

多数の観光候補地から効率良い観光スケジュールを自動的に作成・提案する システム P-Tour の Google Maps を利用した設計と実装

栗山 恭嘉¹ 花野 博司¹ 澤 悠太¹ 村田 佳洋^{2,*}

柴田 直樹^{3,*} 安本 慶一^{1,*} 伊藤 実¹ 藤原 礼征⁴

¹ 奈良先端科学技術大学院大学, ² 広島市立大学, ³ 滋賀大学,

⁴ (株) ソリューション・クルー, * 独立行政法人科学技術振興機構, CREST

{hisaka-k,hiroshi-h,yuta-s,yoshi-m,n-sibata,yasumoto,ito}@is.naist.jp

著者らのグループでは、多数の観光候補地とそれぞれに対する重要度、時間制約を指定するだけで、移動距離や時間制約、重要度を考慮した、複数日に跨る旅程を自動的に算出し、また、ユーザの経路案内、発着案内を行う観光用パーソナルナビゲーションシステム P-Tour を提案、開発してきた。今回、P-Tour の観光スケジュール自動作成機能を Web 上で利用できるようにするため、観光候補地の選択、算出されたスケジュールにおける巡回経路の表示などの機能を、新たに Google Maps を用いるよう再設計し実装した。また、宿泊地自動推薦機能を設計し、組み込んだ。本稿では、Web 版 P-Tour の設計および実装の詳細、および、使用感について報告する。

Design and Implementation of P-Tour's Automatic Tour Schedule Construction Function using Google Maps

Hisaka Kuriyama¹ Hiroshi Hanano¹ Yuta Sawa¹ Yoshihiro Murata^{2,*}

Naoki Shibata^{3,*†} Keiichi Yasumoto^{1,*} Minoru Ito¹ Hiroyuki Fujiwara⁴

¹ Nara Institute of Science and Technology, ² Hiroshima City University, ³ Shiga University,

⁴ Solution Crew, Inc., * Japan Science and Technology Agency, CREST

P-Tour is a personal navigation system for sightseeing which automatically plans a tour schedule from information including time constraint and importance of each sightseeing spot input by users. It also has a function to navigate the user according to the schedule through a portable computing device. In this paper, we design and implement a web interface utilizing Google Maps API for P-Tour's automatic schedule planning. With this interface, each user can intuitively select his/her favorite sightseeing spots and see the computed tour schedule through web browser. We also design and implement a function to automatically select a recommended accommodation. We report details of our design and implementation of P-Tour on Google Maps, and show better usability of our new interface.

1 はじめに

カーナビゲーションシステムは既に出荷台数が 2700 万台を超え、もはやエアコンのように有って当然の装備の一つとして普及している。一方、携帯電話を通して、個人の経路案内を行うパーソナルナビゲーションシステムも既にサービスが提供されている。このような、現在製品化されているナビゲーションシステムが持つ経路探索機能は、二地点間の経路探索、あるいはその組合せに限定されている。観光のように複数の目的地を巡回する場合のスケジュールの立案は、候補となる経路の組合せの数が膨大になるため、ある基準に対して最良のスケジュールを短い時間で算出するのは困難であることが知られている。我々は、このような、複数の目的地を巡回する用途において、時間制約及び重要度、移動距離を考慮して効率よく巡回するスケジュールを自動作成する遺伝的アルゴリズムを用いた経路探索エンジン进行を考案し、携帯端末からクライアントサーバ方式で経路探索エンジン呼び出すことによりユーザを案内するパーソナルナビゲーションシステム P-Tour を提案してき

た [1, 2, 3, 4].

