



저작자표시-비영리-변경금지 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.

다음과 같은 조건을 따라야 합니다:



저작자표시. 귀하는 원저작자를 표시하여야 합니다.



비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.



변경금지. 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 [이용허락규약\(Legal Code\)](#)을 이해하기 쉽게 요약한 것입니다.

[Disclaimer](#)

碩士學位論文

(n+1)TSP 모델링에 기반한
전기자동차 투어-충전 스케줄러

濟州大學校 大學院

電算統計學科

吳豪虔

2014年 12月

(n+1)TSP 모델링에 기반한
전기자동차 투어-충전 스케줄러




指導交綏 李 政 勳

吳 豪 虔

이 論文을 理學 碩士學位 論文으로 提出함

2014年 12月

吳豪虔의 理學 碩士學位 論文을 認准함

審査委員長 _____ 박 경 린 
委 員 _____ 이 정 훈 
委 員 _____ 송 준 모 

濟州大學校 大學院

2014年 12月

A tour and Charging scheduler for electric vehicles
based an $(n+1)$ Traveling Salesman Problem
Modeling.

Ho-Gun Oh

(Supervised by professor Junghoon Lee)

A thesis submitted in partial fulfillment of the requirement for the
degree of Master of Science

2014. 12.

Department of Computer Science and Statistics
GRADUATE SCHOOL
JEJU NATIONAL UNIVERSITY

목 차

Abstract	iii
I. 서 론	1
II. 배경과 관련 연구	4
1. 전기자동차 동향	4
2. 관광-충진 스케줄	6
III. 제안 기법	12
1. 기본 가정	12
2. ODBC를 통한 리눅스와의 연결	13
3. 제안 기법	20
IV. 성능평가	23
V. 결 론	28
VI. 참고문헌	30

List of Tables

[표 1] 제주도내 전기자동차 충전기 설치 현황 (2014년 7월 기준)	5
--	---

List of Figures

[그림 1] 급속충전기 정보 테이블 구조	14
[그림 2] 프로젝트 속성 변경	15
[그림 3] 헤더파일 프로젝트에 포함	15
[그림 4] 라이브러리 디렉토리 추가	16
[그림 5] 프로젝트 폴더에 libmysql.dll 추가	16
[그림 6] 데이터베이스 연동 소스	17
[그림 7] 데이터베이스 내용을 읽어온 결과	19
[그림 8] 관광지 인접행렬	19
[그림 9] 충전기 인접행렬	19
[그림 10] 충전기 제한 거리기반 휴리스틱	22
[그림 11] Effect of the number of destinations	24
[그림 12] Effect of the number of picks	25
[그림 13] Effect of the TSP length	26
[그림 14] Effect of the number of picks	27
[그림 15] Effect of TSP length	27

Abstract

전기자동차는 배터리에 저장된 전기에 의해 구동되므로 운송 시스템마저도 전력 네트워크의 한 부분으로 만들어 에너지 사용의 효율을 향상시킬 수 있을 뿐 아니라 공해물질 배출을 감소시킬 수 있다. 반면 주행거리가 짧고 충전시간이 길다는 단점이 있는데 급속 충전으로도 수십분이 소요된다. 이를 극복하기 위해 지능적인 컴퓨터 알고리즘을 바탕으로 에너지 효율적인 주행계획을 수립하여야 한다. 본 논문에서는 다중목적지를 방문하는 렌터카를 위한 방문-충전 스케줄러를 설계하고 그 성능을 평가한다. 현재 급속 충전기의 확산과 일일 관광 거리를 고려한다면 하루 1회의 급속충전이면 관광에 지장이 없다. 결국, 해결하고자 하는 문제는 n 개의 관광지가 주어졌을 때, m 개의 충전기중 하나를 추가적으로 포함하는 $(n+1)$ 목적지에 대한 방문순서를 결정하는 것이다. 모든 m 개의 충전기에 대해 $(n+1)!$ 의 방문순서에 따르는 비용을 계산하면 최적의 답을 구할 수는 있으나 계산시간이 폭증한다. 따라서, m 대신 오직 m' ($m' \ll m$)개만의 충전기를 사전에 지능적으로 필터링하여 서치 스페이스를 줄임으로써 납득할 만한 수준의 답을 빠른 시간 내에 구할 수 있다. 이를 위하여 모든 m 에 대해 관광지 집합까지의 최소거리를 구하고 가장 근접한 m' 개의 후보를 선택한다면 전체 투어 길이의 증가를 최소화할 수 있으며 배터리 소모도 줄일 수 있어서 추가적인 충전도 피할 수 있다. 각 $(n+1)$ 목적지의 방문순서에 있어서 운행에 따르는 배터리 잔량을 추적하여 배터리 고갈이 일어난다면 그 솔루션은 제외한다. 프로토타입 구현에 의해 실제 제주도 도로 네트워크를 대상으로 수행된 성능 평가는 주어진 인자 집합에 대해 최적의 스케줄과 비교시 투어 길이의 차이는 단지 1%에 불과하고 0.7의 확률로 최적의 스케줄을 발견한 반면, 5/29의 속도향상을 불러왔다. 순수 관광거리에 따르는 실험 결과, 최적의 스케줄과 비교하여 주행거리는 최대 7% 정도밖에 증가하지 않았다.

Abstract

Electric vehicles, or EVs in short, are powered by battery-stored electricity, making even the transport system a part of the power network. They cannot only enhance the energy efficiency but also reduce air pollution, however, their driving range is quite short and it takes tens of minutes to charge an EV even with a fast charger. Intelligent computer algorithms can overcome this problem by finding an energy-efficient route. Targeting at EV rent-a-cars willing to visit multiple destinations, this thesis designs a tour-and-charging scheduler and evaluates its performance. Considering the ongoing penetration of fast chargers and the common daily tour length, a single fast charging is enough for an average trip schedule. In addition to the given n destinations, our scheme selects 1 out of m charging stations, simultaneously making a tour plan for $(n+1)$ spots. For m chargers, the cost estimation of all $(n+1)!$ feasible sequences can obtain the best solution, but inevitably brings computation time explosion. Our approach cuts down the search space not taking m but m' ($m \gg m'$) stations intelligently filtered by the distance criteria. It finds an acceptable quality schedule within a sufficiently fast response time. Here, for all m station, the minimum distance to any of n destinations is calculated and the smallest m' stations are selected. The heuristic can suppress the increase in the tour length, reduce battery consumption, and avoid additional charging. The performance measurement result, obtained by a prototype implementation on real-life tour spot distribution in Jeju, Republic of Korea, shows that our scheme increases the tour length just by 1 %, compared with the optimal scheme, achieving the speedup of 5/29 for the given parameter set having 7 destinations. The increase in the pure tour distance can lead to 7 % overhead in the final tour length.

I. 서 론

전기자동차(EV : Electric Vehicle)는 석유 연료와 엔진을 사용하지 않고, 전기 배터리와 전기 모터를 사용하는 자동차를 말한다[Timpner and Wolf, 2014]. 1873년에 가솔린 자동차보다 먼저 제작되었으나, 배터리의 중량이 무겁고 충전 시간이 너무 긴 문제로 인하여 널리 보급되지는 못하였다. 그래도 구조가 간단하고 내구성이 크며 운전하기가 쉬웠기 때문에 주로 여성용으로 미국에서 1920년대 중반까지 소량 생산되었으나 최근 공해문제가 심각해지고 스마트그리드가 대두되면서 다시 개발되기 시작했다.

전기자동차의 배터리는 그리드로부터 충전되는 동안 전력 네트워크의 일부가 된다. 그리고 전기자동차는 공해를 일으키지 않아 매우 친환경적이고, 풍력이나 태양광 등 신재생에너지원으로 충전이 가능하다[Mischinger et al. 2012]. 또한 발전소를 통하여 생산된 전기를 송배전 과정을 거쳐 전기자동차까지 보내서 사용하는 전 과정(LCA : Life Cycle Assessment)을 감안하더라도 내연기관차에 비하여 최소 약 2배 이상의 에너지 효율을 가진다[박광철. 2012]. 자동차 회사에서는 약 3배 수준(가솔린 30%, 전기차 80%)으로 계산하기도 한다. 내연기관은 에너지 변환효율이 약 29%, 동력전달과정 등의 효율이 66%가 되어 최종효율이 19%이다. 반면 전기자동차는 복합 화력발전 및 송전과정의 에너지 변환효율이 약 45%이고 충전 및 모터 구동과정에서 에너지 변환 효율이 85%로 최종 효율이 38%에 달하여 내연기관대비 에너지효율이 높다. 또한 엑셀레이터를 밟지 않고 관성으로 운행될 때나 브레이크를 밟을 때 그 구동력으로 배터리가 충전이 되기도 한다. 내연기관차를 주행관련 부품이 최소 130여 개 인데 반하여 전기자동차는 2~3개인 것만 봐도 내연기관차의 구동손실을 추산할 수 있다.

이러한 장점들에도 불구하고 전기자동차의 활성화는 아직 미흡하다. 우선 같은 종류의 차에 비하여 가격이 2배 이상 비싸며 충전시간이 내연기관 차량에 비하여 매우 길다. 그럼에도 불구하고 1회 충전 주행 가능한 거리가 짧다. 충전은 교류에 기반한 완속충전과 직류에 기반한 급속충전 방식으로 나누어지는데, 완속충전의 경우 배터리가 100% 충전 될 때까지 6시간가량 소요되고 급속충전 시 80%충전까지 20~30분가량이 소요된

다[Botsford and Szczepanek, 2009][Veneri et al., 2013]. 급속충전기는 전기자동차의 활성화에 바람직하지만 급속충전기는 완속충전기 대비 매우 고가이며, 많은 전기자동차들이 동시에 충전할 경우 에너지 소비의 급격한 증가로 그리드에 악성부하가 될 수 있다[Lee and Park, 2015]. 또한 하루에 2회 이상의 급속충전을 할 경우 배터리 수명을 단축시킬 수 있다.

