Detectarea preferinţelor turiştilor pentru recomandarea de orare de vizitare personalizate

Radu Răzvan Slăvescu
Universitatea Tehnică din Cluj-Napoca
Bariţiu 28, RO-400027 Cluj-Napoca, România
Radu.Razvan.Slavescu@cs.utcluj.ro

Adrian Coapşi
Universitatea Tehnică din Cluj-Napoca
Bariţiu 28, RO-400027 Cluj-Napoca, România
adrian.coapsi@gmail.com

REZUMAT

Articolul explorează posibilitatea de a estima timpul petrecut de un turist într-un obiectiv turistic şi de a construi un orar de vizitare a obiectivelor dorite pe baza acestor estimări. Pentru aceasta, s-a implementat un prototip sub forma unei aplicaţii Android formate din mai multe module care colaborează pentru atingerea acestui scop. Un modul înregistrează periodic poziţia utilizatorului şi momentul în care acesta intră, respectiv părăseşte un anumit obiectiv. Al doilea estimează timpul necesar unei vizite pe baza timpului recomandat pentru acel obiectiv şi a timpilor petrecuţi în obiective similare. Un modul de rezolvare a problemelor de satisfacere a constrângerilor generează apoi un orar pentru vizită, folosind aceste estimări. Eficienţa şi vibilitatea soluţiei sunt discutate şi se propun soluţii de îmbunătăţire a performanţelor.

Cuvinte cheie

Planificator turistic, Android, probleme de satisfacere a constrângerilor

Clasificare ACM

H5.m Miscellaneous

INTRODUCERE

Din ce în ce mai des, se doreşte planificarea vacanţelor turiştilor în funcţie de propriile preferinţe şi propriul istoric al vizitatorilor în locul unor trasee standardizate.

Prezentul articol prezintă o soluţie pentru construirea unui orar de vizitare a unor obiective turistice care ţine cont de timpul pe care posesorul l-a petrecut în obiective similare, precum şi de timpul de vizitare recomandat în ghidurile de călătorie.. Pentru aceasta, s-a implementat un prototip sub forma unei aplicaţii Android formate din mai multe module care colaborează pentru atingerea acestui scop. Acesta înregistrează obiectivele vizitate, tipul lor şi timpul petrecut acolo, estimează timpii de vizitare a unor obiective similare ţinând cont de aceste informaţii şi de timpul recomandat şi planifică seria de vizite tinând cont de aceste estimări şi de orarul oficial al obiectivelor care vor fi vizitate. Ultimul scop se modelează sub forma unei probleme de satisfacere a constrângerilor abordată folosind o variantă de algoritm de tip Stochastic Local Search cu random walk şi random restart.

Spre deosebire de soluţia propusă în [4], sistemul nostru beneficiază de colectarea de date folosind GPS. Articolul [5] propune o metodă de personalizare a recomandărilor, dar nu abordează problema estimării timpului de vizitare. Aceeaşi limitare o are şi sistemul descris în [1]. Aplicaţia GraniteNights [2] este cea mai asemănătoare cu a noastră, aceasta folosind însă o altă abordare pentru generarea orarului şi nepreocupându-se de estimarea personalizată a timpului de vizită.

Restul articolului prezintă un prototip modular dezvoltat pentru rezolvarea acestor probleme, discută principalele sale caracteristici şi apoi prezintă concluziile şi posibilele devoltări ulterioare.

UN SISTEM PENTRU ORARE PERSONALIZATE

Sistemul generator de sugestii de planificare este format din trei module. Primul dintre acestea înregistrează periodic punctele vizitate de utilizator, pentru a calcula timpul petrecut într-un anumit obiectiv turistic. Al doilea modul estimează timpul ncesar vizitării unui nou obiectiv, prin analogie cu timpul petrecut în obiective similare. Cel de-al treilea modul foloseşte timpii estimaţi spre a genera un orar de vizitare a unui set de obiective pe care turistul doreşte să le vadă. O privire de ansambu a sistemului este prezentată în Figura 1.

