2度通れると仮定した場合は93ルートに増える。かぐらスキー場では全13層あるため、その組み合わせ数は重複を許すと爆発的に増加する。以上のことより、再帰法を用いて全探索をせずに計算時間を削減を行った。

### 4-1 スキー場への適用法

バックトラックアルゴリズムをスキー場の経路の探索に応用する場合を考える。表1はリフト・コースの接続

表1 リフト・コースの接続関係

	N1	N2	N3	N4
N1		L1	×	×
N2	C1		L2	L3
N3	×	C2		×
N4	×	C3	×	



図2 かぐらスキー場全体図<sup>[3]</sup>

関係を、表2には各リフトの所要時間の例を表わしたものである。今、コースとリフトの繋がりをもとにノードと定義し、ノードN1とノードN2の連携リフトがみつまた高速リフト2 (L2) となる。またN2からN1, N3, およびN4を接続する滑走路が大会バーンコース (C1)、ファミリーコース (C2)、ゴンドラライン (C3) となっている。滑走経路探索は、この各ノード間の隣接関係をコースとリフトに見立て、所要時間を各価値と与えて計算することになる。

表2 みつまたリフトと所要時間例

L1	みつまた高速リフト2	159s
L2	みつまた高速リフト1	212s
L3	みつまた第2ロマンスリフト	188s
C1	大会バーン	107s
C2	ファミリーコース	291s
C3	ゴンドラライン	180s

### 5. 主要構成データ

ここで用いたリフトとコースデータは、かぐらスキー場HPないしは、湯沢町のパンフレットによる公開情報を利用している。またコースの難易度についても、かぐら

スキー場で定義されたものをそのまま利用している。滑走に必要な所要時間の算定は、著者による計測並びに滑走距離と各コースの傾斜定義より算定している。以上の定義を表3及び表4に、また表5には本システムに用いたスキーヤーの平均滑走速度のデータを示した。リフトの所要時間は、固定循環式チェアリフトの速さを2.3m/sとして、コース長から求めている。また高速リフトに関しては同様に、自動循環式チェアリフトの速さ5.0m/sとして求めている。なお、プログラムに用いた各コースとリフトの接続関係は、付録1に記載した。

コースの滑走時間においても、一般スキーヤーの平均滑走速度をもとに、表3に示すように初級コースでは緩

表3 スキーヤー平均滑走速度

平均滑走速度	緩斜面	中斜面	急斜面
時速(km/h)	13	27	8
秒速(m/s)	3.6	7.5	2.2

表4 各種リフト搬送人数と最高速度

固定循環	一般	最大搬送人数(人/h)	1200
		最高速度(m/s)	2.3
自動循環	高速	最大搬送人数(人/h)	2400
		最高速度(m/s)	5.0

斜面の平均滑走速度 3.6m/sを使い、中級コースでは中斜面の平均滑走速度 7.5m/s、上級コースでは急斜面の平均滑走速度 2.2m/sを使っている。以上のことより、プログラムではコース・リフトの時間データを求めたものが、表5の値である。

表5 ゲレンデコース長と算定滑走時間

難易度	コース名	全長 [m]	滑走時間 [s]
中級	かぐらメインゲレンデ	1,200	160
上級	テクニカルコース	450	203
上級	ジャイアントコース	1,100	496
中級	モーグルバーン	200	27
初級	ゴンドライーストコース	1,000	277
初級	田代アリエスカコース	1,300	360
中級	ダイナミックコース	950	127
上級	チャレンジバーン	500	225
初級	ビギナーズパーク	150	42
初級	ファミリーコース	1,050	291
中級	大会バーン	800	107
中級	Mitumata COLLG PARK	600	80
中級	サンシャインコース	500	67
初級	ゴンドラライン	650	180
上級	下山コース	1,800	811
初級	ゴンドラコース	4,000	1108
中級	パノラマコース	700	93
上級	林間エキスパートコース	700	315
初級	田代・かぐら連絡コース	550	152
中級	田代第一ゲレンデ	350	47
初級	田代第一高速連絡コース	1,120	310
初級	かもしかコース	450	125
初級	田代レディースコース	1,050	291
初級	白樺コース	650	180
初級	スカイラインコース	1,150	319

## 6. プログラムの構成と操作法

### 6-1 プログラムの構成

今回開発したシステムは、javaソースファイルで25KB程度、画像ファイル、データファイル、classファイル、およびhtmlファイルを含めても500KB以下のコンパクトなアプレットとして動作している。従って動作環境を選ばず、タブレット上でも、またスマートフォン上でもほとんど動作速度に機種依存性を感じることはないと考え