1회 충전 주행가능거리는 차량별로 다르나 국내에 시판된 차량들은 평균적으로 130km 정도 알려져있어, 이 정도의 주행거리로는 장거리 운행에 어려움이 있다. 하지만 대한민국에서 차량들의 1일 평균 주행거리는 37km이며 렌터카 75.9km, 화물차 5톤이상 111km, 법인택시 292km로 택시를 제외한다면 200km 이상인 경우가 거의 없다[교통안전공단, 2010]. 택시를 제외한 주행거리는 전기요금의 저렴한 심야시간 때의 완속충전으로 커버가 되지만 낮 시간에 운행을 하다보면 급속충전이 1회 정도 필요하다. 만약 전기자동차의 주행가능거리를 초과할 경우 어디서 충전을 해야 할지 고려하는 것이 중요하다. 또 EV 렌터카가 유망한 제주도에서 1일 관광거리는 100~130km로 추산되며, 다양한 지형을 주행하기 때문에 1회 충전시 주행가능거리를 초과할 경우가 많다. 전기자동차 운전자는 차량 운행 전에 목적지에 따른 주행 경로와 충전이 필요할 때의 충전소를 결정하는 것이 바람직하다.

급속 충전을 포함한 방문순서 결정문제는, 시간복잡도가 NP(Non Polynomial)로 많이 알려진 TSP(Traveling Salesman Problem: 외판원문제) 알고리즘을 변형시키는 것을 우선적으로 고려할 수 있다[Shim et al., 2012]. TSP와 관련하여 많은 휴리스틱 알고리즘들이 개발되어 있지만 기본적으로 n 개의 목적지의 방문 순서를 정한다. DC충전을 고려한다면 N 개 목적지에 그치지 않고 m 개의 충전기 중에서 하나를 선택함과 동시에 $(n+1)$ 개의 목적지에 대한 방문순서를 결정하여야 한다. 최적을 솔루션을 찾기 위해서는 목적지 세트에 후보 목적지를 하나씩 교체시키면서 각각 TSP를 수행시켜야 할 필요가 있다. 따라서 이 세트의 시간복잡도는 $O(m \times (n+1)!)$ 이다. 하지만 후보 목적지로 선택될 가능성이 낮은 충전기를 제외하면 응답시간을 줄일 수 있다. 또한, 가능성이 높은 충전기인 목적지 주변 충전기만을 조사함으로써 계산시간을 좀 더 줄이고 현실화할 수 있다. 기존연구에 따르면, 목적지 근처 DC 충전기는 총 관광서비스 거리와 시간을 줄일 가능성이 높다[Lee and Park, 2013].

이와 관련하여, 본 논문에서는 충전기 정보의 공간 데이터베이스를 연계하고 주어진 목적지 집합에 대한 관광 및 충전 일정을 확인하기 위해 급속 충전기를 선택하는 거리 기반 휴리스틱을 설계한다. 여기에서 시스템 설계는 대한민국 제주시의 도로망과 충전기 및 관광지의 지리 분포를 대상으로 하고 있다. 본 논문의 구성은 다음과 같다. 1장에서 논문에서 해결하고자 하는 문제를 개관하고 2장에서는 배경과 관련연구들을 조사한다. 3장에서는 제안하는 기법을 구축하기 위한 서비스 환경을 기술하고 이에 따르는 관광 충전 스케줄 기법을 자세히 설명한다. 4장에서는 제안된 기법을 프로토타입 구현을 통해 성능평가하며 최종적으로 5장에서는 본 논문을 요약하고 결론을 도출한다.

II. 배경과 관련 연구

1. 전기자동차 동향

제주특별자치도에서는 2009년 12월부터 2013년 5월까지 제주 스마트그리드 실증사업을 진행하였다[지능형전력망협회 and (주)한국전력신문사, 2012]. 실증사업은 지능형 소비자, 지능형운송, 지능형신재생, 지능형전력망 등 5개의 분야로 구성되어 168개의 유관기업으로 구성된 12개의 컨소시엄이 참여하여 진행되었다. 5개의 분야 중 지능형 운송에 전기자동차가 포함되는데 실증사업을 통하여 다양한 서비스들이 개발되었다. 전기자동차 기반 가상발전소 운영서비스라하여 전력수요 피크 시 미운행 중인 전기자동차의 배터리 전력을 충전기를 활용하여 전력시장에 판매하는 서비스, 전기차 급·완속 충전 서비스로 전기차 배터리에 전기를 충전하는 전력 재판매 서비스, 전기자동차 이동충전서비스로 전기차 배터리 방전 시 응급구조차량이 출동하여 급속충전 서비스를 제공, 전기차 대여 서비스 등이 개발되었다.

2014년 현재 전기자동차 대여사업, 전기버스사업, 배터리 리스사업, 전기택시사업 등이 진행 중이거나 진행예정이다. 또한 전기화물차를 이용한 택배사업은 서울시에서 2014년 12월부터 2014년 5월까지 시범사업을 진행할 예정이다. 도심 택배에서 많이 이용되는 1톤 이하 소형 경유트럭은 저속운행을 하는데다 정차하는 경우가 많아 매연저감장치를 부착하기 힘들다고 한다. 이 때문에 주행거리 대비 오염물질의 배출이 엄청나다. 서울의 경우 초미세먼지의 52%가 자동차와 건설기계에서 발생한다.

질소산화물(NOx)은 67%가 경유차에서 발생된다. 또한 2007년 기준 수송수단별 온실가능 배출량을 살펴보면 도로 77%, 해운 12.7%, 항공 8.9%, 철도 0.7%로 도로 부

분에서 가장 많은 온실가스를 배출하고 있었다[김정구 and 이도경, 2009] .

제주에는 하반기 보급예정인 완속충전기를 포함하여, 완속·급속충전기는 약 1,056기가 설치될 예정이다. 2014년 7월 기준 급속충전기는 73기이며 DC급속충전기는 29대이며 계속 증설될 예정이다.

[표 1] 제주도내 전기자동차 충전기 설치 현황 (2014년 7월 기준)

구 분	합계	급속	완속
합 계	815	73	742
공공기관	211	39	172
전기렌터카	28		28
민간보급	435	8	427
스마트그리드 소계	141	26	115

제주특별자치도는 제주형 저탄소 녹색성장이라 하여 Carbon Free Island Jeju by 2030을 추진하고 있다[제주특별자치도, 2012]. 「가파도 Carbon Free Island구축」, 「스마트그리드 거점지구 추진」, 「전기자동차 시범도시 구축, 제주 해상풍력 2GW 개발」, 「제주에너지공사 설립」 등이 포함되어 있다.

이중 전기자동차 시범도시 구축과 관련하여 3단계로 나누어 내연기관 자동차를 전기자동차를 대체하는 목표를 세웠고 전기자동차를 1단계(2017년까지) 10%, 2단계(2020년까지)30%, 3단계(2030년까지) 100%를 보급하는 계획을 세웠다.

그리고 전기자동차 선도도시 구축을 위하여 택시, 세어링, 택배 등 각각의 산업에 대한 비즈니스 모델을 분석하였으며, 전기자동차의 단계별 보급에 따른 전력수요를 분석하고 각 단계별 보급방안과 충전인프라의 구축 방안을 모색하였다. 현재 전기차 세어링 및 렌터카, 택시 사업이 한창 진행 중이며 내년 전기버스 및 배터리 리스 사업이 진행될 예정이다.

2. 관광-충전 스케줄

외판원 문제(TSP : Traveling Salesperson Problem)는 처음 해밀턴과 영국의 수학자 토모스 커크먼에 의해 1800년대에 제안되었다[Li and Sun, 2009]. TSP는 조합 최적화 문제의 한 예로서 주어진 n 개의 도시들을 단 한번씩 만 거쳐 출발지로 되돌아오는 경로들 중 가장 작은 비용이 소요되는 경로를 찾는 문제이며, 고전적인 최적화 문제로 널리 알려져 있다[Lin and Kernighan, 1971]. 다시 말해서, TSP는 외판원이 C_0 라는 어떤 도시에서 시작해서 나머지 도시를 한번씩 만 방문하여 최종적으로 시작 도시인 C_0 로 다시 돌아오기 위한 최단 경로를 찾는 문제이다. 도시 개수 증가에 따라 해를 구하는 데에 소요되는 계산량이 폭증하여 NP-Hard 문제로 분류된다[Garey and Johnson, 1979]. NP 문제들은 대부분 휴리스틱 알고리즘이나 메타 휴리스틱 알고리즘을 적용한다. TSP는 위치 문제, 통신 문제, 우편 라우팅, 컴퓨터 파일 순서, 창고에서 순서 뽑기, 철도 최적화 등 여러 가지 응용을 갖고 있다. 지금까지 다양한 TSP 응용에서 최단경로를 찾기 위하여 다양한 최적화 알고리즘이 제안되었다. TSP를 해결하기 위한 대표적인 메타 휴리스틱 알고리즘[Bianchi et al. 2008]에는 유전자 알고리즘(Genetic Algorithm, GA)[Freisleben and Merz, 1996] [Goldberg and Kuo, 1987], 개미 집단 최적화(Ant Colony Optimization, ACO)[Gambardella and Dorigo, 1997][Dorigo and Blum, 2008], 입자 군집 최적화(Particle Swarm Optimization, PSO)[Zhong et al. 2006] 등이 있다. 메타 휴리스틱 형태의 최적화 알고리즘은 제한된 공간과 시간에서 해결되지 않는 문제를 해결하고자 많이 사용되고 있다. 메타 휴리스틱은 특정문제가 갖는 정보에 영향을 받지 않는 수준에서 다양한 문제에 적용 가능한 기법이다[Bianchi et al. 2008].

유전자 알고리즘은 인공유전 시스템으로 자연세계의 진화과정에 기초한 계산모델로 John Holland에 의해서 1975년 개발된 전역적인 최적화 알고리즘이다[Freisleben and Merz, 1996]. 유전자 알고리즘은 하나의 세대(generation)를 이루는 염색체(chromosome)가 사용자의 의도에 맞게 변화하면서 최적값을 찾아가는 과정으로 말

할 수 있다. 일반적으로 유전자 알고리즘의 특징을 언급할 때 다음의 4가지를 말한다[Goldberg and Kuo, 1987].

- ① GA는 하나의 매개변수 각각을 이용하는 것이 아니라, 매개변수의 군(set)을 이용해서 최적화를 실시한다.
- ② GA는 하나의 점에서 최적값을 찾아나가는 것이 아니라, 세대내의 염색체에서 동시에 최적값을 찾아나간다.
- ③ GA는 목적함수에 관한 특정한 수학적 표현을 요구하지 않고, 목적함수값(payoff)만을 이용한다.
- ④ GA는 확률개념을 이용해서 염색체를 변화시킨다.