P-Tour では、ユーザの入力した、出発地と出発時刻、帰着地と帰着時刻、複数の観光候補地と各地への立ち寄り希望度と時間制約（到着時間帯や滞在時間など）等の情報から、複数日程にまたがる観光スケジュールの自動算出を行うことが可能である。複数日程にわたる観光においては、観光候補地の数が多くなるために経路を算出するための時間が長くなるという問題の他に、ユーザに提示する観光候補地の数、および地図の広さが大きくなり、必要な情報の入力が煩雑になってしまうという別の問題が存在する。専門知識のないユーザでも直感的に必要な情報を入力できるためには、入力、および出力されたスケジュール確認のための高機能な GUI が必要である。また、このようなシステムの有用性を評価し、問題点を洗い出す上で、多数のユーザによる評価が重要である。

本稿では、Google Maps API を利用した、P-Tour を Web 上で使用可能にするためのインタフェースの設計・実装に関する詳細について報告する。本インタフェースは、著

者らが文献 [3] で提案したシステムに基づいている。本稿では、このインタフェースの設計と実装の詳細と使用感について述べる。

2 観光用ナビゲーションシステム P-Tour の概要

我々の研究グループでは、これまで観光のためのパーソナルナビゲーションシステム “P-Tour” [1] を提案してきた。図 1 に示すように、P-Tour は、サーバ・クライアントシステムで構成され、ユーザが出発地と出発時刻、帰着地と帰着時刻、複数の観光候補地と各地への立ち寄り希望度と時間制約（到着時間帯や滞在時間など）を入力すると、システムは、制限時間内で巡回可能かつ最も満足度が高くなるような観光地の巡回経路と各観光地への到着・出発予定時刻を含むスケジュールを算出し、これをユーザに提示する。スケジュールは、遺伝的アルゴリズムを用いて算出される。また P-Tour は決定したスケジュールに従い、GPS 機能を備えた携帯端末を介して、ユーザをナビゲーションする機能も有する。

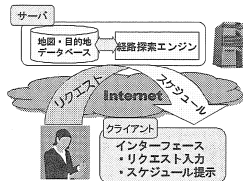


図 1: P-Tour の概要

我々はこれまでに、P-Tour に対するいくつかの拡張機能を提案してきた。

観光においては、たくさん観光地をまわりたい、しかしできるだけ安価に済ませたい、というような、同時に満たすことが難しい多くの相反する要求が存在する。そこで、個人観光のためのスケジュール立案問題を多目的最適化問題として定式化し、多目的 GA を用いて、それぞれの要求に対する折衷案である準パレート最適解を得る手法を提案した [2]。また、訪れたい観光地が少しずつ異なる複数のメンバがグループで観光する際に、メンバそれぞれの希望を満たしつつ、希望の合致する部分を共有するようなスケジュールを算出することは難しい。そのため、この問題を複数スケジュールの同時立案問題として定式化し、それを実行時間で計算する GA を用いた近似アルゴリズムを提案した [4]。

一方、複数日に渡った経路案内を行う場合、スケジュールに含まれる観光地の組み合わせが多く、また前後の観光を考慮しつつ宿泊地を選定する必要があるため、実行時間でユーザの希望に沿ったスケジュールを算出するには工夫を要する。この問題に対し、我々は、複数日にわたる観光のためのスケジュール作成手法を考案した [3]。この手法では、観光対象エリアを幾つかの部分エリアに分割し、エリア境界の越境回数を制限することにより組み合わせ数を減少させている。最適化手法として用いている GA において、宿泊地を符号化する際の遺伝子座を限定することにより、日ごとに宿泊地をただ一つ含む経路を作成することができる。この手法により、5泊6日程度の比較的長期にわたる観光スケジュールを実行時間で算出することが可能になった。

複数日程にわたる観光においては、観光候補地の数が多くなるために経路を算出するための時間が長くなるという問題の他に、ユーザに提示する観光候補地の数、および地図の広さが大きくなり、必要な情報の入力が煩雑になって

しまうという別の問題が存在する。従来の P-Tour のユーザインタフェースは、図 2 に示すように、ユーザが情報を入力する際、web ブラウザ上のフォームに直接数値を打ち込む仕様になっていた。しかし、専門知識のないユーザでも直感的に必要な情報を入力できるようにするには、入力、および出力されたスケジュール確認のための高機能な GUI が必要である。本稿では、Google Maps API を利用した、P-Tour のためのインタフェースを設計、実装する。



