



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](#)

博士學位論文

# Power Consumption Prediction and Scheduling Schemes for Microgrids

濟州大學校 大學院

電算統計學科

金 惠 珍

2017年 2月

# Power Consumption Prediction and Scheduling Schemes for Microgrids

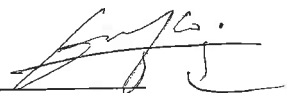
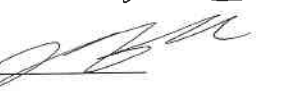
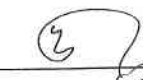

指導教授 朴 景 麟

金 惠 珍

이 論文을 理學 博士學位 論文으로 提出함

2016年 12月

金惠珍의 理學 博士學位 論文을 認准함

審査委員長	이 경훈	
委員	곽 경권	
委員	박 찬정	박찬정
委員	홍 준호	
委員	고 봉우	

濟州大學校 大學院

2016年 12月

# Power Consumption Prediction Scheduling Schemes for Microgrids

Hye-Jin Kim

(Supervised by professor Gyung-Leen Park)

A thesis submitted in partial fulfillment of the requirement for the  
degree of Doctor of Philosophy

2017. 2.

Department of Computer Science and Statistics

GRADUATE SCHOOL

JEJU NATIONAL UNIVERSITY

# Contents

<b>Abstract</b> .....	<b>vii</b>
<b>I. 서론</b> .....	<b>1</b>
1. 연구 동기 .....	1
2. 연구 목적 .....	4
<b>II. 연구의 배경 및 관련 연구</b> .....	<b>6</b>
1. 수요반응 .....	6
2. 제약만족 최적화 문제(CSOP)의 접근 .....	10
1) 백트래킹 .....	10
2) 백트래킹 기반 전기 자동차 충전 대기 시간을 최소화하기 위한 관광스 스케줄링 기법 .....	11
3) 유전 알고리즘 .....	16
4) 유전 알고리즘을 이용한 전기 자동차 재배치 스케줄링 기법 .....	25
3. 전력 수요 예측 .....	29
<b>III. 전력 수요 예측</b> .....	<b>32</b>
1. 전력 사용량의 월별 예측 .....	33
2. 전력 사용량의 15분별 1일 예측 .....	38
3. 전력 사용량의 15분별 70일 예측 .....	43
<b>IV. 백트래킹 기반의 전력 소비 스케줄링 기법</b> .....	<b>55</b>
1. 태스크 모델 .....	55
2. 제안 기법 .....	61

3. 실험 및 결과 .....	66
1) 피크 부하 감소와 수행 시간에 대한 실험 .....	67
2) 전역(global) 피크의 효과에 대한 실험 .....	69
<b>V. 유전 알고리즘 기반의 멀티쓰레드 전력 소비 스케줄링 기법 .....</b>	<b>74</b>
1. 제안 기법 .....	74
2. 실험 및 결과 .....	78
<b>VI. 결론 .....</b>	<b>84</b>
<b>VII. References .....</b>	<b>87</b>

## List of Tables

Table 1. 인센티브 기반의 수요반응 종류 .....	8
Table 2. Relocation weights .....	25
Table 3. 2015년도 월별 전력 수요량 (단위: kwh) .....	36
Table 4. 2016년 6월 1일 전력 수요량 예측 (단위: kwh) .....	40
Table 5. 전력 수요 예측 오차 (15분별) .....	43
Table 6. 전력 수요의 실제값과 예측값의 차(70일치 15분별) .....	46
Table 7. 최대 전력 수요의 실제값과 예측값의 차 (70일치 15분별) .....	47
Table 8. 평균 전력 수요의 실제값과 예측값의 차 (70일치 15분별) .....	48
Table 9. 최대 전력 수요의 실제값과 예측값의 차 (70일치 요일별) .....	49
Table 10. 평균 전력 수요의 실제값과 예측값의 차 (70일치 요일별) .....	50
Table 11. 최대 전력 수요의 예측 오차 (70일치 15분별) .....	51
Table 12. 평균 전력 수요의 예측 오차 (70일치 15분별) .....	52
Table 13. 최대 전력 수요의 예측 오차 (70일치 요일별) .....	53
Table 14. 평균 전력 수요의 예측 오차 (70일치 요일별) .....	54
Table 15. 스케줄링에 사용될 태스크의 분류 .....	56

## List of Figures

Figure 1. 인 홈 네트워크와 에너지 저장 시스템(EMS) .....	3
Figure 2. 제주대학교 시간대별 최대수요 (2015.07 ~ 2015.08) .....	6
Figure 3. 수요반응으로 인한 전력 시장 가격의 영향 .....	7
Figure 4. 전기 자동차의 관광 스케줄 모델 .....	11
Figure 5. 대기 시간의 추정 .....	13
Figure 6. TSP와 전기 자동차 스케줄러의 비교 .....	15
Figure 7. 유전 알고리즘의 순서도 .....	17
Figure 8. 유전 알고리즘의 과정 .....	17
Figure 9. 전력 데이터의 인코딩 예시 .....	18
Figure 10. 유전 알고리즘의 교차 연산 .....	21
Figure 11. 전기 자동차-충전소 매칭 .....	26
Figure 12. Encoding scheme .....	27
Figure 13. ARIMA 구축의 흐름도 .....	30
Figure 14. 한국전력의 iSMART가 제공하는 전력 수요 데이터 .....	32
Figure 15. 2008년 1월부터 2014년 12월까지의 월별 데이터 .....	33
Figure 16. auto.arima() 함수의 결과 .....	34
Figure 17. Diagnostic Display for the ARIMA(0, 1, 1) .....	35
Figure 18. 2015년도 월별 전력 수요량 예측 결과 그래프 .....	36
Figure 19. forecast() 함수를 통한 예측 결과 그래프 .....	37
Figure 20. forecast() 함수를 통한 예측값 .....	37
Figure 21. 2016년 4월 1일부터 2016년 6월 1일까지의 전력 사용량 .....	38
Figure 22. 모형 구축을 위한 코드 .....	39
Figure 23. 15분 후 전력 수요 예측 오차 그래프 .....	41
Figure 24. 15분별 1일치 전력 수요 예측 .....	42