유전자 알고리즘에서 가장 기본적으로 사용하는 연산자는 염색체를 직접 변화시키는 돌연변이연산자(mutation operator)와 교배연산자(crossover operator)가 있으며, 염색체를 표현하는 방법에 따라서, 이진코딩법(binary coding)과 실수코딩법(real-value coding)으로 나눌 수 있다. 이와 같이 기본적인 연산자만을 이용해서 구현된 유전자 알고리즘을 SGA(simple genetic algorithms)이라고 말하며, 일반적으로 제약조건이 아주 엄격하거나, 목적함수가 특이한 형태를 가지지 않는 한, SGA만으로도 충분히 좋은 결과를 얻을 수 있다.

이진코딩법의 경우에는 돌연변이연산자나 교배연산자가 염색체를 구성하는 이진수의 변화만으로 간단히 구현될 수 있는 장점이 있는 반면에, 염색체를 이루는 비트 중에서 자리수가 큰 왼쪽의 비트가 변화된다면 너무 큰 변화가 일어나는 단점이 생길 수 있다. 또한, 같은 영역의 제약조건을 가정한 경우, 실수코딩법에 비해서 필요한 메모리의 용량이 상당히 커지기 때문에, 동적계획법의 ‘차원의 저주(curse of dimensionality)’와 유사한 메모리 용량면의 제약이 생기게 된다. 실수코딩법은 이진코딩법이 가지고 있는 메모리용량면의 단점이 없고, 각 최적화문제에 알맞은 고유의 염색체를 구성하기가 용이하지만, 돌연변이나 교배와 같은 연산자가 염색체를 변화시켰을 때 각 염색체에 주어진 제약조건을 위반하지 않도록 조정하거나, 제약조건 위반에 따른 벌칙(penalty)이 각 염색체에 적절히 주어지도록 조정해야 한다.

개미 집단 최적화는 조합 최적화 문제를 해결하기 위한 메타 휴리스틱 탐색 방법으로, 에이전트라 불리는 개미들이 목적지를 향해 나아가는 동안 각 경로에 페로몬(pheromone)을 분비하고, 이후에 지나가는 에이전트들은 그 경로에 쌓여있는 페로몬 정보를 이용해 다음 경로를 선택하는 방법이다[Gambardella and Dorigo, 1997]. 에이전트들은 각 경로에 있는 페로몬 양을 기반으로 서로 간의 정보 교환을 통해 최적의 경로를 찾아가고 이러한 에이전트들의 행동 양식을 그대로 적용한 휴리스틱 탐색 방법으로 조합 최적화 문제인 순회 외판원 문제(traveling salesman problem), 순서 문제(sequential ordering problem), 이차 배정 문제(quadratic assignment problem), 차량 경로 문제(vehicle routing problem), Job-shop 스케줄링 문제(scheduling problems), 그래프 착색 문제(graphcoloring problem), telecommunications networks 등에서 최적의 해를 구하기 위해 사용되고 있다. 이러한 다양한 조합 최적화 문제를 풀기 위한 휴리스틱 탐색에서 강화와 다양화의 조화를 맞추는 문제는 해당 알고리즘의 성능에 많은 영향을 미치고 있다.

TSP를 풀기 위해 ACO 접근법의 하나인 메타 휴리스틱 방법인 개미 집단 시스템(Ant Colony System, ACS)은 실제 개미들이 먹이에서 집까지 가장 짧은 경로를 찾는 능력을 모방한 메타 휴리스틱 기법이다[Dorigo and Blum, 2008]. 일반적으로 ACS는 다음과 같은 방법으로 수행을 한다. 먼저 m 개의 에이전트들이 초기화 규칙(initialization rule)에 따라 무작위로 n 개의 노드(node)를 선택한 다음, 각 에이전트들은 상태전이 규칙(State Transition Rule)에 따라 다음에 방문할 노드를 선택하고 계속해서 탐색 과정을 거친다. 이러한 탐색과정을 거치는 동안 에이전트들은 지역 갱신 규칙(Local Updating Rule)에 따라 방문한 각 간선에 페로몬 양을 갱신하게 된다. 그리고 일단 모든 에이전트들이 탐색과정을 마치게 되면 전역 갱신 규칙(Global Updating Rule)에 따라 다시 한번 페로몬 양을 갱신하게 된다. 결국, 각 에이전트들은 짧은 간선을 선택하려는 휴리스틱 정보와 많은 양의 페로몬을 가진 간선을 선택하려는 페로몬 정보에 따라 탐색 경로를 완성하게 된다.

입자 군집 최적화는 새 무리가 둥지를 틀기에 가장 적합한 장소를 찾는 모습을 모

방한, 통계학적 최적화 방법 중 비교적 새로운 방식의 전략이다[James and Russell, 1995]. 전자공학자 R.Eberhart와 사회 심리학자 J.Kennedy가 1995년 유전 알고리즘 방식의 변형으로 개발한 PSO는 간단하고 효율적인 최적화 기법이다. 입자 군집 최적화 기법은 새 군집이 새로운 곳으로 이동하는 방법을 모사한 것이다. PSO에서 미지수 x 는 군집 내 새 한 마리의 행동 양식을 따라 하는데, 이 때 x 를 입자(Particle)라고 한다. PSO에서 입자는 GA의 염색체(chromosome)와 같은 역할을 한다. 하지만 GA와 달리 입자는 염색체처럼 각 세대 이후 폐기되는 것이 아니라 자신의 행동 즉 이동 방향과 속도를 바꾸는 식으로 진화한다. 이때 입자의 이동 방향과 속도는 목적지, 해를 향하는데 이를 위해 각 입자는 서로 정보를 교환하며 자신의 위치를 더 나은 곳으로 바꿔나간다. 이러한 PSO의 특징 덕분에 이를 사회적 모델이라고 지칭하기도 한다. 모든 입자가 자신의 정보와 사회적 정보를 동시에 이용해 새로운 장소를 찾기 때문이다. 즉 개개의 입자가 자기 주변을 탐색(local search)한 결과와 전체 군집이 탐색한 결과(global search)를 모두 참조해 새로운 위치로 진화하는 것이다. 이 때문에 자신의 정보와 군집의 정보에 따라 각각 얼마나 의존하느냐에 따라 PSO의 성능과 효과가 달라진다. 만일 입자가 자신의 정보를 더 많이 수용하면 현재 군집이 형성하는 장소가 아닌 대안 장소를 찾을 가능성이 커져 국소 최소값이 빠질 가능성을 줄이지만, 전체 군집의 수렴성은 줄어들고 최적 해를 찾는 시간이 길어진다. 반대로 군집의 정보를 더 신뢰하면 개개 입자가 주변으로부터 더 많은 정보를 얻게 되고 최적해를 찾는 시간은 빨라지지만, 군집이 국소 최소값에 이를 가능성이 증가한다.

GA은 최종단계 탐색능력이 부족하여 알고리즘을 수렴시키는 부분에 많은 조작이 필요하다[Freisleben and Merz, 1996]. ACO는 TSP처럼 최단거리 문제로 환원되는 문제에서 특히 좋은 성능을 보인다[Gambardella and Dorigo, 1997]. 그러나 많은 최적해 탐색문제들은 그래프로 표현될 수는 있지만 최단거리 문제로 환원되지 않는다. 따라서 응용 최적해 문제에서 좋은 성능을 내지 못하고 있다. PSO는 다른 메타휴리스틱 알고리즘들에 비해 수렴속도가 빠른 반면, 지역해에서 탈출하는 능력이 부족하다[Zhong et al. 2006].

GA를 TSP에 적용하기 위해서는 적합도와 교차연산자나 전위연산자 등을 문제에 맞게 수정해야 한다. 특히 두 개의 방문순서를 가지고 단순교차연산을 했을 때 중복되는 도시나 방문을 하지 않은 도시가 생기게 될 수 있다. 따라서 TSP에 적합한 제약조건을 만족하는 연산자를 사용해야 한다[Goldberg, 1989]. TSP를 해결하기 위해 GA를 개선하여 최적화 효율을 향상시키는 연구가 많이 이루어지고 있다. [Kondo and Watanabe, 2011]은 파라미터 없는 특성을 유지시키면서 최적화 효율을 높이는 파라미터 없는 분산 유전자 알고리즘을 제안하였다. [Matei and Pop, 2010]는 TSP를 일반화한 GTSP를 해결하기 위하여 지역(local)과 전역(global) 연결을 구분한 새로운 GA를 개발하였다. [von Poser and Awad, 2006]는 기존 GA에 특정 행동을 나타내는 실수코딩법을 이용하여 고체 폐기물 수집을 위한 최적의 경로를 제공하였다.

ACO는 TSP 문제에 적용시킬 때 좋은 예가 된다. ‘모든 도시는 연결되어 있다.’, ‘모든 도시는 한번씩만 거쳐야 한다.’, ‘모든 도시들을 방문한 뒤 제자리로 돌아온다.’의 전제조건을 주고 여러 마리의 개미를 보낸다[23]. 탐색 가능한 에지들을 이용하여 경로를 생성하는 개미들이 최적해를 선택하기 위해서는 에지들의 페로몬 양에 대한 정보가 중요한 역할을 한다[Gambardella and Dorigo, 1997]. 따라서 에지의 페로몬 정보를 갱신하는 연구가 많이 이루어지고 있다[23]. [Li et al. 2011]은 대규모 TSP를 대상으로 한 노이즈를 갖고 있는 응용에 대한 자기 적응 밀도 기반 공간 클러스터링하는 새로운 ACO를 제안하여, 수렴 속도를 증가시켰고 인공 경험에 대한 알고리즘 의존도를 감소시켰다. [Li et al. 2008]은 휴리스틱 정보의 영향력과 페로몬 자취 사이의 트레이드 오프(Trade-off)를 결정하기 위한 Held-Karp 하한 경계를 사용함으로써, 확률 선택 메커니즘을 채택하여 ACO를 개선하였다. [Hara et al. 2010]는 정상개미 이외에 부분적 솔루션을 구축하는 이기종 개미인 탐구개미를 사용하여 탐색 성능을 향상시키는 새로운 ACO를 제안하였다.