図 2: 従来の P-Tour のユーザインタフェース

3 システムの全体設計

従来の P-Tour を、一般のユーザが Web 経由でスムーズに使用するには、下記のインタフェース上の問題があると思われる。(1) 目的地等の情報を数値で入力するようになっているため、直感的でなく、システムの内部知識を必要とすること。(2) 多数のユーザの同時利用が不可能であること。これらの問題を解決するために、Google Maps API [5] を用いた、直感的な操作が可能な Web インタフェースの設計を行う。

本章では、まず Google が提供する地図作成 API である Google Maps API について述べ、さらに、Google Maps API を用いた Web インタフェースの設計について順に述べる。

3.1 Google Maps API

Google Maps は Google が提供する地図検索サービスである。地図や衛星写真の表示が可能であり、マウスを用いて地図や衛星写真をスムーズにスクロールしたり、ズームしたりすることが可能である。Google Maps API は、これらの機能を Web アプリケーションで使えるように公開された、JavaScript 用の API である。

Google Maps API では、地点を登録しアイコンで場所を示したり、吹き出しで情報を示す機能が提供されている。また、地図上に、ユーザが作成したプログラムの指示どおりに線を引くことが可能であり、この機能を使用してユーザに経路や境界を示すことができる。

3.2 画面デザイン

インタフェースは、図 3 に示すように、6つの画面から構成されるよう設計した。以下、設計したインタフェースの画面ごとの詳細を述べる。

出発/到着地の選択画面

この画面では、スケジュールの日程や範囲を決めるため、初日の出発地点と最終日の到着地点、各日時を設定する。これらの位置は、Google Maps を用いて地図上で選択可能で

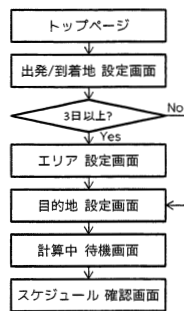


図 3: 画面遷移

ある。従来の、登録された地点の一覧のセレクトボックスから選択する方法では、登録数が増えたと地点を探し選択するのが困難になるが、本インタフェースにより容易に探し出すことが可能である。

この画面では、地図にあらかじめ登録された地点のアイコンが表示され、アイコンをクリックすると地点の情報が吹き出しで表示される(図 4)。設定したい地点の「選択」ボタンによって出発・到着地を設定する。表示範囲内に希望する地点が無い場合、地図をドラッグしたり範囲を広げることによって表示範囲を変えられ、表示範囲に合わせて地点表示が変化する。

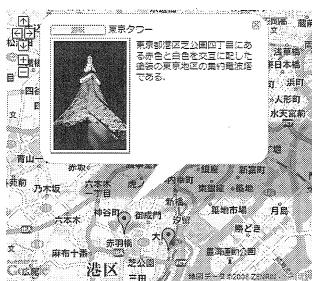


図 4: 地点の入力

エリアの設定画面

この画面では、観光対象エリアを設定する。また、出発・到着地の選択画面で設定された地点の確認地図を表示する。ただし、日程が2日以下の場合出発・到着地によって観光エリアが決まるためこの画面は省略され目的地の設定画面に遷移する。この画面では、出発・到着地で選択された地点のエリアと日程から、巡回可能なエリアのパターンの一覧を表示し、ユーザはそこから希望するパターンを選択する。日程が長くなると、パターン数が多くなりユーザが選択しにくくなるため、パターンを絞込むための機能を有する。

目的地の設定画面

この画面では、ユーザが周りたい目的地とその優先度や日時を設定する。また、各日程の宿泊地の設定も行う。宿泊地は、システムにより自動的に決定することも可能である。この画面では、地図にあらかじめ登録された候補地が表示され、アイコンをクリックすると地点の情報が吹き出しで表示される。ユーザが訪問を希望する地点を選択すると、その地点が目的地テーブルに追加される。目的地テーブルは