Figure 25. 15분별 1일치 전력 수요 예측 오차 .....	42
Figure 26. 70일치 15분 후 전력 수요 예측 .....	44
Figure 27. 70일치 1시간 후 전력 수요 예측 .....	44
Figure 28. 70일치 2시간 후 전력 수요 예측 .....	45
Figure 29. 70일치 3시간 후 전력 수요 예측 .....	45
Figure 30. 70일 최대 전력 수요 예측 .....	46
Figure 31. 70일 평균 전력 수요 예측 .....	47
Figure 32. 요일별 최대 전력 수요 예측 .....	48
Figure 33. 요일별 평균 전력 수요 예측 .....	49
Figure 34. 일별 최대 전력 예측 오차 .....	50
Figure 35. 일별 평균 전력 예측 오차 .....	51
Figure 36. 요일별 최대 전력 예측 오차 .....	52
Figure 37. 요일별 평균 전력 예측 오차 .....	53
Figure 38. 스케줄링 전의 전력 소비 프로파일 .....	57
Figure 39. 전력 소비 프로파일에 대한 슬롯 할당 테이블 .....	58
Figure 40. 스케줄링 후의 슬롯 할당 테이블 .....	59
Figure 41. 백트래킹 기반의 전력 소비 스케줄링 .....	61
Figure 42. 스케줄링을 위한 Search Tree .....	63
Figure 43. 전력 소비 스케줄링 알고리즘 .....	64
Figure 44. 스케줄링 후의 전력 소비 프로파일 .....	65
Figure 45. 태스크 수에 따른 최대 전력 요구량 비교 .....	67
Figure 46. 태스크 수에 따른 수행 시간 비교 : 1개의 선점형 태스크 .....	68
Figure 47. 태스크 수에 따른 수행 시간 비교 : 2개의 선점형 태스크 .....	69
Figure 48. 로컬 스케줄링 수에 따른 평균 피크 비교 .....	71
Figure 49. 로컬 스케줄링 수에 따른 최대 피크 비교 .....	72
Figure 50. 태스크 수에 따른 최대 전력 피크의 비교 (5 units) .....	73
Figure 51. 전력 소비 프로파일에 대한 유전 알고리즘 탐색체 .....	75

Figure 52. 유전 알고리즘을 이용한 멀티 쓰레드 전력 소비 스케줄링 .....	77
Figure 53. 피크로드에 대한 태스크 수의 영향 .....	79
Figure 54. 피크로드에 대한 해 집단의 크기의 영향 .....	80
Figure 55. 피크로드에 대한 iteration의 영향 .....	81
Figure 56. 피크로드에 대한 쓰레드 수의 영향 .....	82
Figure 57. 피크로드에 대한 수행 길이의 영향 .....	83
Figure 58. 제안된 기법의 서비스 구조 .....	86

## Abstract

스마트그리드에서 가장 주목하는 것은 전력 사용의 피크를 평준화하는 것이다 [Mady et al. 2010]. 그만큼 피크 시 전력 사용이 집중되면 전력 계통은 위협을 받게 되고, 이로 인해 발생하는 블랙아웃은 국가 경제에도 큰 손실을 가져온다. 따라서 안정적인 전력 계통을 운용하기 위해 전력 수요에 대한 보다 정확한 예측을 통해 전력 수급 계획을 세우고, 전력 사용량을 분산시켜 피크를 감소할 수 있어야 한다. 따라서 본 논문은 마이크로그리드에서의 전력 수요 예측과 특정 시간대에 전력 기기들의 사용 순서를 조정하는 효율적인 전력 스케줄링 기법들을 제안하고 그 성능을 분석한다. 논문은 크게 세 개의 주제, 즉 전력 소비 수요에 대한 예측, 백트래킹 기반의 전력 소비 스케줄링, 유전 알고리즘 기반의 멀티 쓰레드 전력 소비 스케줄링으로 구성된다.

첫 번째 파트에서는 제주대학교의 전력 사용 데이터를 토대로 자기회귀 이동평균(ARIMA; Auto Regressive Integrated Moving Average) 모형을 사용하여 월별, 15분별 전력 사용량을 예측하였다. 또한 15분별 예측은 1일치 뿐만 아니라 70일치까지의 전력 사용량 예측과 요일별 예측을 함께 진행하였으며, 예측 오차를 구한 후 모형의 성능을 평가하였다. 예측의 범위가 넓어질수록 예측 오차는 커지기는 하지만 예측값은 실제값과 상당히 일치한다.