PSO를 TSP에 적용하기 위하여 GA 이론과 결합을 이루고 있다[Zhong et al. 2006]. PSO의 변경량은 연속치로 표현된다. 따라서 연속형 변수를 포함하는 문제와 일반적인 이산형 변수를 포함하는 문제에 확장 적용된다. 이에 PSO에서는 개별 경로탐색 알고리즘을 삽입하는 방법으로 제안되고 있다[Shi et al. 1993]. [Liu and Huang,

2010]에서는 GA의 교배(crossover)와 돌연변이(mutation) 연산자를 사용하여 새로운 속도공식을 얻어서 특별한 혼합 개별 PSO기법을 제안하여 수렴 속도를 향상시켰다. [Akhand et al. 2012] 는 기존 PSO보다 세대수가 더 적은 조건하에서 최적의 솔루션을 찾아내는 부분적 탐색 기능을 갖는 PSO를 제안하였다. 특히, [Chen et al. 2010] 는 다양한 관광객들의 요구를 충족시킬 수 있도록 적응형 여행 경로 계획 멀티미디어 시스템을 제안하였다. 여행 경로 계획이라는 TSP를 해결하기 위해서, PSO를 사용하여 최적의 계획된 여행 경로를 찾아내고 교환 휴리스틱 기법을 사용하여 성능을 향상시켰다. 결국 해당 멀티미디어 시스템은 관광객에게 요구에 따라 관광비용을 최소로 하는 여행, 관광시간을 최소로 하는 여행, 관광 비용과 시간을 둘다 최소로 하는 여행 등을 추천해 준다.

III. 제안 기법

본 장에서는 제안된 기법에 대해 설명한다. 우선 TSP을 기반으로 한 전기자동차 충전기 추천 기법과 공간 데이터베이스를 소개하고 다음으로 자세한 동작을 설명한다.

1. 기본 가정

본 기법은 대한민국 제주특별자치도를 대상으로 적용하였으며, 현재 29개의 급속충전기가 설치되어 있다. 관광 서비스를 위해서는 목적지를 경유하는 TSP경로를 결정하여야 하며, TSP경로의 결정을 위하여 각 목적지 쌍의 거리, 즉 비용을 알 필요가 있다. 관광 서비스는 관광을 위해 방문할 목적지의 리스트가 미리 주어진다. 서비스 시작 전 목적지가 결정되기 때문에 Dijkstra or A*알고리즘에 의하여 실행시간에 관계없이 각 목적지 쌍의 비용의 계산이 가능하다. 본 논문에서는 제주도 협력업체가 제작한 도로네트워크를 이용하여 목적지들 간의 단대단 거리를 계산한다.

관광서비스는 보통 승용차, 또는 승합차를 이용한다. 교통안전공단의 2009년 자동차 주행거리 실태조사 보고서를 살펴보면 렌터카들의 1일 평균 주행거리를 알 수 있다. 전국의 경우 75.9km, 제주의 경우 71.2km를 보이며 월간 2,2136km(71.2km×30일)를 주행한다[교통안전공단. 2010]. 제주지역 렌터카 승용차량은 월 평균 398.9시간을 대여한다[KB금융지주 경영연구소, 2013]. 이를 시간단위로 계산하면 1시간당 주행거리는 5.4km ($5.355\text{km} = 2,2136\text{km} / 398.9\text{시간}$)로 이를 평균시속 60km로 가정할 때 약 5.4분정도만 운전하였음을 알 수 있다. 따라서 주행 이외의 시간에 충전인프라에 접근이 가능하다면 충전기를 충분히 활용가능하다.

앞서 설명한 제주 렌터카의 평균 주행 길이는 71.2km라고 할 수 있으나 실제 관광을 위한 주행은 이보다 훨씬 높거나 작을 수 있다. 현재 전기자동차는 국내에서도 민간에 보급 중이며 이 차량들의 주행가능거리는 70~146km 사이에 있다. 전기자동차의 짧은 주행거리와 충전 인프라 부족으로 인해 운전자는 주행 중 방전에 대한 우려가 높고 실제 계기판은 15%의 오차를 포함한 것으로 알려져 있어서 실제 주행 가능한 거리보다 더욱 짧은 거리만 운행한다[제주관광공사, 2013].

따라서 렌터카의 일일 주행거리와 전기자동차의 주행가능거리를 고려하면, 일일 관광서비스시 한번의 충전이 필수적이며 이로 인하여 관광서비스의 시간과 주행거리가 증가할 수 있다.

2. ODBC를 통한 리눅스와의 연결

현재 제주대학교 스마트그리드 연구센터의 웹서버인 jejuevrc.jejunu.ac.kr에는 웹 페이지 콘텐츠 외에도 전기자동차를 위한 다양한 ICT 서비스를 제공하고 전기자동차에서 발생하는 다양한 스트림 데이터를 분석하기 위해 mysql 데이터베이스와 Apache Hadoop이 설치되어 있다. mysql 데이터베이스에는 제주도내 충전 설비에 관련된 테이블들이 정의되고 급속 완속 충전기들에 대한 정보가 축적되고 있다. [그림 1]에서 보는 바와 같이 이 테이블을 충전기 id, 주소, 타입, 제조사, 위치, 커넥터, 관리자 등을 관계형 테이블로 저장하고 있으며 type 필드는 AC 혹은 DC 값을 나타낸다. 또 loc 필드는 충전기의 경위도 좌표를 갖고 있는데 단순히 실수형의 WGS84 좌표를 사용하지 않고 mysql spatial extension에서 제공하는 point 타입을 사용하여 다양한 공간질의 지원을 지원하도록 한다.

DC 충전기를 선택하기 위한 서비스에서는 동적 혹은 주기적으로 DC 충전기들의 위치를 다운로드받아 관광객들과의 거리를 저장하고 있어야 한다. 이를 위해 Visual Studio로 제작한 C 프로그램에서 mysql database와의 연동을 하여 현재 혹은 향후 수정되고 추가될 정보들을 충전 서비스에 결합하여야 한다.

```

mysql>
mysql> desc tmpChargers;
+-----+-----+-----+-----+-----+-----+
| Field      | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id         | int(5)    | NO   | PRI | NULL    |       |
| addr      | char(100) | YES  |     | NULL    |       |
| type      | int(5)    | YES  |     | NULL    |       |
| manufacturer | char(20)  | YES  |     | NULL    |       |
| loc       | point     | NO   |     | NULL    |       |
| connector  | int(5)    | YES  |     | NULL    |       |
| manager   | char(20)  | NO   | PRI | NULL    |       |
+-----+-----+-----+-----+-----+-----+
7 rows in set (0.00 sec)

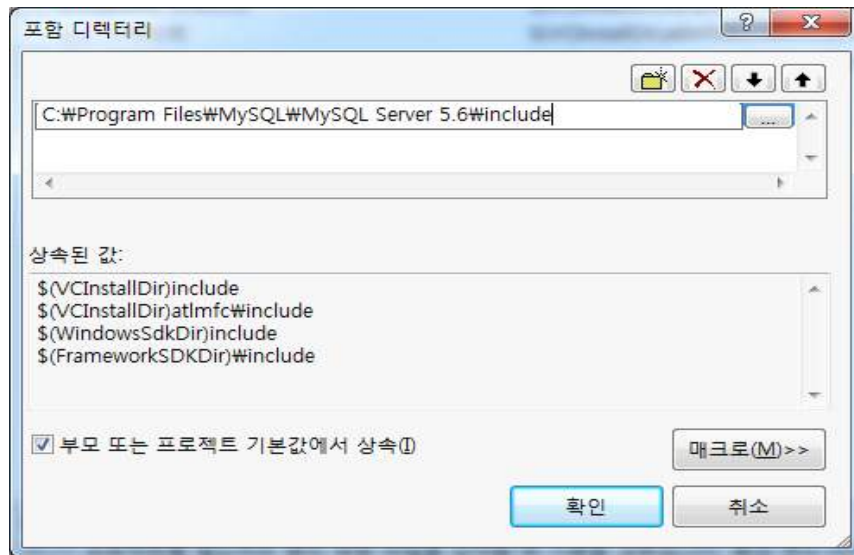
mysql> select count(*) from tmpChargers;
+-----+
| count(*) |
+-----+
|        295 |
+-----+
1 row in set (0.00 sec)

mysql>

```

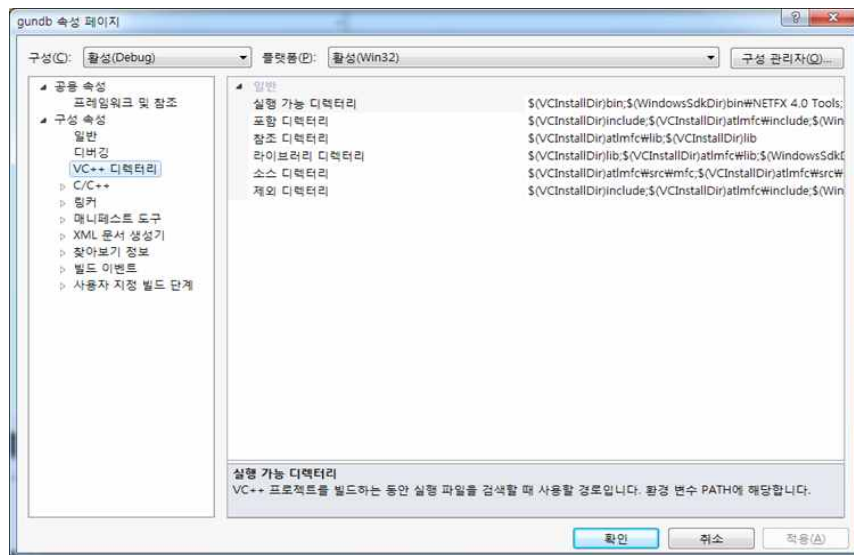
[그림 1] 급속충전기 정보 테이블 구조

또다른 테이블의 하나인 ChargerType에 저장된 내용을 읽어오기 위해서는 ODBC (Open DataBase Connectivity)를 설정하여야 하며 그 과정은 다음과 같다. 먼저 Visual Studio에서 Console Application (혹은 Windows Application) 프로젝트를 생성한다. 그 다음 [그림 2]에서 보는 바와 같이 메뉴 중 프로젝트->속성을 선택하여 프로젝트의 속성을 변경한다.