GUIの要素であり、各目的地の優先度や到着日時、希望滞在時間などが設定できる。

位置を確認して選択できる様にするために、Google Mapsで地点を選択した上で、選択した目的地に対して日時と優先度を設定することが可能である。従来のインタフェースでは、登録された目的地一覧から、行くか行かないかをチェックした上で、日時や優先度の設定をする必要があった。この方法では、登録数が増えたと地点を探し選択するのが困難になる問題があった。

計算中の待機画面

この画面は、これまでの画面で設定された情報をスケジュール計算エンジンに送信した後、計算結果が出るまで表示される。計算が完了するまでに、条件によっては数十秒かかる場合がある。そこでユーザを退屈させないために、各目的地の情報をスライド表示する。

スケジュール確認画面

スケジュール計算エンジンで計算された結果を視覚的に分かりやすく表示するために、Google Mapsを用いて地図に目的地のアイコン、巡回経路を日程ごとに色分けして表示する。

従来のインタフェースでは、白地図上に巡回経路を表示し、ユーザに提示していた。従来のインタフェースでは、地図を拡大したり、周囲の情報を知るための機能が提供されていなかった。今回作成したインタフェースでは、Google Mapsの機能を用いることにより、スムーズに拡大縮小移動ができ、また、Google Mapsで提供される周囲の情報(交通機関、グルメ施設等)が参照可能になった。

巡回経路は国土地理院の道路情報から作成されており、Google Mapsの道路上に合わせて表示され、車や徒歩で巡回可能な経路となっている。地図の下部にはスケジュールテーブルが表示され、目的地を周る細かい日時や滞在時間などが表示される。

4 実装の詳細

本章では、3章で述べたWebインタフェースの実装の詳細について述べる。今回のインタフェースの開発においては、従来のバージョンをもとにして開発を行ったが、今回開発に関わったメンバーは、従来のバージョンの作成者とは異なる。開発効率等を考慮し、従来のバージョンの各モジュールのAPIを新たに定義する方法を採用した。

図5に、全体のモジュール構成を示す。システムは大きくP-Tourサーバ、ユーザのブラウザ、Google Mapsに分けられる。P-Tourサーバでは、WebサーバApacheが動作しており、ユーザはWebブラウザを使用して、スケジュール作成のための入力等を行う。ユーザのブラウザ操作によって発生したイベントはJavaScriptで捕捉し、地図に関する操作であればGoogle Maps APIに処理を要求する。各ページの設定は、セッションにまとめてスケジュール計算エンジンに計算要求を送出する。計算結果はXHTMLやJavaScriptに成型して出力し、ユーザのWebブラウザに提示する。ユーザの操作を分かりやすくするために、JavaScriptはDOM(Document Object Model)の操作も行う。例えばユーザが目的地を選択した時に、追加された目的地をハイライト表示させる。また、現在地図に表示されている地点のリストをリアルタイムに更新するためにも使用している。これらの処理のため、今回は、Ruby on Rails [6]を用いて開発を行った。

以下、各モジュールについて順に述べる。

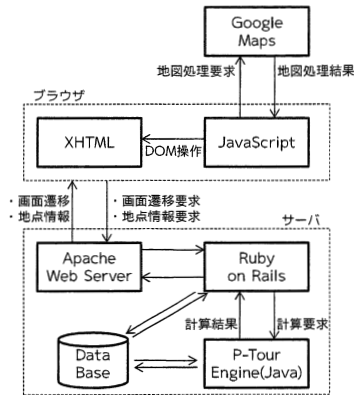


図 5: システム全体のモジュール構成

4.1 経路探索エンジン

経路探索エンジンは、文献 [3] で提案したものをもとに、幾つかの拡張を行っている。最も大きな変更点は、もとのシステムでは、エリア複数日に渡る同一エリアの観光には対応しておらず、一度観光したエリアは別の日程において再度訪れることが出来なかったが、今回の拡張では、この制約を撤廃した。また、従来システムにウェブ処理部およびデータベースを新たに追加した。図 6 に、経路探索エンジンの API を構成する主なクラスを示す。