두 번째 파트에서는 실질적인 전력 부하 프로파일을 기반으로 태스크 모델을 생성하고 전력 소비 스케줄러를 설계한다. 전력 소비 스케줄러는 태스크들을  $M \times N$ 의 할당 테이블로 할당하는데 비선점형의 태스크는 프로파일 엔트리는 단지 할당 테이블로 하나씩 차례차례 복사하고, 반면 선점형 태스크는 프로파일이나 바로 복사되는 게 아니라 조합 벡터들을 이용하면서 할당 테이블로 사상(mapping)된다. 슬롯 당 각 태스크의 총합은 스케줄이 되어지는 전력 수요량이 되며, 총합 중 가장 높은 값인 전력 소비 스케줄의 피크로드를 줄일 수 있도록

각각의 타임 슬롯에서 전체 소비량을 고려하여 스케줄링한다. 백트래킹 기반의 전력 소비 스케줄링 기법은 전력 소비 동작을 태스크 모델로 변환하여 태스크 셋의 가능한 모든 할당을 순회하기 위해 서치 스페이스를 반복적으로 확장해 나가며 피크로드를 줄일 수 있는 스케줄을 찾는다. 확장해 가는 동안 하나의 단말 노드에 도달했을 때 계산되어진 이 스케줄의 피크 소비량과 현재의 최적의 피크로드의 값과 비교해가며 현재 최적 피크로드를 업데이트한다. 불필요한 확장을 줄여 스케줄링의 속도를 향상시키기 위해 부분적인 솔루션에 해당하는 각각의 중간 노드에서 이미 최적 피크값을 넘었는지를 체크하고, 그렇지 않으면 남은 서브 트리는 제거한다. 또한 선점형 태스크의 경우는 가능한 조합을 미리 구해 놓고 스케줄링함으로써 수행 시간을 대폭 감소시킬 수 있다. 성능 평가 결과 제안된 기법은 스케줄링 전보다 피크 부하를 23.1%까지 줄일 수 있었고, 서치 스페이스 가지치기 효과에 따라 수행 시간이 단축되었다.

세 번째 파트에서는 유전 알고리즘을 이용하여 멀티 쓰레드 전력 소비 스케줄링 기법을 제안한다. 유전 알고리즘을 이용한 전력 소비 스케줄링 기법은 염색체를 정수의 엘리먼트를 포함하는 벡터로 구성하고, 이 벡터들로 구성된 해 집단이 생성되면 이 해 집단에 대해 선택, 교차, 대치 연산을 이용하여 새로운 자식 해와 다음 세대를 생성할 수 있다. 세대를 거듭하면서 반복하면 점점 최적의 해에 가까워지게 된다. 전력 소비 스케줄의 피크로드에 대한 평가를 위해 적합도 함수는 슬롯 당 전력 소비의 합을 계산하고 그 중 가장 큰 값 피크로드를 찾는다.

유전 알고리즘 기반의 멀티 쓰레드 전력 소비 스케줄링은 프로파일의 정보들을 만족할 수 있는 스케줄링의 결과들의 조합에 해당하는 해 집단을 생성할 수 있고, 이로부터 랜덤하게 초기 해 집단을 선택하게 되는데, 이 때 초기 해 집단을 다양하게 한 후 각각의 쓰레드에 그 해 집단을 할당하게 하여 각각의 쓰레드가 유전 알고리즘 프로시저를 독립적으로 작동하게 한다. 상이한 초기해 집합으로부터 각각 진화하게 함으로써 솔루션의 다양성을 만족 시키고, 멀티 쓰레드를 통해 스케줄러의 수행 시간을 단축시킬 수 있다. 실험 결과 태스크의 수가 적을수록, 해 집단의 크기가 작을수록, 유전 알고리즘의 반복 횟수(iteration)의 수가 1,000 회 정도일 때 멀티 쓰레드는 매우 효과적으로 작동한다.

# I. 서론

## 1. 연구 동기

글로벌 에너지 위기와 온실가스 배출에 의한 대기 오염이 심해지고 있어 이를 극복하고자 하는 관심이 증가하고 다양한 방안들이 제시되고 있다 [Ipakchi et al. 2009]. 특히 기후변화로 인한 피크타임의 전력 소비 증가로 나타난 대규모의 정전사태는 기존의 전력망의 한계를 인식하게 하며 발전, 송전, 배전 등의 전력 시설의 현대화를 요구하였다. 이에 따라 스마트그리드라 불리는 차세대 전력망이 등장하였고, 이로써 기존의 전력망은 정보통신기술(ICTs)과 융합하여 전력시설과 통신, 인터넷, 전력 기기들이 서로 유기적으로 복잡하게 연결되는 하나의 상호 네트워크 형태로 진화하고 있다 [Gellings 2009].

스마트그리드는 원래 노후된 전력시설 교체에 대한 논의를 시작으로 북미와 유럽 등지에서 시작되었다. 그러나 이제 스마트그리드는 대부분의 나라에서 많은 관심을 기울이고 있으며, 에너지 산업을 비롯한 스마트그리드 관련 산업 육성에 앞장서고 있다. 우리나라의 경우도 에너지 수입 의존도가 크지만 전력 요금이 낮아 전력 소비가 지속적으로 증가하고 있어 에너지 효율 향상이 요구되고 있다. 우리나라는 특히 2009년 G8 정상회의에서 이탈리아와 함께 스마트그리드 선도국가로 지정되면서부터 스마트그리드에 대한 관심이 더욱 높아졌다. 이에 정부는 먼저 스마트그리드 국가 로드맵을 발표하고 제주도에는 스마트그리드 실증단지를 구축하였다. 제주의 스마트그리드 실증 사업을 통해 KEPCO, KT등의 많은 참여 기업들이 컨소시엄을 이루어 스마트그리드를 위한 인프라를 조성하고 스마트그리드 모델을 연구 개발하며 그에 따른 다양한 실증을 하였다. 뿐만 아니라 전력 공급을 안정화시키고 신재생 에너지와 전기 자동차 관련 사업을 육성을 통해 제주를 탄소 없는 섬(Carbon Free Island)으로 만들겠다는 계획을 추진하고 있다. 더 나아가 스마트그리드 국가 로드맵의 5대 분야 추진을 통해 2030년까지

전국토를 스마트그리드화하기 위한 다양한 노력을 하고 있다.