[그림 2] 프로젝트 속성 변경

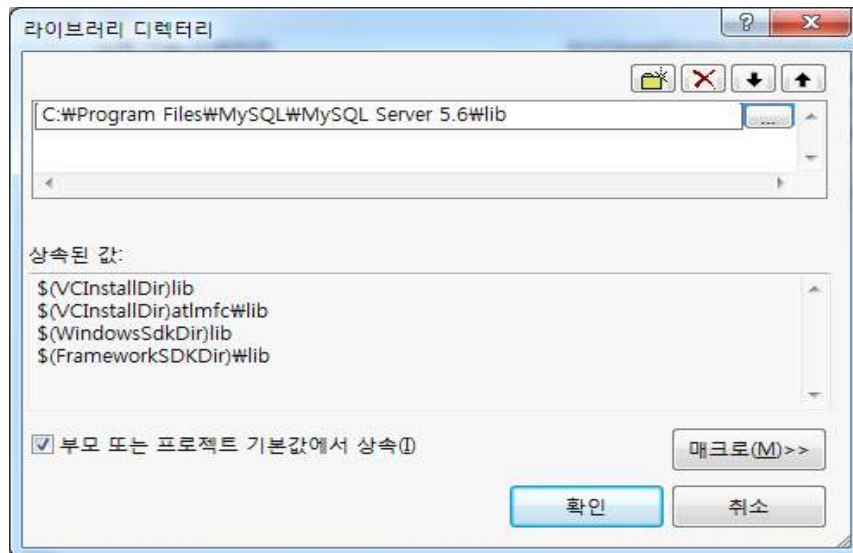
먼저 include 디렉토리의 내용을 추가하여 mysql에 관련된 헤더 파일들이 프로젝트에 포함되도록 한다. [그림 3]은 mysql이 설치된 폴더, 즉, C:\Program Files\MySQL\MySQL Server 5.6의 include 폴더를 포함시키는 과정을 보인다.



[그림 3] 헤더파일 프로젝트에 포함

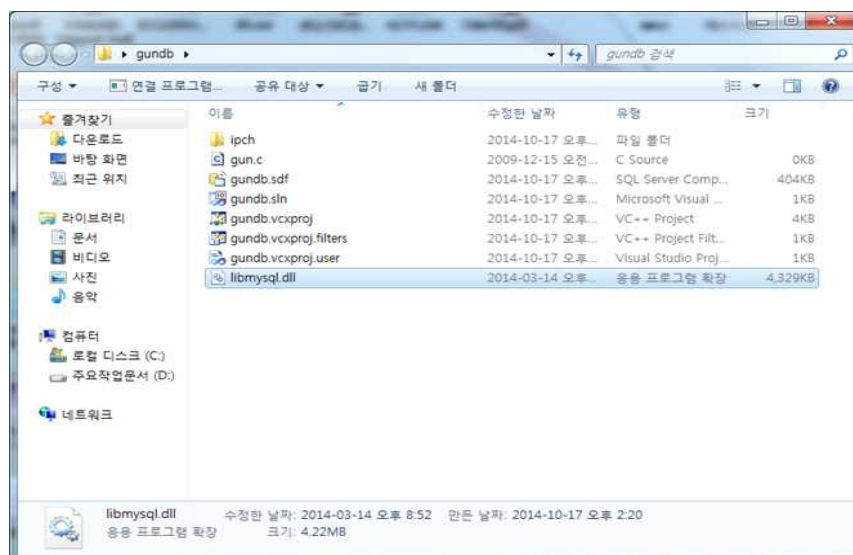
다음은 라이브러리 디렉토리의 내용을 추가하여 mysql에 관련된 헤더 파일들이 프

로젝트에 포함되도록 한다. [그림 4]는 mysql이 설치된 폴더, 즉, C:\Program Files\MySQL\MySQL Server 5.6의 lib 폴더를 포함시키는 과정을 보인다.



[그림 4] 라이브러리 디렉토리 추가

이후 C:\Program Files\MySQL\MySQL Server 5.6\lib 에 있는 libmysql.dll을 프로젝트 폴더로 복사한다.



[그림 5] 프로젝트 폴더에 libmysql.dll 추가

이상의 과정에 의해 mysql 관련 함수들을 C 언어 프로그램에서 호출할 수 있는 설정이 완성되었으며 [그림 6]에서 보는 바와 같이 SQL 문장에 의해 데이터베이스의 내용을 읽어올 수 있다.

```
#include <stdio.h>
#include <winsock2.h>
#include <mysql.h>
#include <errno.h>
#define DB_HOST "203.253.221.***"
#define DB_USER "lee"
#define DB_PASS "*****"
#define DB_NAME "lee"

int main(void) {

    MYSQL *conn;
    MYSQL_RES *res;
    MYSQL_ROW row;

    conn = mysql_init(NULL);
    if(!mysql_real_connect(conn, DB_HOST, DB_USER, DB_PASS,
DB_NAME, 3306, (char*)NULL, 0)) {
        printf("%s \n", mysql_error(conn));
        return -1;
    }
    printf("Connection OK \n");
```

```

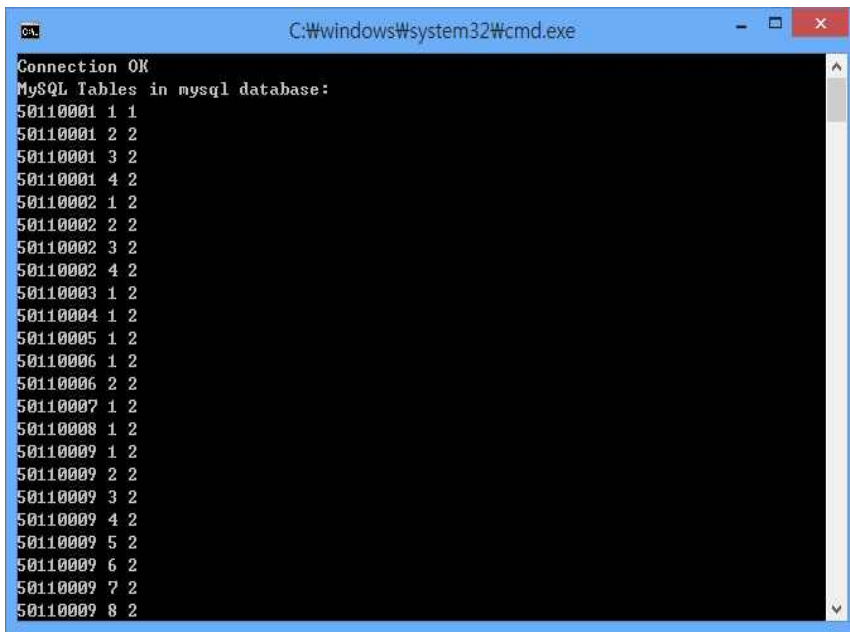
        // send SQL query
    if(mysql_query(conn, "select * from ChargerType")) {
        fprintf(stderr, "%s \n", mysql_error(conn));
        return -1;
    }
    res = mysql_use_result(conn);
    // output table name
    printf("MySQL Tables in mysql database: \n");
    while ((row = mysql_fetch_row(res)) != NULL) {
        printf("%s %s %s \n", row[0], row[1], row[2]);
    }

    mysql_close(conn);
    return 0;
}

```

[그림 6] 데이터베이스 연동 소스

[그림 7]은 이 프로그램에 의해 C 프로그램에서 데이터베이스의 내용을 읽어온 결과이며 Console project로 프로그램을 빌드하였다. 이 모듈은 응용의 목적에 따라 다른 타입의 프로젝트와 쉽게 결합할 수 있다.



[그림 7] 데이터베이스 내용을 읽어온 결과

이와 같이 InterDest 와 ToDC 테이블에 있는 내용을 읽어 와서 덤프한 내용은 각각 [그림8]와 [그림9]과 같다. 각각 86*86 2차원 배열, 86*59 2차원 배열이며, km로 환산되어 있다.

[그림 8] 관광지 인접행렬

[그림 9] 충전기 인접행렬

3. 제안 기법

관광 서비스 구현을 위하여 n 개의 관광 서비스의 목적지에 N 집합에 급속충전기를 하나 추가하여 $n+1$ 개의 전체 목적지를 구성 후 각각의 세트들을 경쟁시켜 서비스의 길이가 가장 작은 세트를 선택한다 이 과정에서 관광서비스 스케줄러는 실행 가능한 모든 스케줄을 조사한다.

여기서, m 개의 충전기가 있다고 할 때 탐색트리는 m 개의 서브트리를 갖는다, 각각의 서브트리는 $n!$ 의 리프이며 깊이가 $n+1$ 인 트리를 만든다. 리프 노드에 도달하는 다른 최적화 기술과 마찬가지로 하나의 스케줄이 완성되며 그 비용은 소정의 기준에 따라 평가한다. 만약 서비스 길이가 현재 가장 짧고 제약조건을 위반하지 않는 경우 이 스케줄이 현재의 가장 짧은 스케줄을 대체한다. 스케줄에서 먼 거리에 존재하는 충전기는 최적의 스케줄이 되기 어려움으로 제외 시켰다.

이러한 스케줄을 제거하기 위하여 탐색절차는 경로에 따라 SOC(State OF Charge : 가용 잔존량)의 변화를 추정한다[Lee and Park, 2013]. SOC는 EV(Electroc Vehicle : 전기자동차)가 이동한 거리에 따라 감소한다. SOC는 스케줄에서 지정된 순서에 따라 감소하며 0이하로 떨어질 경우 추가 노드 확장 없이 바로 스케줄에서 삭제된다. 또 각각의 노드가 순차적으로 전개되며 검색 공간을 탐색하는 과정에서 현재 확장 노드를 기점으로 하는 부분경로는 현재의 최적의 경로보다 더 클 수 있다. 이때에 경로를 가지치기함으로써 즉시 확장을 정지시킬 필요가 있다.