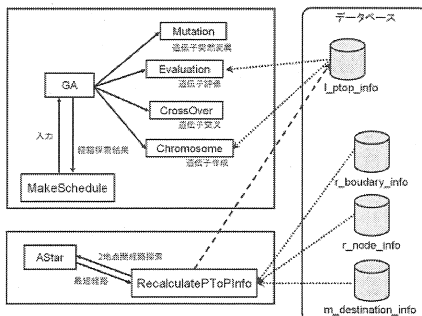


図 6: 経路探索エンジンのモジュール構成

図 6 中の MakeSchedule クラスは、ウェブインタフェース部から渡される、ユーザの入力した情報を基に、ユーザ要求を満たすような経路を探索する API である。従来システムにおける遺伝的アルゴリズムおよびユーザ入力処理するクラスを基にしている。

図 6 中の RecalculatePToPInfo クラスは、新たに目的地が追加された際に、既存の全目的地候補との間の最短経路を計算するための API である。従来システムにおける A* 探索を実行するクラスを基にしている。

4.2 インタフェース仕様

以下、新しく設計したウェブインタフェースと、探索エンジン間のインタフェースについて述べる。従来のウェブインタフェースでは、表 1(a) の形式で記述されたテキストファイルを用いて、ウェブインタフェースと、探索エンジン間で、ユーザから入力された情報の受け渡しを行っていた。

今回実装したシステムでは、上記のデータ形式は変えずに、データの受け渡し方法をテキストファイルを介する方式から、パイプを用いて通信する方法に変更した。

従来システムでは、算出したスケジュールを表示するために、国土地理院発行の数値地図電子閲覧用 xml ファイルを用いて、システム独自のウィンドウに地図および探索した経路を描写する方式をとっていた。今回、Google Maps API を使用して探索した経路を描写するために、表 1(b) で定義するデータ形式をパイプを用いてウェブ処理部と通信する方法を探ることとした。

4.3 データベース

従来システムでは、入力、目的地情報および道路情報を格納するためにテキストファイルを使用していた。このため、テキストファイルの読み込みのための計算コストによるオーバーヘッドがあった。今回開発したシステムでは、データ管理にデータベースを用いることで、このオーバーヘッドを軽減した。このことによる、スケジュール計算コストの軽減は約 60% に上る。データベースシステムとして、PostgreSQL を使用した。

以下、データベースの詳細について順に述べる。

目的地間情報を格納するテーブル

遺伝的アルゴリズムを使用して経路を探索する際のコストの評価のために、異なる 2 つの目的地の組合せ全てに対して、レコードを保持する。経路探索の際および、その結果得られた目的地間の経路情報を得るために必要になる。レコードの要素を以下に示す。

column name	type
ID	整数
目的地 1 の目的地 ID	整数
目的地 1 の経度	浮動小数点
目的地 1 の緯度	浮動小数点
目的地 1 のエリア	整数
目的地 1 の属性	整数
目的地 2 の目的地 ID	整数
目的地 2 の経度	浮動小数点
目的地 2 の緯度	浮動小数点
目的地 2 のエリア	整数
目的地 2 の属性	整数
距離	浮動小数点
経路情報	文字列

各目的地に関する情報を格納するテーブル

Google Maps にて目的地を選択するため、および新たに目的地が追加された際に目的地間情報を計算するために参照する。レコードの要素を以下に示す。

ID	整数
エリア	整数
属性	整数
名称	文字列
経度	浮動小数点
緯度	浮動小数点
営業開始時間	時刻型
営業終了時間	時刻型
料金	文字列
住所	文字列
電話番号	文字列
URL	文字列
画像ファイルパス	文字列
公開/非公開フラグ	論理値
計算済みフラグ	論理値

表 1: ウェブインタフェースと経路探索エンジン間で交換されるテキストファイル形式
(a) 従来の形式

出発地：出発時刻
訪れたい観光地 1：到着時刻
：
訪れたい観光地 m：到着時刻
1 日目の宿泊地：到着時刻：出発時刻
：
2 日目の宿泊地：到着時刻：出発時刻
：
1 日目の宿泊地：到着時刻：出発時刻：滞在時間
：
到着地：到着時刻