스마트그리드는 시맨틱 웹, 웹서비스, 지능형 에이전트, 실시간 통신 등의 다양한 IT 기술을 접목하여 에너지 효율을 최적화하는 미래의 전력망으로 공급자와 소비자 간의 양방향 통신을 통해 소비자의 지능적인 전력소비를 가능하게 하는 기술이다. IT가 결합되기 전 기존의 전력망은 화석 연료 위주의 중앙 집중식 발전으로 전력 생산 및 소비 정보들이 공급자에서 소비자로 가는 단방향(one-way) 전력 시스템 체계였다. 따라서 기존의 전력망은 서로간의 양방향 통신이 아니기 때문에 당연히 전력 수요를 실시간으로 예측하기 힘들 수밖에 없으며 전력 사용에 대한 실시간의 정보를 얻는 데에 한계가 있어 소비자의 참여 또한 수동적으로 만든다.

전력망과 IT의 결합은 일차적으로 AMI(Automated Metering Infrastructure) 구축을 가능하게 하였는데 이 기술에 의해 기존의 전력 시스템보다 에너지의 효율적인 생산 및 송전, 배전 등이 더욱 지능적으로 관리 및 운영된다. AMI는 스마트미터에서 측정된 많은 다양한 데이터들을 원격 검침기에서 측정하여 자동적으로 전력 사용에 대한 분석을 해주는 기술이다. 따라서 이러한 AMI 기술을 통해 가정과 빌딩의 전력 소비를 지능적이고 효율적이게 제어할 수 있다.

인 홈 네트워크(in-home network)는 전력 조절 시스템, IHD(In-Home Display), 홈 게이트웨이, PC 등과 같은 여러 구성 요소들을 포함한 스마트 홈 시스템이다. 인 홈 네트워크 내에서 구성 요소들은 서로 연결되어 있기 때문에 데이터 또는 제어 신호들이 구성 요소들 사이에서 송·수신 된다. 인 홈 네트워크는 HAN(Home Area Network)과 WAN(Wide Area Network)과 같은 다양한 네트워크 인터페이스를 사용할 수 있다 [M. Jujano-Rojas et al. 2012].

Figure 1은 가정에서 각 전력 기기들이 스마트미터 등의 기술에 의해 모니터링되고 제어 구동, 중지, 재개, 종료되는 예를 보인다. 스마트미터는 PLC(Power Line Communication) [Yousuf et al. 2007], ZigBee [Gislason 2008], WiFi와 같은 통신 기술을 통해 전력 사용량을 그리드로 실시간 전송하고, 소비자는 IHD를 통해 실시간 모니터링할 수 있다. 뿐만 아니라 제어기(controller)는 인터넷과 셀룰러 네트워크와 같은 WAN을 통해 전력회사(utility company)와 실시간 변화하는 가격 정보와 전력 공급에 대한 정보들을 교환할 수 있다. 전력회사는 현재 전력망

의 부하 상태에 대한 정보를 소비자에게 전달함으로써 전력 부하 조절, 가격 조정, 수요반응(DR; Demand Response)기술을 활성화 시킬 수 있다. 즉, 각각의 전력 기기는 전력 요금이 낮을 때는 표준 모드 상태를 유지하다가 전력 요금이 높을 때는 저 전력 모드 상태로 바꾸며 전력 소비를 조절할 수 있게 된다.

이처럼 전력 기기는 HAN과 같은 통신 네트워크 내에서 실시간 전력 소비량을 지능적으로 조절함으로써 피크 시 전력망의 부하를 낮추어 전력 수요를 셰이핑할 수 있다. 결국 소비자에게는 피크 요금을 피해 전기요금의 절감을 가져올 수 있으며, 국가적으로는 발전소 건축을 억제하는 효과를 갖는다.

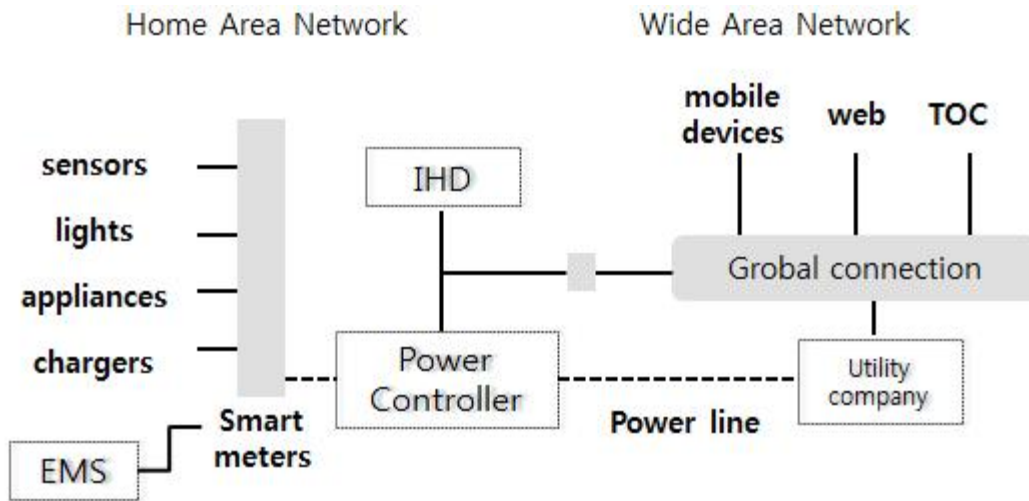


Figure 1. 인 홈 네트워크와 에너지 저장 시스템(EMS)