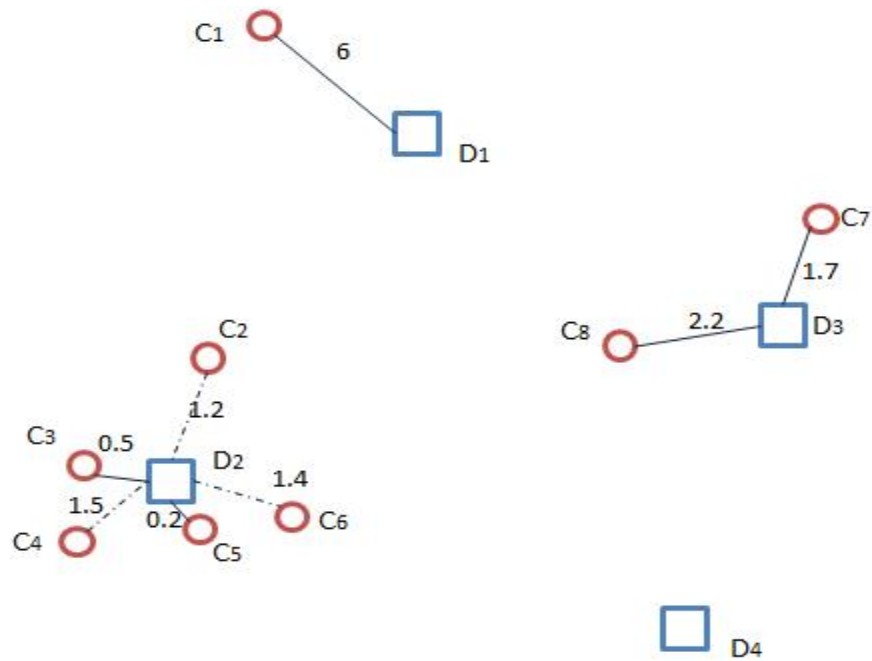
관광 일정에서 선택된 관광지의 수 n 은 대부분 10 미만이다. 반응시간이 많이 향상되었으나 충전기의 수가 많은 경우 수행시간이 오래 걸릴 수 있다. 본 논문의 주요 아이디어는 급속 충전기의 설치에 따른 m (충전기의 수)이 커지더라도 허용 가능한 수행 시간 안으로 맞추는 것이다. 스케줄링 과정에서 전체 충전기 m 중 조정 가능한 충전기 m' 만 조사하여도 작은 정확도의 손실만 있을 뿐이다.

주요 아이디어는 [그림 9]에서 설명되어진다. D1,D2,D3, D4는 전기자동차 운전자가 방문하고자 목적지이다. 여행 거리는 방문 순서에 의해 결정되며, 전기자동차는 D1에서 시작하고 D1으로 반환한다고 가정한다. C1에서 C5까지 5개의 DC충전소가 있고 전기자동차는 관광하는 동안 한번은 충전해야하기 때문에 목적지들 외에 추가로 5개의 DC충전소 중 하나를 방문해야한다. 게다가, 순수한 여행 거리는 {D1,D2,D3,D4}용 TSP 솔버에 의해 결정되며, TSP의 길이로 정의된다. 이는 일부 Ci의 첨가를 포함한 여행거리로부터 구분되어야한다.

전처리 과정은 배터리 소모량 및 목적지 2개의 목적지로부터 각각 짝지어진 각 쌍과 충전소에 대한 거리를 계산한다. 전 처리과정은 배터리 소모 및 2개의 목적지로부터 각각 짝지어진 거리의 각 쌍과 충전소에 대한 거리가 계산된다. 상호 목적지로 선정된 행렬을 기반으로 n개의 목적지를 위한 최적의 스케줄을 찾을 수 있다. 따라서, 문제는 SOC를 0으로 만들지 않고 {D1,D2,...,Dn,Ck}를 위한 일정으로부터 여행거리를 최소화 할 Ck를 찾는 것이다. 기본적으로 TSP 솔버는 최적의 스케줄을 찾기 위해 {D1,D2,..,Dn,C1}, {D1,D2,..,Dn,C2}, ... ,{D1,D2,..,Dn,Cm}과 같이 m회 호출된다. TSP 솔버는 기본적으로 O(n!)의 시간복잡도를 갖지만, m이 큰 경우 실행시간이 너무 많이 연장된다. 합리적인 품질의 m'후보(m'≪m)를 찾을 수 있다면, 검색 공간을 현저히 줄어들며, m'/m은 속도 향상 비율이다.

m' 후보를 검색하는 휴리스틱은 [그림 10]의 예시로 더 알맞게 설명할 수 있다. 여행 중에 추가적으로 충전소로 이동하게 된다면, 불가피하게 여행 거리와 시간을 증가된다. 하지만 투어 루트에 충전소를 선정한다면, 여행 거리에는 거의 영향을 미치지 않는다. 시간이 많이 걸리는 TSP 솔버가 스케줄 생성을 완료할 때까지 최종 일정에 포함되는 링크를 알수 없다. 하지만 목적지에 가까운 충전기가 선택된다면, 여행의 길이는 증가하지 않을 가능성이 높다. 그러므로 각 충전 스테이션에 가까운 목적지를 찾을 필요가 있다. 여기서 한 가지 문제점이 도출될수 있다. 만약 한 목적지에 많은 수의 충전기가 가까이 있다면 그 TSP솔버는 스케줄을 생성하지 못하거나 최적의 스케줄에 비해 낮은 효율을 보여줄 수 있다. [그림 10]은 그 예를 보이고 있는데 각 충전 스테이션에 가까운 목적지를 찾고 목적지에 따른 충전기 수를 제한을

하여 스케줄러의 효율을 향상 시켰다. 여기서, C4과 C5는 일반적으로 D1에 가장 가깝다. 모든 목적지의 스테이션으로부터 행렬 값으로 또한 기록되어 있다. 이 그림에서는, C5는 0.2를 갖고 먼저 m' 후보로 선정될 것이다. 만약 m' 가 3 라면 C5, C3, C7이 선정될 것이다. m' 는 조정가능한 인자로 응답시간과 스케줄의 질에 대한 요구사항에 따라 결정한다.



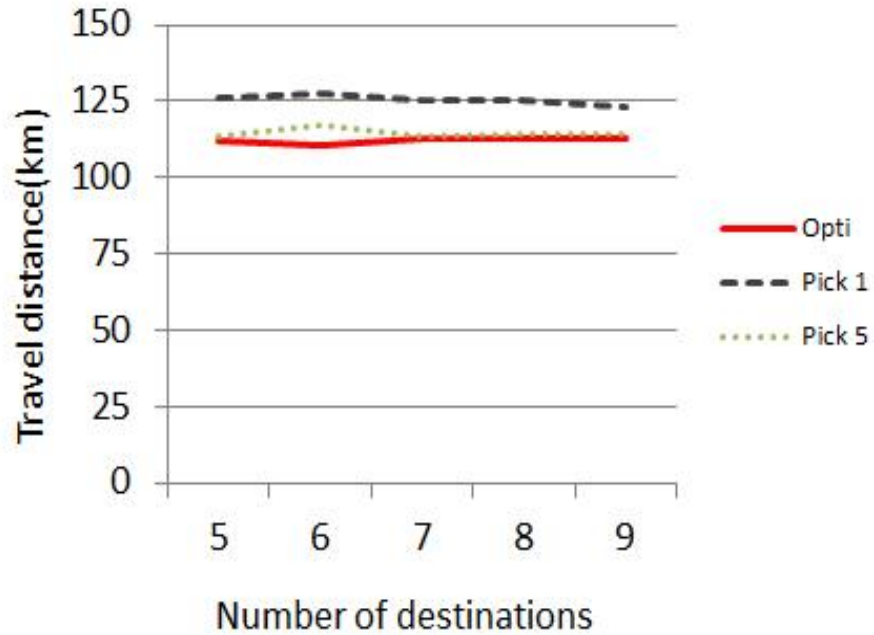
[그림 10] 충전기 제한 거리기반 휴리스틱

IV. 성능평가

본 장에서는 제안한 관광서비스 스케줄러의 프로토타입을 구현하고 그 성능을 평가한다. 최적의 스케줄러는 가능한 모든 스케줄을 조사하며, 제안한 관광서비스 스케줄러와 성능 비교를 위해 개발했다. 같은 충전기에 동일 시간에 다수의 충전요청이 발생할 경우 대기시간을 고려해야한다. 또한 m 개의 후보를 선택할 때, 전기자동차의 도착시간에 사용 중인 충전소를 피해야한다. 하지만 알고리즘의 순수한 효과에 초점을 맞추기 위하여 급속충전기 선택에 대기시간이 없다고 가정하였다. 이러한 가정은 관광서비스 스케줄에 급속충전기의 선택이 쉽게 통합될 수 있도록 한다.

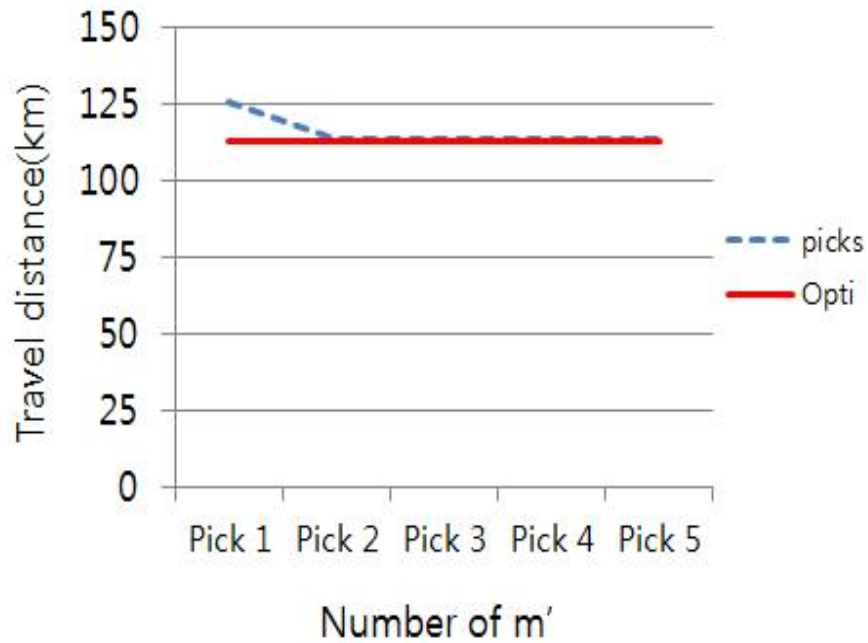
성능평가 그래프는 관광서비스 스케줄의 길이와 정확도를 중점으로 평가하였다. 여기서 정확도는 스케줄러가 최적의 순서를 발견할 확률을 의미한다. 대상인 86 개의 관광지는 그들 각각의 쌍에 대한 거리가 선형적으로 계산된다. $(n-1)$ 개의 대상을 임의로 결정하며, 출발지는 공항으로 설정하였다. 스케줄 길이에 대한 평가를 위하여 목적지 수, 순수 TSP 거리 최소 및 최대를 제한하여 일정 집합을 완성하고 이를 이용하여 최적의 스케줄과 제안 스케줄러를 비교한다. 각 파라미터의 설정을 위하여 10 개의 세트가 생성되고 그 결과를 평균한다.