(b) 新しい形式

出発地：出発時刻
訪れる観光地 1：到着時刻：出発時刻：滞在時間
2 地点間の経路情報 < 緯度, 経度 >
訪れる観光地 2：到着時刻：出発時刻：滞在時間
：
訪れる観光地 m：到着時刻：出発時刻：滞在時間
1 日目の宿泊地：到着時刻：出発時刻：滞在時間
：
1 日目の宿泊地：到着時刻：出発時刻：滞在時間
：
到着地：到着時刻：滞在時間

新たに目的地が追加された際には、このテーブルより ID・エリア・属性・経度・緯度を取得し、A*探索を用いて新たな目的地間情報を得る。

5 使用例

本章では、今回実装したシステムを実際に使用した際の使用感を報告する。例として、東北地方、特に秋田・青森の 2 泊 3 日旅行について計画する場合について述べる。

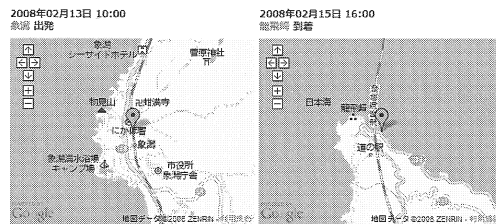


図 7: 出発地の設定

まず、出発・到着地の設定画面で出発・到着地と日時を設定する。図 7 に示すように、地図からアイコンをクリックし、希望の地点を選択する。日時は上部のセレクトボックスで選択する。

今回は出発を 2 月 13 日 10 時に「象潟」、到着を 2 月 15 日 16 時に「龍飛崎」とした 2 泊 3 日旅行を計画する。

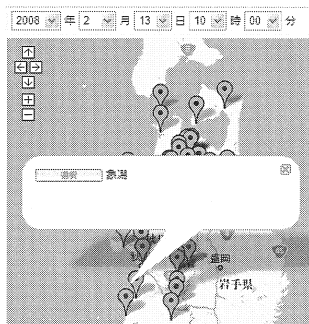


図 8: 出発・到着地と日時の確認

次に、エリアの設定画面で図 8 に示すように出発・到着地と日時を確認し、図 9 に示すようにエリア一覧からエリ



図 9: エリアの設定

アを選択を行う。今回はエリアを「秋田→秋田→青森」とした。



図 10: 目的地の選択

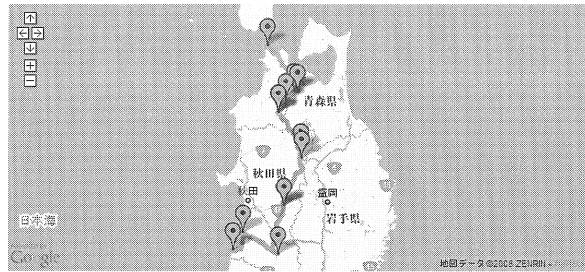
観光地	経路地	住所	優先度	希望到着時刻	希望滞在時間
龍飛崎	川原毛地区/川原毛地区郡	住所27	3	2008/02/15 14:00	30分程度
龍飛崎	本荘公園	住所30	1	---	30分程度
龍飛崎	六郎滝水郷	住所29	1	---	30分程度

図 11: 目的地テーブル

次の目的地の設定画面では、出発・到着地設定画面と同様に、図 10 に示すように地図からアイコンをクリックし、希望の地点を選択する。選択された目的地は、図 11 のように目的地テーブルに追加される。ここで日程や優先度、滞在時間など細かい条件が設定できる。目的地を複数選びたい場合は同様に各目的地を選択し設定を行う。

クリックされたアイコンが宿泊地の場合、図 12 のように泊まる日程も選択する。選択された宿泊地は図 13 のように、宿泊地テーブルの選択された日程の欄に設定される。ここで宿泊地への到着時間や翌日の出発時間を設定できる。宿泊地が決まっていない場合は宿泊地を選択せず「おまかせ」、2 日目を「弘前プリンスホテル」に設定した。