또한 소비자들은 네트워크를 통해 전력 시장의 수많은 공급자들 중 선호하는 공급자를 선택하여 전력을 구입할 수도 있다. 소비 후 남은 전력은 EMS(Energy Management System)에 저장되었다가 다시 전력 시장에 되팔 수도 있기 때문에 궁극적으로는 효율적인 에너지 사용과 관리가 가능하다 [Lin et al. 2010]. 이는 V2G(Vehicle-to-Grid) 기술로서 지속적으로 상용화 방안이 강구되고 있다. 뿐만 아니라 가정과 빌딩에서도 개별적으로 태양광 시설과 풍력 발전기와 같은 전력 생산 시설을 보유하여 신재생에너지를 생산함으로써 전체적인 전력 소비량을 감

소시키고 비상시 전원으로도 사용할 수 있다 [Lee et al. 2012i].

## 2. 연구 목적

특정 시간에 동시에 많은 사용자들의 전력 사용을 요구하게 되면 전력망의 피크 부하 상태가 발생하게 된다. 현재 전력 사용량은 이상 기후로 인하여 점점 증가하고 있고, 향후 많은 가정의 전기자동차 보급으로 인해 충전에 사용되는 전력이 집중됨에 따라 피크 부하는 더욱 더 심각해질 수 있게 된다. 이는 일시적인 전력 부족을 넘어 전력 공급의 붕괴까지도 초래할 수 있다. 이렇듯 전력 부하가 집중되는 것은 매우 심각한 문제이다. 전력 공급의 부족은 곧 더 많은 원자력 및 화력 발전소의 건설로 이어지고 이는 경제적 뿐만 아니라 환경적인 관점에서도 바람직하지 않다. 따라서 전력 부하를 수요가 낮은 시간대로 옮김으로써 전력 부하의 심각한 문제를 완화시켜줄 수 있는 효율적인 방법이 요구된다.

따라서 피크 시 전력 사용량의 평준화를 위해서는 특정 시간대에 전력 기기들의 사용 순서를 조정하는 효율적인 스케줄링이 필요하며, 이러한 요구사항을 만족하기 위해 본 논문은 가정과 빌딩에서의 피크부하를 감소시킬 수 있는 다양한 스케줄링 방법을 제안하고 실질적인 전력 부하 프로파일을 기반으로 태스크 모델을 생성하고 전력 소비 스케줄러를 설계한다.

전력 스케줄러는 전력 기기들을 전력 요금에 따라 동적으로 스케줄링한다. 전력 기기 스케줄링은 전력 기기의 태스크들이 시작시간, 실행시간, 종료 시한을 갖고 있고, 이는 주어진 시간 제약 조건 하에 프로세서 상의 태스크들의 실행을 처리하는 실시간 처리 스케줄링과 매우 유사하다 [Facchinetti et al. 2010]. 그러나 총 전력량이 전력선의 현재 허용치를 초과하지 않는 한 여러 개의 전력 기기들의 에너지를 병렬적으로 스케줄링이 가능하다는 점은 다르다. 실행시간은 전력소비량과 비슷하나 전력소비는 더 복잡한 자원들의 요구사항을 갖는다. 게다가 전력 스케줄러는 종료 시한을 만족하는 비율이나 스케줄 능력보다 피크부하 감소나 에너지 비용 절감에 초점을 맞춘다.



전력 소비 스케줄링은 최적화 문제들 중의 한 종류이고, 각각의 전력 기기는 하나의 태스크로 모델링되어진다. 전력 소비 스케줄러는 태스크들을  $M \times N$ 의 할당 테이블로 할당하면서 슬롯 당 전력 소비량의 최대치를 가장 작게 만든다. 이 때,  $M$ 은 스케줄링 윈도우의 타임 슬롯의 개수이고,  $N$ 은 태스크의 개수에 해당한다. 대개의 경우 스케줄링은 태스크의 수에 민감하고 시간복잡도(time complexity)가 매우 큰 문제이기 때문에 실제 시스템 상에서 기존의 평범한 스케줄링 기법으로는 문제를 해결하기가 쉽지 않을 뿐만 아니라 태스크의 수가 증가하면 실행시간은 폭발적으로 증가한다. 휴리스틱 메소드와 유전 알고리즘은 비록 최적의 스케줄을 얻지는 못하지만 서치 스페이스의 사이즈를 감소시켜 실행시간을 줄일 수 있는 합리적인 솔루션이다 [Lee et al. 2011c].

본 논문은 다음과 같이 구성되어 있다.

1장은 이슈가 되고 있는 문제점들과 본 연구의 필요성을 제시한다.

2장은 본 논문의 이론적 배경과 전력 스케줄링과 관련된 수요반응, 유전 알고리즘, 백트래킹, ARIMA 모형에 관련된 연구를 설명한다.

3장은 ARIMA 모형을 생성하여 전력 수요에 대한 예측 및 성능을 평가한다.

4장은 백트래킹을 이용한 전력 소비 스케줄링 기법을 제시하고 제안된 기법의 성능을 평가한다.

5장은 유전 알고리즘을 이용한 전력 소비 스케줄링 기법을 제시하고 제안된 기법의 성능을 평가한다.