첫 번째 실험은 목적지 수에 따른 이동거리를 측정하였으며, 그 결과는 [그림 11]와 [그림 12]에 나타내었다. [그림 11]와 같이 실험은 목적지의 수를 5~9로 변경하며 진행되었으며, TSP의 길이가 100km~120km 사이인 목적지들로 구성되었다. 10 개 이상의 목적지를 선택하였을 때 해당 길이의 TSP 길이를 만족하는 집합을 찾기 어렵다. [그림 11]는 3 개의 곡선을 나타내었다. 예상대로 최적의 스케줄은 전체범위에서 가장 작은 이동거리였다. 성능차이는 6에서 가장 크며 목적지의 수가 6이상 일 때 대상의 수에 따라 작아진다. 9 개의 목적지의 경우 Pick1과 Pick5는 최적의 결과보다 각각 8%와 1% 크다. 목적지 개수의 증가에 따라 스케줄러가 목적지에 가까운 후보를 찾을 가능성이 높아지는 것으로 관찰된다.



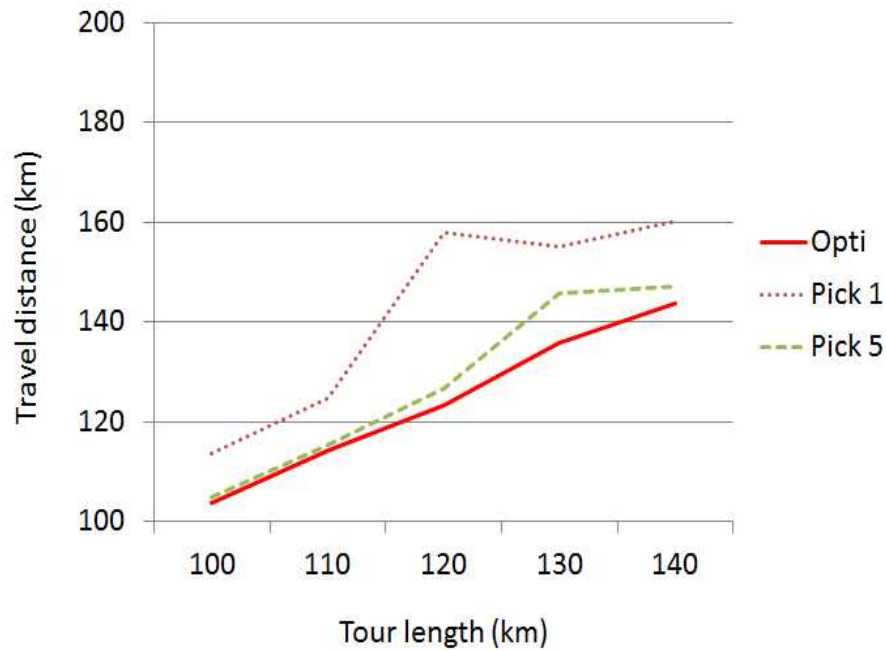
[그림 11] Effect of the number of destinations

다음 [그림 12]은 후보지 수에 따른 영향을 나타내 었다. 여기서 목적지의 수는 7로 고정하였으며 100km~120km사이의 TSP길이를 갖는 대상을 범위로 설정하여 실험하였다. 최적의 스케줄은 더 많은 후보를 통하여 제안 스케줄 보다 더 짧은 주행거리를 나타내었다. 최적의 스케줄은 모든 후보 충전기 29개를 모두 검색하지만 Pick1에서는 단지 1개를 선택하며 pick2에서는 2개를 선택한다. 이동거리는 최적의 스케줄 보다 단지 Pick1에서는 11%가 길며 Pick2에서는 1%가 길다. 스케줄은 충전기 후보가 추가 될 때마다 이동거리가 감소한다. Pick5의 경우 실행시간은 최적 방식에 비해 5/29만으로 거의 최적 방식과 같은 품질의 스케줄을 찾는다.



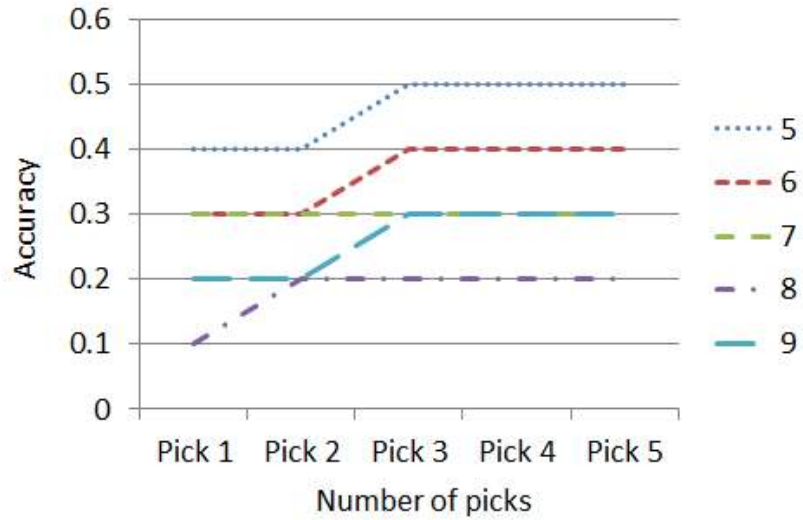
[그림 12] Effect of the number of picks

두 번째 실험은 TSP 거리 증가에 따른 여행거리를 측정한다. [그림 13]은 TSP 길이에 따른 스케줄러의 이동거리를 나타낸다. 여기서 각각의 TSP의 거리는 100km에서 110km, 140km에서 150km로 10km구간에 대하여 10개의 집합을 평균화하였다. 목적지의 수는 7 개로 고정하였으며 pick1 그래프의 TSP길이 120km이상인 경우 이전보다 많은 차이를 보인다. 100-110km의 범위에서 9%의 차이를 보이고 있으며, 120-130km의 범위에서는 가장 큰 28%의 차이를 보이고 있다. 단순히 Pick1은 목적지 중 어느 하나로부터의 거리를 고려하여 선택한다. 시작 또는 마지막 순서에 대상 급속 충전기가 선택된 경우 길을 찾지 못할 수 있다. Pick5의 경우 목적지와의 거리를 고려하여 5개의 후보 충전기들이 선택된다. [그림 13]과 같이 최적의 방식과 최소 1%, 최대 7%의 차이를 보이는 스케줄을 생성한다.



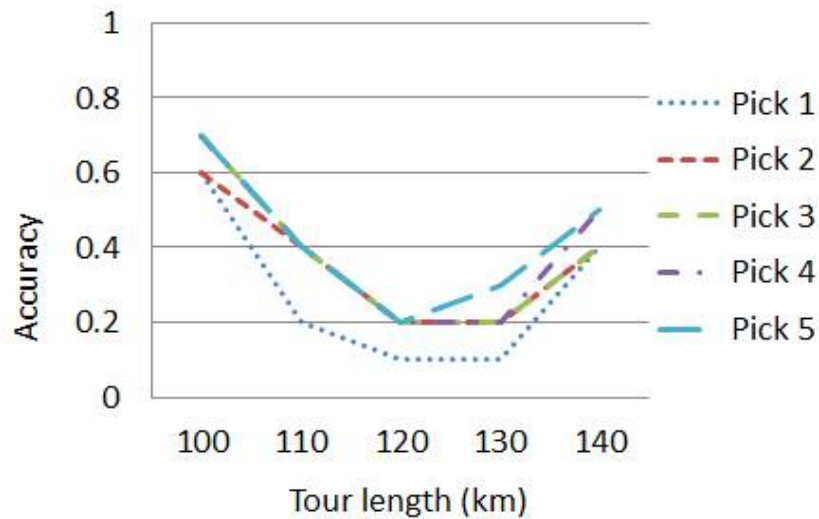
[그림 13] Effect of the TSP length

마지막 실험은 목적지 수와 TSP길이에 따른 정확성 측정이며 [그림14]와 [그림 15]에 나타내었다. [그림 14]는 Pick에 따라 최적의 스케줄과 제안 스케줄의 정확도를 나타내며, 모든 실험은 100km에서 120km범위에서 이루어졌다. 각각의 곡선은 대상 번호의 수만큼의 목적지를 의미한다. 모든 곡선이 Pick의 증가에 따라 정확도가 향상되었다. Pick5의 경우 정확성이 목적지의 수 5와 8에서 0.5, 0.2이다. 이러한 경우에도 최적의 스케줄러와 같은 스케줄을 찾는다는 실패하지만 품질은 비슷하다. 이 결과는 제안 방법이 훨씬 더 적은 비용으로 최적의 스케줄에 가까운 스케줄을 찾을 수 있음을 나타낸다. 또한 최적의 스케줄러는 많은 충전소들을 검색하여 하나씩 교체하며 스케줄을 생성하기 때문에 제안기법보다 훨씬 많은 시간이 걸릴 수 있다.



[그림 14] Effect of the number of picks

[그림15]는 TSP 길이에 따른 정확성을 나타내며 목적지의 수는 7로 설정하였다. 이 그림은 Pick의 각 번호와 관련된 5 개의 곡선으로 나타내었다. 정확도는 TSP의 길이가 증가할수록 감소하다 120km 이상에서 증가하는 경향이 나타난다. 120km 주변을 제외한다면 Pick5는 적어도 0.4이상의 정확도를 유지하고 있으며, 이 결과는 전체 TSP 길이에 적당한 품질의 스케줄을 얻을 수 있음을 확인하였다.



[그림 15] Effect of TSP length

V. 결 론

스마트그리드는 ICT 기술을 결합하여 에너지 효율화를 추구하는 차세대 전력망으로 각 전력 요소에 적합한 고급 컴퓨터 알고리즘과 통신 기술에 의해 계속 지능화하고 있다. 전기자동차는 그리드에서 충전이 되기 때문에 운송부분마저도 전력망의 일부로 만들고 있으며 전기자동차는 주행거리와 충전시간의 약점 때문에 양방향 상호작용에 기반한 효율적인 컴퓨터 알고리즘을 필요로 한다. 특히 일일 이동거리가 주행가능거리를 넘을 수 있는 택시와 관광용 렌터카에서는 보다 지능적인 경로 탐색 기능이 중요하다.