以上の設定を終えて、「スケジュール作成」ボタンをクリッ



日程	出発	目的地	到着時刻	滞在時間	出発時刻	帰着
1日目	10:00	余湯	-	-	10:00	角館プラザホテル
		本荘公園	10:48	30	11:18	
		川原毛地蔵／川原毛地蔵菩薩	12:38	112	14:30	
		角館プラザホテル	15:56	-	-	
2日目	12:00	角館プラザホテル	-	-	12:00	私前プリンスホテル
		玉川温泉	13:48	30	14:18	
		大湯環状列石	14:31	30	15:01	
		私前プリンスホテル	16:17	-	-	
		私前プリンスホテル	-	-	12:00	
3日目	12:00	湯岡城址	12:24	30	12:54	龍飛崎
		青森温泉	13:22	30	13:52	
		三内丸山遺跡	13:59	30	14:29	
		龍飛崎	16:04	-	-	
		龍飛崎	-	-	-	

図 14: 得られたスケジュール

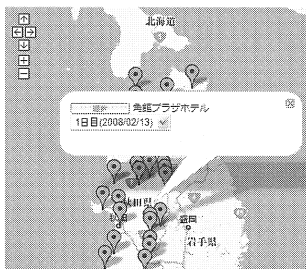


図 12: 宿泊地の選択

日程	宿泊地	希望到着時刻	希望出発時刻(翌日)	おまかせ
(1日目) 2008/02/13	角館プラザホテル	17時頃	12時頃	「希望到着時刻」
(2日目) 2008/02/14	私前プリンスホテル	17時頃	12時頃	「希望到着時刻」

図 13: 宿泊地テーブル

クすると、図 14 のように数十秒後に計算結果が表示される。スケジュール確認画面では地図で目的地の場所や経路が確認でき、下のスケジュールテーブルで詳細な行程が確認できる。「象潟」を出発し 2 泊 3 日で「龍飛崎」に到着するという、今回設定した通りのスケジュールが作成された。

6 まとめ

本稿では、著者らがこれまで提案してきた観光用パーソナルナビゲーションシステム P-Tour を Web 上で利用可能にするための、Google Maps API を利用したインタフェースの設計と実装の詳細について報告した。また、実装したシステムの使用感について報告した。今後、一般ユーザによる

使用感のアンケート調査等を行った上で、問題点を洗い出し、システムの改善につなげていくことを予定している。

謝辞

本研究の一部は、独立行政法人科学技術振興機構 CREST と奈良先端科学技術大学院大学支援財団新産業創出支援事業の支援を受けて行った。また、(株) イージーネットの久保人氏には、本システムを開発する上で多大なるご支援を頂いた。ここに記して謝意を表す。

参考文献

- [1] 丸山 敦史, 柴田 直樹, 村田 佳洋, 安本 慶一, 伊藤 実: 観光スケジュール作成支援とスケジュールに沿った経路案内を行うパーソナルナビゲーションシステム, 情報処理学会論文誌, Vol. 45, No. 12, pp. 2678 - 2687 (2004).
- [2] Shiraishi, T., Nagata, M., Shibata, N., Murata, Y., Yasumoto, K. and Ito, M.: A Personal Navigation System with Functions to Compose Tour Schedules based on Multiple Conflicting Criteria, IPSJ Digital Courier, Vol.1, pp. 528-536 (2005).
- [3] 木下降正, 永田宗伸, 村田佳洋, 柴田直樹, 安本慶一, 伊藤 実: 複数日にわたる観光のためのパーソナルナビゲーションシステム, 情報処理学会論文誌, Vol.47, No.12, pp. 3179-3187 (2006).
- [4] 永田 宗伸, 村田 佳洋, 柴田 直樹, 安本 慶一, 伊藤 実: 分離・合流を伴うグループ観光スケジュール作成機能の提案, 情報処理学会論文誌 数理モデル化と応用 (TOM17), Vol.48, No.SIG56, pp.23-31 (2007).
- [5] Google マップ API, <http://code.google.com/intl/ja/apis/maps/>
- [6] Ruby on Rails, <http://www.rubyonrails.org/>