6장은 마지막으로 본 논문을 전체적으로 요약하고 결론을 맺는다.

## II. 연구의 배경 및 관련 연구

이 장에서는 전력 수요를 줄일 수 있는 수요반응 기술, 전력 스케줄링과 같은 제약만족 최적화문제(COP; Constraint Satisfaction Optimization Problem)를 해결하는 대표적인 방법인 백트래킹 기법과 유전 알고리즘, 전력 수요를 예측하는데 쓰이는 ARIMA 모델에 대한 배경과 관련된 연구들을 소개한다.

### 1. 수요반응

스마트그리드에 있어서 수요반응은 아주 중요한 요소로서 전력의 수요상황, 피크 부하율 등의 환경 변화에 따라 가격을 조정하고, 소비자는 변동하는 가격에 반응하여 전력 소비량을 줄이거나 전력 사용 시간대를 옮기는 기술이다. 수요반응을 통해 소비자는 에너지를 스스로 관리하고 에너지 절약에 적극적으로 참여할 수 있다 [Spees et al. 2007].

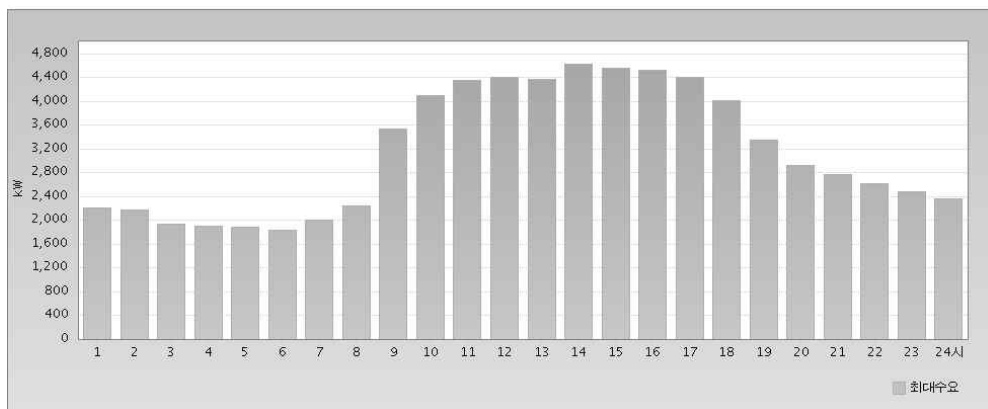


Figure 2. 제주대학교 시간대별 최대수요 (2015.07 ~ 2015.08)

전력 회사는 피크 전력이 감소하는 방향으로 수요반응을 유도하여 공급비용을 절감하고 전력 수급 안정을 도모하기 위해 수요관리(DSM; Demand Side Management)정책을 사용한다 [Mehrshad et al. 2013]. 수요관리는 부하관리를 포괄하는 상위 개념으로 1970년대 미국에서 시작되어 현재 우리나라에서도 시행되고 있다. 미국 EPRI의 보고서에 의하면 공급 가격 상승 시 수요의 2.4% 절감은 도매가격의 25% 하락 효과를 유발하여 시장의 안정성에 기여한다.

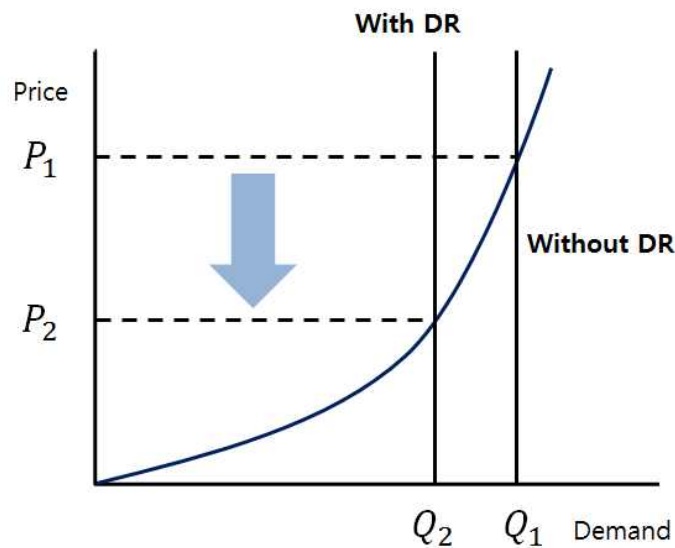


Figure 3. 수요반응으로 인한 전력 시장 가격의 영향

수요반응 프로그램은 가격기반 수요반응과 인센티브 수요반응으로 나뉜다. 가격기반 수요반응(PBP; Price Based Program)은 경제성 기반 수요반응이라고도 하며 전기요금의 가격을 시간대별로 차등 책정하여 소비자가 전력 사용을 자발적으로 줄일 수 있도록 한다. PBP는 계시별 요금제(TOU; Time-Of Use), 피크 요금제(CPP; Critical-Peak Pricing), 실시간 요금제(RTP; Real-Time Pricing)로 나뉜다. TOU는 전력 소비가 계절과 시간에 따라 차이가 클 때 요금을 2부제(On-peak, Off-peak)나 3부제(최대 부하시간, 중간부하시간, 최소부하시간)로 차등 부과하는 방법으로 대부분의 나라에서 시행 중이며 현재 우리나라는 봄·가을,

여름, 겨울철 각각의 전력 수요에 따라 3부제로 구분하여 요금을 차등 부과하고 있다. TOU는 인프라 투자가 다른 요금제에 비해 작고 소비자가 쉽게 접근할 수 있지만 전력 시장 상황을 실시간으로는 반영하지 못한다. CPP는 전력 수요가 높은 시간대 즉, 1%에 해당하는 구간에 특수 고가의 요금을 적용하여 사전에 공지하는 방법으로 TOU보다는 일반적인 소비자가 접근하기 불편하다.

