

웹 서비스에 기반한 전기자동차용 관광-충전 스케줄러의 개발¹⁾

이병준, 이슬비, 임대용, 김혜진, 이정훈
 제주대학교 전산통계학과

e-mail:{eothsk, gwregx, dlaeodyd123, hjkim82, jhlee}@jejunu.ac.kr

Development of a driving-charging scheduler based on web services for electric vehicles

Byung-Jun Lee, Seulbi Lee, Dae-Yong Im, Hye-Jin Kim, Junghoon Lee
 Dept of Computer Science and Statistics, Jeju National University

요 약

본 논문은 전기자동차를 이용한 관광에 있어서 사용자가 선택한 목적지 집합에 대해 전기자동차 충전까지 고려하여 방문 순서를 결정하고 그 결과를 사용자 단말기 App을 통하여 보여주는 관광 스케줄러를 구현하였다. 스케줄링 엔진은 방문총거리가 짧으면서 충전대기시간을 최소화하기 위해 A* 알고리즘과 유전자 알고리즘으로 구현되었고 웹서비스 방식으로 단말기와 상호작용함으로써 서비스의 확장성과 편의성을 기한다.

1. 서론

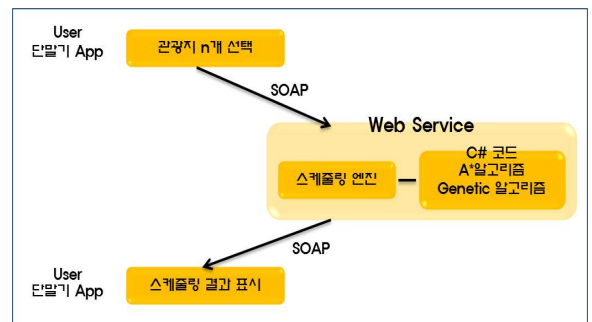
최근 스마트그리드가 이슈가 되고 스마트그리드의 한 분야인 스마트 트랜스포테이션에 대한 관심도 급증함에 따라 전기자동차가 핵심 아이콘으로 부상하고 있다. 이에 다양한 전기자동차 텔레매틱스 서비스에 대한 연구 개발이 필요한데 특히 제주도는 관광과 결합한 서비스의 도입이 더욱 필요하다.

전기자동차를 렌트한 관광객은 대개 공항을 출발하여 여러 목적지를 방문하고자 한다. 이 때 전기자동차가 완전충전이 된 경우라 할지라도 도로 여건과 주행스타일에 의해 배터리의 소모가 발생하게 되고 사용자는 관광 중에 전기자동차를 충전해야 한다[1]. 예를 들어 사용자가 출발전에 방문하고자 하는 목적지가 해안도로, 공연장, 음식점 등이라면, 드라이브 코스는 주행 중에는 충전이 불가능하고 공연장은 운전자가 공연을 관람하는 동안 차량 충전이 가능하다.

이처럼 다중 목적지의 경우 어떤 순서로 가야 방문 총거리가 짧은지에 대한 고전적인 TSP (Traveling Salesman Problem) 문제뿐만 아니라 어떤 순서로 방문하면 충전 대기시간이 최소화될 수 있는지에 대한 문제를 해결해주면 관광에 있어서 낭비시간을 줄여 최적의 관광 스케줄이 가능하게 된다. 또한 각각의 관광지별 소요시간을 계산하고 관광지 주변의 충전 인프라 상태를 파악하면 정확한 스케줄링이 가능하다.

TSP문제는 목적지가 n 개일 때 $O(n!)$ 의 시간 복잡도를 갖

는 NP-problem에 해당한다. 따라서, 최적성은 희생하더라도 빠른 시간에 적정수준의 답을 찾을 수 있는 휴리스틱의 도입이 바람직하다[2]. 본 논문에서는 순서 결정을 위해 유전자 알고리즘을, 점대점 경로를 위해서 A* 알고리즘을 사용하여 스케줄링 엔진을 C#으로 구현하고, 이를 웹 서비스와 연동하여 스케줄링 결과를 안드로이드 App으로 보여준다.