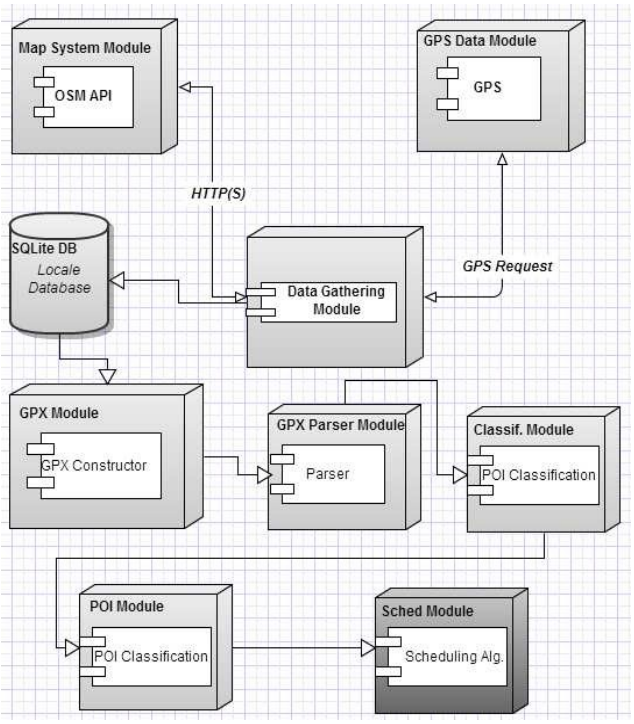

Figura 1. Arhitectura sistemului

Colectarea datelor GPS

Un modul înregistrează periodic, la un interval configurabil, poziţia turistului. Se creează un fişier .gpx (GPS Exchange Format) ce conţine informaţii despre traseele parcurse de către utilizator. Comportamentul general este rezumat în Algoritmul 1.

Vom exemplifica în continuare pe cazul unor muzee, dar aplicaţia oferă posibilitatea de a nota o varietate de categorii de obiective. La intrarea într-un anumit muzeu, utilizatorul va apăsa butonul *check-in,* apoi va alege tipul obiectivului (Fig. 2). La ieşire, va apăsa butonul *check-out.* Pe baza acestor date, se va stabili timpul petrecut în muzeu.

Modulul permite folosirea OpenStreetMap pentru afişarea traseului parcurs de către utilizator pe hartă (Fig. 3).

Procedure CreateGPX

```
repeat
  if NOT haveGPSSignal then TurnGPSOn
  else
    if CheckInButton then
      lat = GetLatitude
      long = GetLongitude
      ti = GetTime
      wp = Createwaypoint(lat, long, ti)
      SaveIntoDataBase(wp)
      if DrawOption then DrawOnMap
      endif
    endif
    if CheckOutButton then
      to = GetTime
      UpdateDataBase(wp, to)
    endif
    if GenerateGPXButton then
      for all GPStrk do
        GenerateGPX
        SaveOnSDCard
      endfor
    endif
  endif
until termination
```