이에 반해 RTP는 전력 시장의 상황을 그대로 반영하며 소비자 요금이 시간대별로 변동된다. 실시간 가격은 소비자들이 전기를 소비하기 하루 전(day-ahead pricing) 혹은 한 시간 전(hour-ahead pricing)에 통보됨으로써 소비자들은 이 가격에 맞춰 전기 사용을 다양하게 할 수 있다 [Mohsenian-Rad et al. 2010]. 그러나 일반 소비자의 경우 전력 가격을 실시간으로 모니터링 하기는 쉽지 않은 단점이 있다. 우리나라는 제주 스마트그리드 실증단지 가정에 전자식 전력량계와 모니터, 제어기 등의 시스템을 제공하고 1년간 RTP를 시범으로 시행하였다. 인센티브 기반 수요반응(IBP; Incentive Based Program)은 신뢰도 기반 수요반응이라고도 하며 전력 공급자 또는 전력 시스템 운영자에 의해 운용되고, DR 프로그램에 참여할 소비자를 미리 모집하여 참여자로 하여금 전력 사용을 줄이게 하여 이에 대한 인센티브를 지급하는 방법이다. Table 1은 인센티브 기반의 수요반응 종류를 나타낸다.

Table 1. 인센티브 기반의 수요반응 종류

종류	특징
직접 부하 제어 (Direct Load Control)	전력계통 신뢰도 및 안정성에 문제가 발생할 경우 단기 통보를 통해 원격으로 참여자의 부하를 차단하고 이에 대한 인센티브를 지불하는 제도
차단 가능 서비스 (Interruptible Service)	전력계통의 사고 동안 부하를 절감하는 대신 요금을 감면해주거나 보상금을 지급하는 제도

수요 입찰 (Demand Bidding)	부하를 절감하는 참여가가 절감 시 지급받기 원하는 가격과 절감 가능 부하량을 제시하여 낙찰 받는 제도
비상시 DR (Emergency DR)	위급한 상황 동안에 부하를 줄인 고객에게 인센티브를 제공하는 제도
용량 시장 (Capacity Market)	보험 개념의 서비스로 일정 용량을 필요시에 미리 정해진 부하 감소량을 소비자들에게 부담하는 제도
보조 서비스 (Ancillary Service)	운영 예비력의 형태로 보조 서비스 시장에 부하 절감을 입찰하는 제도

수요반응을 위한 전력 스케줄링에 관련된 많은 연구들이 있다. 전력 스케줄링에 관한 연구들은 최적의 전력 스케줄링 문제를 해결하기 위해 선형 프로그래밍(linear programming), 개체 군집 최적화(PSO; Particle Swarm Optimization), 게임 이론(game theory), 유전 알고리즘(GA; Genetic Algorithm) 등이 제안되었다. [Kim et al. 2010]은 스마트그리드에서 가정에서 전력 요금의 최소화와 사용자의 만족도를 최대화하기 위해, 수요반응에 관한 가정용 여러 종류의 전력 기계의 특성을 고려하여 최적화된 스케줄링 알고리즘을 제안하였다. 최적화 문제를 해결하기 위해 기존의 0-1 knapsack 문제의 해결 방법을 응용하고, 전력 요금의 한계 값을 넘지 않는 범위 내에서 전력 요금에 대한 전력 기계의 중요도가 높은 기계를 On으로 선택하고 나머지 전력 기계를 Off로 선택함으로써 전력 요금의 한계 내에서의 사용자의 만족도를 최대화할 수 있게 한다.

[Park et al. 2012]은 스마트 홈에서 사용 가능한 최대 전력을 고려하는 홈 단위의 전력 소모에 대한 스케줄링과 스마트그리드 단지의 최소 보장 전력량과 단지 내 전체 이용 가능한 최대 전력량을 고려하고 여러 건물에서 동시에 전력을 소비하는 최적화된 모형을 설계하였다.

## 2. 제약만족 최적화 문제(CSOP)의 접근

태스크의 개수가  $N$ 개 일 때의 스케줄링의 시간 복잡도는  $O(N!)$ 로 알려져 있다. 이러한 문제는 제약만족 최적화문제에 해당하는 NP-hard 문제이다. 이를 해결하기 위한 주요한 방법에는 나무탐색에 해당하는 백트래킹과 인공지능(AI; Artificial Intelligence)에 해당하는 유전 알고리즘, 시뮬레이티드 어닐링(simulated annealing), 타부 서치(tabu search)등이 있다. 이 장에서는 백트래킹과 유전 알고리즘에 대해 소개하고 백트래킹과 유전 알고리즘의 유관 연구를 서술한다.

### 1) 백트래킹

백트래킹은 1950년대 미국의 수학자 Derrick Henry Lehmer가 처음 사용한 용어로 해를 얻을 때까지 모든 가능성을 고려하여 답을 찾아내는 방법이다. 모든 가능성은 하나의 트리로 구성되며 가지 중에 해결책이 존재한다. 백트래킹은 트리를 검사하기 위해 깊이우선탐색(depth first search) 방식과 재귀함수를 사용하며 해를 찾기 위해 전진과 후진을 반복한다. 즉, 탐색 중에 오답을 만나면 이전 분기점으로 돌아간다. 그러나 모든 가능한 경우를 조사하려면 수행 시간이 오래 걸리기 때문에 분기 한정을 사용한다. 분기 한정은 해가 나올 수 있는 모든 가능한 경우를 전부 찾지 않고도 원하는 답을 찾을 수 있다. 가지치기는 답이 될 가능성이 없는 값을 제거하여 속도를 향상시킨다 [Kim et al. 2012].