られる。主要な構成は以下のものである。

- ・変数定義
- ・画面部品配置の定義
- ・初期画面の設定（画像の読み込み）
- ・アプレットの開始
- ・データの読み込み
- ・入力要求（コース、希望度、制限時間）
- ・目的関数、制約条件、重み関数の設定
- ・経路探索の開始（ナップザック+バックトラック）
- ・経路候補の出力
- ・経路表示の変更の可否

アプレットはURLによってリンク公開可能であるが、今のところ画像の利用許諾等の問題から公開の予定はない。また、今後更なる操作性の向上と、モードの多様化を検討している。

### 6-2 プログラムの操作法

図3にはアプレットの起動画面を示す。画面左上部の制限時間には、初期値として30分を設定している。ユーザーは希望の制限時間を直接分単位で入力する。画面中央には、かぐらスキー場の全景とコース配置が示され、その右側にそれぞれのコースに対する希望度を設定するボタンが配置されている。ユーザーは滑走したいコースの希望度を設定した後、表示ボタンを押すことで、滑走コース順で希望度に沿った経路候補を画面下部に閲覧することができる。また、画面上部右端の昇順ボタンを押すことで、希望度合計値の高いものを上位に表示できるよう、並び替え可能とした。

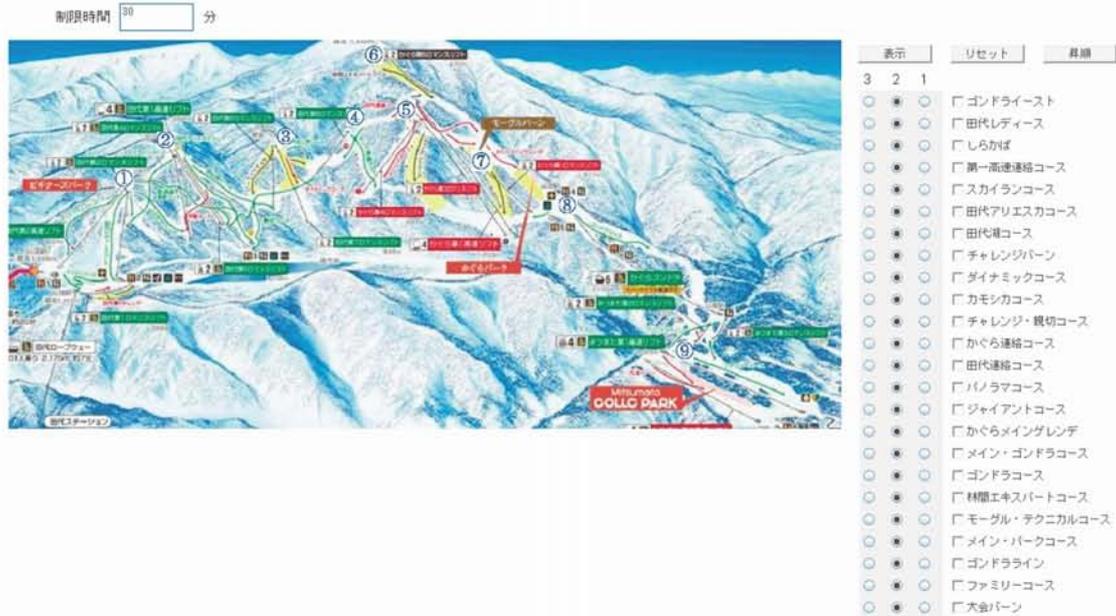


図3 アプレット起動画面

7. 検索結果

初めに、図4には制約時間を20分に設定し、各滑走コースの選択が終わった実行結果を示した。先ほどの起動画面とは異なり、コースごとの希望度が画面右に示されている。希望度は3段階のラジオボタンで定義されたものであり、ユーザーは各グレンデコースの希望度をボタンをタッチして入力している。本システムの起点はかぐら山頂駅となっている。画面下には、滑走経路並びに滑

走の所要時間、希望度の合計が示されている。

本システムでは、山頂駅を出発点として田代第二リフトは無条件で選択されるため、希望度に関係なく第1番目に表示されている。従ってルートを選択肢は2行目以降に示される。候補はここに示される8ルートとなるが、実際は繰り返しが許されるため、さらに多くの組み合わせを許容している。ここでは最大15行までのルート表示に留めているが、その範囲であれば画面スクロールで候補をさらに閲覧することができる。



図4 制約時間1000秒における実行結果