Algoritmul 1. Crearea GPX

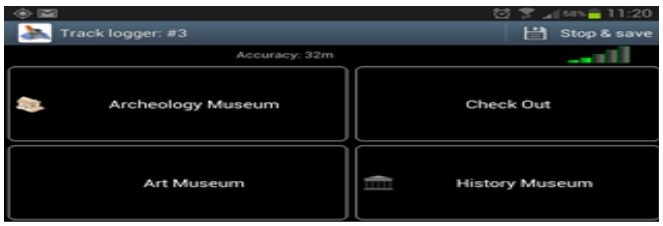

Figura 2. Interfaţa pentru notarea tipului obiectivului

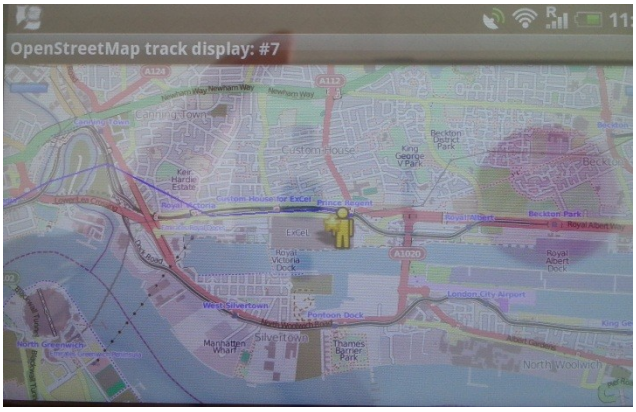

Figura 3.Vizualizarea traiectoriei utilizatorului

Estimarea timpului de vizitare

Al doilea modul foloseşte un parser care extrage din fişierul .gpx punctele de interes înregistrare de utilizator, calculează timpul petrecut în fiecare obiectiv şi returnează o lista cu punctele de interes si timpul petrecut în fiecare din ele.

Pentru estimarea timpului de vizitare a uni obiectiv, se foloseşte o raţionare bazată pe instanţă. Obiectivele vor fi clasificate după mai multe criterii, într-o structură de tip arbore (Fig. 4). De exemplu, in cazul unui muzeu se va lua in considerare domeniul (istorie, arheologie, artă) şi mărimea sa (mică, medie, mare) conform numărului de artefacte expuse. Pentru un muzeu al cărui timp de vizitare recomandat îl cunoaştem şi al cărui timp de vizitare efectiv dorim să îl estimăm, se va căuta prima dată în ierarhia de clase toate muzeele similare ca mărime şi tip cu acesta care au fost vizitate deja, se va stabili pentru fiecare raportul dintre timpul efectiv petrecut acolo şi cel recomandat, apoi media acestor rapoarte se va propaga la nivelul clasei din care acestea fac parte. Timpul estimat pentru noul muzeu va fi egal cu produsul dintre timpul recomandat şi coeficientul clasei. Procedura e detaliată în Algoritmul 3 şi este exemplificată în Fig. 4.

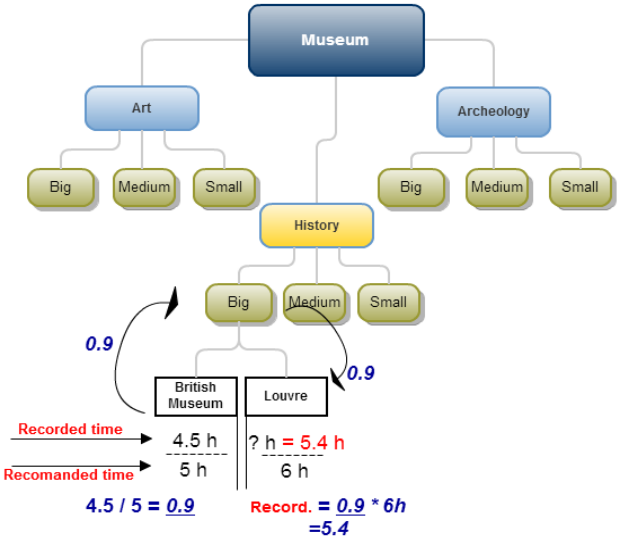

Figura 4. Exemplu de estimare a timpului de vizitare

```
Procedure EstimateTime
  Require: MusList, MusTypes, MusSize

  GetMusFromGPX
  for all museums do
    GetNamesAndType
    GetTime
    if Type = Mus.GetType then
      AddMusToList
    endif
  end for
  for all museumsFromTypeList do
    AddMusToListSize
    GetTimeSpent
  end for
  for all museumsFromSizeList do
    if hasTimeSpent then
      Cf = TimeSpent/TimeRecommended
      SendCfToSuperClass
    else
      Cf = UpdateCf
    endif
  end for
  return Cf *TimeRecommended

Function UpdateCf
  Require: OldCf: existing coefficient
           NewCf: the newly added coefficient
           NrObj: number of objectives in a class

  if not(FirstV isited(Mus)) then
    Cf = (OldCf *(NrObj−1) + NewCf) / NrObj
  else
    Cf = TimeSpent/TimeRecomended
  endif
  return Cf
```