본 논문에서는 다수의 목적지를 방문하는 관광용 렌터카를 주요 대상으로 관광-충전 스케줄을 작성하는 기법을 설계하고 성능을 평가하였다. 제주도와 같이 충전인프라를 구축하고 있는 스마트그리드 시티에서는 급속 충전기들이 여러 곳에 산재해 있으며 렌터카 기반 관광에 있어서는 심야 완속 충전 이외에 관광 중 하루 한번 정도의 급속충전이 필요하다. 방문하고자하는 관광지들에 따라 어떤 급속충전에서 충전하는가가 배터리 소모와 관광시간에 영향을 주는데 이를 결정하기 위해서는 상당한 계산시간이 소모된다.

본 논문에서 설계된 기법은 주어진 관광지 집합에 대해 모든 충전기를 고려대상으로 간주하지 않고 최종 스케줄에 포함될 가능성이 높은 충전기들만을 선택적으로 계산에 포함시킨다. 선택된 관광지 집합에 대해 각 충전기마다 가장 가까운 관광지와 거리를 계산한 후 이 거리가 가장 짧은 일부의 충전기들만을 선택한 후 하나씩 관광지 집합과 결합하여 TSP 거리를 계산한다. 이후 가장 TSP 거리가 짧은 관광-충전 스케줄이 최종적으로 선택이 된다. 이 과정에서 한 관광지에 바운드된 충전기의 개수를 제한함으로써 서치 스페이스가 한두 관광지에 밀집되는 현상을 완화한다.

이러한 서비스를 구축하기 위해 Linux 운영체제 위에 Mysql 데이터베이스를 설치하고 충전기와 관광지 정보들을 관계형 테이블로 저장하였으며 추후 이 정보들이 변경되는 경우에도 데이터베이스를 통해 응용들에 쉽게 파급될 수 있다. 특히 Mysql의 공간정보 처리 기능을 결합하기 위해 공간객체로 위치정보들을 명세하였다.

성능 평가는 대한민국 제주시의 도로망과 충전기 및 관광지의 지리 분포를 대상으로 수행하였다. 최적의 스케줄과 평가 시 투어 길이의 차이는 1% 증가하였으며 0.7의 확률로 최적의 스케줄을 발견하였으며, 5/29의 속도향상을 불러왔다. 다음으로 TSP길이 변화에 대하여 최적의 스케줄과 비교하였으며 거리의 차이는 최대 7%가 난다. 결국, 본 논문의 스케줄러는 빠른 응답시간에 최적의 방식과 유사한 성능의 스케줄을 생성하기 때문에 사용자의 편리성과 서비스의 실용성을 지원할 수 있다.

결론적으로, 본 논문에서 설계하고 프로토타입을 제작한 전기자동차 렌터카용 투어-충전 스케줄러는 전기자동차 사용자 측의 배터리 방전에 대한 우려를 최소화하여 렌터카에서도 전기자동차가 보급될 수 있는 기반을 제공한다. 또 이 서비스 구조를 바탕으로 신규 서비스들이 속속 도입되어 각 차량의 목적에 맞는 서비스들의 개발을 촉진할 것으로 기대된다.

VI. 참고문헌

- [Timpner and Wolf, 2014] Timpner, J., Wolf, L. 2014. Design and evaluation of charging station scheduling strategies for electric vehicles. IEEE Transactions on Intelligent Transportation Systems 15(2), 579 - 588
- [Mischinger et al. 2012] Mischinger, S., Hennings, W., Strunz, K. 2012. Integration of surplus wind energy by controlled charging of electric vehicles. In: 3rd IEEE PES Innovative Smart Grid Technologies Europe
- [박광칠. 2012] 박광칠. 2012.10. 전기자동차의 단점에 이해와 개선전망
- [Botsford and Szczepanek. 2009] Botsford, C., Szczepanek, A. 2009. Fast charging vs. slow charging: Pros and cons for the new age of electric vehicles. In: International Battery Hybrid Fuel Cell Electric Vehicle Symposium
- [Veneri et al. 2013] Veneri, O., Capasso, C., Ferraro, L., Pizzo, A. 2013. Performance analysis on a power architecture for EV ultra-fast charging stations. In: International Conference on Clean Electrical Power, pp. 183 - 188
- [Lee and Park. 2015] Lee, J., Park, G.-L. 2015 Dual battery management for renewable energy integration in EV charging stations NeuroComputing. In press.
- [교통안전공단. 2010] 교통안전공단, 2010.12. 2009년도 자동차 주행거리 실태분석 연구
- [Shim et al. 2012] Shim, V., Tan, K., Tan, K. 2012. A hybrid estimation of distribution algorithm for solving the multi-objective multiple traveling salesman problem. In: IEEE World Congress on Computational Intelligence

- [Lee and Park, 2013] Lee, J., Park, G.-L. 2013. Design of a multi-day tour-and-charging scheduler for electric vehicles. In: Ramanna, S., Lingras, P., Sombatheera, C., Krishna, A. (eds.) MIWAI 2013. LNCS, vol. 8271, pp. 108 - 118. Springer, Heidelberg
- [지능형전력망협회 and (주)한국전력신문사, 2012] 지능형전력망협회,(주)한국전력신문사, 2012.12. 2013 대한민국 스마트그리드 연감
- [김정구 and 이도경, 2009] 김정구, 이도경, 2009.12. 도로교통량 통계연보를 활용한 도로부문 온실가스 배출량 추정
- [제주특별자치도, 2012] 제주특별자치도, 2012.5. 제주형 저탄소 녹색성장, Carbon Free Island Jeju by 2030
- [Li and Sun. 2009] Li, D., and Sun, H.X., 2009. An Application Research of TSP Based on Genetic Algorithm, Science Technology of Heilongjiang Province, no. 13, pp. 27
- [Lin and Kernighan, 1971] Lin, S. and Kernighan, B.W. 1971. An Effective Heuristic Algorithm for the Traveling Salesman Problem, Bell Telephone Laboratories, Incorporated, Murray Hill, N.J.
- [Garey and Johnson, 1979] Garey, M.R. and Johnson, D.S.1979. A Guide to the Theory of NP-Completeness, Computers and Intractability, Freeman
- [Bianchi et al. 2008] Bianchi, L., Dorigo, M., Gambarella, L.M., and Gutjahr, W.J., 2008. A survey on metaheuristics for Stochastic combinatorial optimization,” Natural Computing
- [Freisleben and Merz, 1996] Freisleben, B. and Merz, P., 1996. Genetic local search algorithm for solving symmetric and asymmetric traveling salesman problems,” Proceedings of IEEE International Conference of Evolutionary Computation, IEEE-EC96, IEEE Press, pp.616-621
- [Goldberg and Kuo, 1987] Goldberg, D. E., and Kuo, C. H. 1987. Genetic Algorithms in Pipeline Optimization, Journal of Computing in Civil Engineering, ASCE, 1(2), 128-141.
- [Gambardella and Dorigo, 1997] Gambardella, L.M, and Dorigo, M., 1997. Ant

Colony System: A cooperative Learning Approach to the Traveling Salesman Problem, IEEE Transactions on Evolutionary Computation, Vol.1, No.1

[Dorigo and Blum, 2008] Dorigo, M. and Blum. C., 2005. Ant colony optimization theory: A survey, Theoretical Computer Science, 344(2-3), pp.243-278

[Zhong et all. 2006] Zhong, Y., Yang, X., Ning, z.. 2006. A Discrete Particle Swarm Optimization Algorithm for Solving Traveling salesman Problem, System Engineering and Theory Practice, pp.88-94,

[James and Russell, 1995] James K. and Russell E. 1995, "Particle Swarm Optimixation", Proceedings of the 1995 IEEE Conference on Neural Networks, 4, pp 1942-1948.

[Goldberg, 1989] Goldberg, D.E.. 1989. Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley

[23..] Raed Abu Zitar, Huessein Hiyassat. 2004 Optimizing the parameters of Ant colony Algorithm Using the Genetics," Enformatika Transaction on Engineering, Computing and Technology VI, December, pp.228-231

[Shi et al. 1993] Shi, X.H., Liang, Y.C., Lee, H.P., Lu, C., and Wang, L.M. 1993 An improved GA and novel PSO-GA-based hybrid algorithm, Information Processing Letters

[Kondo and Watanabe, 2011] Kondo, F., Watanabe, T., 2011 A study on distributed parameter free genetic algorithm for TSP problem for TSP problem", 2011 IEEE International Conference on Systems, Man, and Cybernetics (SMC), pp.675-680

[Matei and Pop, 2010] Matei, O., Pop, P., 2010 "An efficient genetic algorithm for solving the generalized traveling salesman problem", 2010 IEEE International Conference on Intelligent Computer Communication and Processing (ICCP), pp.87-92

[von Poser and Awad, 2006] von Poser, I., Awad, A.R., 2006 Optimal Routing

- for Solid Waste Collection in Cities by using Real Genetic Algorithm”, Information and Communication Technologies, ICTTA '06, pp.221-226
- [Li et al. 2011] Li, X., Liao, J., Cai, M., 2011 Ant colony algorithm for large scale TSP”, 2011 International Conference on Electrical and Control Engineering (ICECE), pp.573-576
- [Li et al. 2008] Li, L., Ju, S., Zhang, Y., 2008 Improved Ant Colony Optimization for the Traveling Salesman Problem”, 2008 International Conference on Intelligent Computation Technology and Automation (ICICTA), pp.76-80
- [Hara et al. 2010] Hara, A., Matsushima, S. Ichimura, T., Takahama, T., 2010. Ant Colony Optimization using exploratory ants for constructing partial solutions”, 2010 IEEE Congress on Evolutionary Computation (CEC), pp.1-7
- [Liu and Huang, 2010] Liu, Z., Huang, L., 2010 A mixed discrete particle swarm optimization for TSP”, 2010 3rd International Conference on Advanced Computer Theory and Engineering
- [Akhand et al. 2012] Akhand, M.A.H., Akter, S., Sazzadur Rahman, S., Hafizur Rahman, M.M., 2012 Particle Swarm Optimization with partial search to solve Traveling Salesman Problem”, 2012 International Conference on Computer and Communication Engineering (ICCCE), pp.118-121
- [Chen et al. 2010] Chen, R., Yan, A., Lin, C., 2010 Tour route planning multimedia system with particle swarm optimization designed, 2010 4th International Conference on New Trends in Information Science and Service Science (NISS), pp.725-730
- [KB금융지주 경영연구소, 2013] KB금융지주 경영연구소. 2013.11. 친환경 자동차 성장 전망 및 유망 분야
- [제주관광공사, 2013] 제주관광공사, 2013.1. 제주관광시장동향분석