(그림 1) 관광 스케줄링의 개요

2. 관광 스케줄링 엔진

스케줄을 생성하는데 있어서 기본적으로 노드와 링크로 구성된 그래프 형태가 경로설정에 제공된다. 도로네트워크의 비용에 따라 각 목적지간 비용을 계산하는 A* 알고리즘에서 main 함수는 목적지까지의 최단거리를 구하고자 하는 AStarPath 함수와 약 17,840개의 도로 자료를 포함한 text 파일을 불러오는 함수를 포함하고 있다. text 파일에서 교차로의 x, y 좌표와 각 교차로간의 거리, 연결

1) 본 연구는 중소기업청에서 지원하는 2012년도 산학연공동기술개발사업(2012-0418)의 연구수행으로 인한 결과물입니다.

된 다음 교차로의 번호 등의 자료를 불러온다. 이 자료를 바탕으로 출발 지점의 교차로와 도착 지점의 교차로를 지정하면 AStarPath 함수가 현재 위치까지의 거리를 계산하고 인접해 있는 교차로를 알 수 있다. 특히 GetMinPotential 함수는 (그림 2)와 같이 현재 위치에서 연결된 교차로 중 목표 지점까지의 일직선 거리 값을 추정하는 계산을 포함하고 있으며 그 값들을 비교하여 가장 작은 값을 찾아내고 다음으로 검색할 교차로를 결정한다. 결국 A*알고리즘을 통해서 적합도를 비교하며 다음 교차로를 선택하다가 목적 지점까지 도착하면 탐색을 멈추게 되며 해당 경로와 그 거리를 알 수 있다.

```
int GetMinPotential(){
    int i, idist, curmin = -1;
    double dist, min = 1000000.0, dx, dy;
    for (i=0; i < totalNode; i++) {
        if (done[i] == 1) continue;
        if (t[i] == LARGEST) continue;
        dx = fabs (node[Ldest].x - node[i].x);
        dy = fabs (node[Ldest].y - node[i].y);
        dx = dx * 31.0 / 0.0002777778;
        dy = dy * 25.0 / 0.0002777778;
        dist = sqrt( dx * dx + dy * dy);
        if (t[i]+(int)dist < min) {
            curmin = i;
            min = t[i]+(int)dist;
        }
    }
    return (curmin);
}
```

(그림 2) A*알고리즘

3. 관광 스케줄러의 구현 결과

또한 유전자 알고리즘에 의한 진화방식인 selection과 crossover, mutation을 구현하고 초기 population으로부터 효율적으로 진화함으로써 충전대기시간이 개선된 방문 스케줄을 찾는다. (그림 3)

```
begin
initialize population
For the given # of iterations
    selection
    crossover
    mutation
    evaluate fitness
}
end
```

(그림 3) 유전자 알고리즘

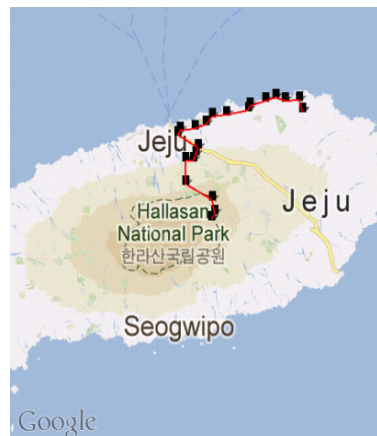
C#으로 구현된 A* 알고리즘과 유전자 알고리즘의 스케줄링 엔진은 안드로이드 기반에서는 속도가 느리게 된다. 따

라서 이를 고성능 서버에 구현하고 이기종 시스템 간 정보 교환이 가능한 웹 서비스를 이용하여 다양한 이기종 단말기들에게 공통적인 서비스를 제공한다.

(그림 4)와 같이 사용자가 관광지 select를 누르게 되면 안드로이드 App으로 계산 결과를 XML로 파싱하여 (그림 5)와 같은 지도 화면으로 보여주게 된다. 이러한 서비스는 식당 선후관계, 개인별 선호도 등 다양한 요구사항을 효율적으로 반영할 수 있다[3].



(그림 4) 관광지 선택 화면



(그림 5) 관광 순서 결과 화면

참고문헌

[1] J. Lee, et al., "Integration of battery charging to tour schedule generation for an EV-based rent-a-car business," LNCS, vol. 7332, pp.399-406, 2012.
 [2] B. Lee, et al., "Tour and charging scheduler development based on simulated annealing for electric vehicles," CCIS, vol. 353, pp.189-194, 2012.
 [3] J. Lee, et al., "Tour schedule generation integrating dining options for electric vehicles," LNEE, vol. 214, pp.299-305, 2012.