Algoritmul 2. Estimarea timpului de vizitare

Generarea orarului

Al treilea modul responsabil de planificarea unui traseu, bazat pe timpii personalizaţi din modulul doi, rezolvă această problemă prin modelarea sa ca o problemă de satisfacere a constrângerilor (PSC) şi utilizarea metodelor de *Stochastic* Local *Search* cu *Random walk* (mutare într-un vecin ales la întâmplare din spaţiul soluţiilor) şi cu *Random restart* (reasignare periodică de valori aleatoare tuturor variabilelor) [3,6] pentru generarea unei soluţii. Testele preliminare arată că spaţiul soluţiilor pentru acest tip de probleme este suficient de dens pentru ca o soluţie să fie găsită întotdeauna. Algoritmul folosit este prezentat ca Algoritmul 3.

```
Procedure GenerateTimeTable
  Require: csp: o problema de tip CSP
           max_steps: numar de incercari
           max_tries: numar de reporniri
           p: probabilitatea de random walk

  for t=1 to max_tries do
    current = o asignare aleatoare, completa
              de valori pentru csp
    for i=1 to max_steps do
      if current e solutie a csp then
        return current
      endif
      var = o variabila cu conflicte, aleasa aleator din
            VARIABLES[csp]
      value = valoarea v pentru var care minimizeaza
              CONFLICTS(var, v, current, csp)
      cu probabilitatea p executa
        value = o valoare aleatoare pentru var
      var = value in current
    end for
  end for
  return failure
```

Algoritmul 3. Generarea unui orar de vizitare

Problema de satisfacere a constrângerilor consideră ca variabilă o pereche de tip muzeu-interval de vizitare. Se presupune că orarul muzeului este cunoscut. De exemplu, dacă muzeul M poate fi vizitat doar lunea între 9 şi 13 şi timpul estimate de vizitare este 3 ore, vom avea 2 variabile: M_Lu_9_12 şi M_Lu_10_13. Putem presupune, fără a restrânge generalitatea, că lucrăm cu cuante de timp de 1 oră, adică intrarea în muzeu se face întotdeauna la o oră fixă şi durează un număr întreg de ore.

Setul de valori posibile pentru fiecare variabilă este mulţimea {T,F} unde M_Lu_9_12 înseamnă că muzeul M este programat spre a fi vizitat lunea de la 9 la 12.

Constrăngerile implicate sunt de două tipuri. Primul tip cere ca două variabile ce reprezintă muzee diferite să nu aibă ambele valoarea T daca intervalele de timp corespunzătoare se suprapun. De exemplu, M_Lu_9_12 şi N_Lu_11_13 nu pot avea ambele valoarea T. Aceasta permite extinederea constrângerii astfel încât să modeleze şi timpul necesar deplasării între două muzee, de exemplu dacă timpul necesar deplasării de la M la N este 1 oră, atunci M_Lu_9_12 şi N_Lu_12_14 devine şi ea o combinaţie interzisă.

Al doilea tip de constrângeri cere ca, între toate variabilele care corespund unui anumit muzeu, exact una să aibă valoarea T. Aceasta constrângere ar putea fi relaxată astfel încât să permitem un număr de valori T corespunzătoare unui muzeu oarecare să fie, să spunem, maximum 3, ceea ce s-ar traduce prin a permite vizitarea unui muzeu în etape (cu condiţia suplimentară ca suma lungimilor intervalelor acorespunzătoare să fie egală cu timpul estimat). Algoritmul de rezolvare a PSC este ulterior folosit pentru generarea orarului.

DISCUŢII

Prototipul dezvoltat permite un număr de experimente şi dezvoltări. O primă problemă ar fi dacă estimarea timpului de vizitare se poate face pe baza similarităţii muzeelor şi care ar fi atributele relevante pentru a stabili muzeele din aceeaşi clasă. O divizare prea fină a claselor ar putea prezenta riscul de a avea prea puţine instanţe concrete pentru fiecare (muzee efectiv vizitate). Sistemul este însă suficient de flexibil pentru a permite extiderea setului de atribute.

O a doua problemă ar fi eliminarea operaţiilor manual de *check-in/check-out*. Salvarea coordonatelor GPS permite detectarea automată a momentului în care utilizatorul intră/iese dintr-un muzeu, precum şi detectarea unor puncte de interes pentru acesta, conform abordării din [7,8].

Preluarea datelor GPS ridică problema capacităţii bateriei; testele efectuate pe un telefon cu baterie de 1400 mA şi GPS pornit au arătat că aceasta nu a rezistat mai mult de 8 *ore*, în funcţie de valoarea parametrului care dă periodicitatea pentru citirea GPS. Un compromis între acurateţe (frecvenţă mare de citire a GPS) şi consumul de energie trebuie explorat.

De asemenea, trebuie investigată performaţa algoritmului PSC şi valorile optime ale parametrilor de care acesta depinde, pentru cazul în care lungimea cuantei de timp cu care se lucrează nu este 1 oră, ci mai puţin.

CONCLUZII

S-a implementat o aplicaţie prototip pentru Android, multimodulară, care permite înregistrarea timpului petrecut de un turist într-un obiectiv turistic, estimarea prin analogie a timpului pe care îl va petrece în obiective similare şi generarea unui orar de vizitare a obiectivelor dorite pe baza acestor estimări. Momentele în care turistul intră, respectiv părăseşte un anumit obiectiv sunt înregistrate manual, dar în viitor acest lucru intenţionăm să îl facem automat. Un modul de rezolvare a problemelor de satisfacere a constrângerilor prin *Stochastic Local Search* generează apoi un orar pentru vizită, folosind aceste estimări.

Ca dezvoltări ulterioare, dorim explorarea parametrilor optimi (periodicitatea) de înregisrare de date pentru conservarea energiei bateriei, detectarea automată a momentelor de intrare şi părăsire a unui obiectiv, detectarea automată a obiectivelor care eventual nu sunt înregistrate şi îmbunătăţire a performanţelor algoritmilor de căutare prin detectarea experimetală a probabilităţilor care ghidează comortamentul acestora.

MULŢUMIRI

Parte din cercetare s-a derulat în cadrul proiectului "Argumentare structurată pentru suport decizional cu constrângeri normative", program PN-II Cooperări Bilaterale România-Republica Moldova, 2013-2014, UEFSCDI. Modelarea problemei ca un PSC a folosit ca punct de plecare o soluţie propusă de Szilard Mandici.

REFERINŢE

1. M. D. Choudhury, M. Feldman, S. Amer-Yahia, N. Golbandi, R. Lempel, and C. Yu, "Automatic construction of travel itineraries using social breadcrumbs," in *HT*, 2010, pp. 35–44.
2. G. A. Grimnes, S. Chalmers, P. Edwards, and A. D. Preece, "Granitenights - a multi-agent visit scheduler utilising semantic web technology," in *Proceedings of the 7th International Workshop on Cooperative Information Agents*, 2003, pp. 137–151.
3. H. H. Hoos and T. Stützle, *Stochastic Local Search: Foundations & Applications*. Elsevier / Morgan Kaufmann, 2004.
4. F. Lorenzi and F. Ricci, "Case-based recommender systems: a unifying view," in *Proceedings of the 2003 international conference on Intelligent techniques for Web Personalization*, ser. ITWP'03. Berlin, Heidelberg, Springer-Verlag, 2005, pp. 89–113.
5. C. Lucchese, R. Perego, F. Silvestri, H. Vahabi, and R. Venturini, "How random walks can help tourism," in *ECIR'12*, 2012, pp. 195–206.
6. S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 2nd ed. Prentice Hall, 2003.
7. Y. Zheng and X. Xie, "Learning travel recommendations from user generated gps traces," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 1, pp. 2:1–2:29, 2011.
8. C. Zhou, N. Bhatnagar, S. Shekhar, and L. Terveen, "Mining personally important places from gps tracks," in *Data Engineering Workshop, 2007 IEEE 23rd International Conference on*, 2007, pp. 517–526