전력 스케줄링에서 모든 전력 기기들은 같은 시간에 동작할 수 있기 때문에 프로세서의 개수는 제약 조건이 아니다. 대신에 전력 스케줄링은 약간의 특정 제약 조건을 갖는 배낭 채우기 문제(knapsack problem)와 유사하다 [Arbaz et al. 2015]. 이러한 이유로 백트래킹 기반 스케줄러의 정책은 배낭 채우기 문제와 다른 조합 최적화 문제에 대한 가장 간편한 기법 중의 하나인 백트래킹 알고리즘을 따른다.

## 2) 백트래킹 기반 전기 자동차 충전 대기 시간을 최소화하기 위한 관광스케줄링 기법

Figure 4와 같이 전기 자동차를 렌트한 관광객은 일반적으로 관광을 시작하기 전 호텔에서 묵는 동안 전기 자동차를 충분하게 충전하게 된다. 호텔을 출발하여 미리 짜놓은 관광지를 순서대로 방문하게 된다. 전기 자동차가 관광지에 도착했을 때 운전자는 보는 관광에서부터 체험에 이르기까지 관광 지점마다 특정한 관광 활동을 하게 되며 운전자는 각각의 관광지마다 관광을 하는 동안 전기 자동차를 충전한다.

이 때 전기 자동차가 완전 충전이 된 경우라 할지라도 관광 시 일일 주행거리가 100 km를 넘을 가능성이 높고, 도로 여건과 날씨, 운전자의 주행스타일에 의해 배터리의 소모가 발생하게 되므로 배터리의 양이 부족할 수 있다. 즉, 다음 목적지에 도달하기 위한 충분한 배터리의 양이 남아 있지 않다면 충전을 위한 대기 시간이 필요하며, 관광을 하는 동안 전기 자동차의 충전이 이루어져야 한다 [Lee et al. 2012k].

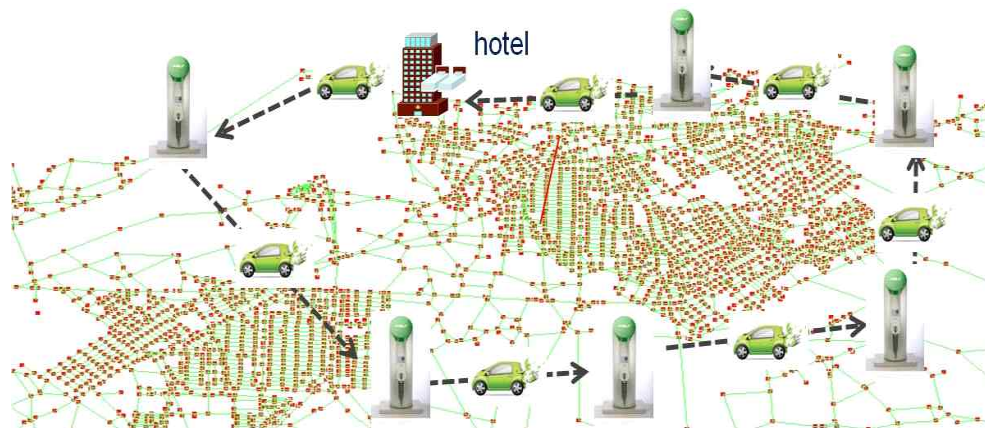


Figure 4. 전기 자동차의 관광 스케줄 모델

예를 들어, 운전자가 출발 전에 방문하고자 하는 목적지가 해안도로, 공연장, 음식점등이라 했을 때 만일 전기 자동차를 이용하여 관광을 하고 있는 도중 다음

관광지로 이동하기 전의 배터리 잔량이 충분하지 않다면 충전을 위한 대기 시간이 필요하다. 운전자의 충전은 해안도로의 드라이브 코스는 주행을 하는 관광지가 되기 때문에 충전이 불가능하고, 공연장과 음식점 등은 관광을 하는 동안 충전이 가능하며 또한 공연장이 음식점보다 체류시간이 많기 때문에 충전량도 많게 된다.

이처럼 전기 자동차를 이용한 관광은 방문 스케줄에 따라 사용자의 대기 시간이 달라지며 관광지에서의 관광 시간에 따라 충전 가능량이 달라지게 된다. 체류 시간 동안 전기 자동차의 충전이 가능하기 때문에 전기 자동차 충전 시간과 체류 시간을 겹치게 하는 것이 대기 시간을 줄일 수 있다. 즉, 효과적인 관광 스케줄은 대기 시간을 최소화하는 것이며, 이를 위해서는 가능한 많이 관광에서의 체류 시간과 충전을 하는 시간을 겹칠 수 있게 해야 한다 [Lee et al. 2011i]. 따라서 다중 목적지의 경우 어떤 순서로 가야 방문 총거리가 짧은지에 대한 고전적인 TSP (Traveling Salesman Problem) 문제뿐만 아니라 어떤 순서로 방문하면 충전 대기 시간이 최소화될 수 있는지에 대한 문제를 해결해주면 관광에 있어서 낭비시간을 줄여 최적의 관광 스케줄이 가능하게 된다. 특정 방문 순서의 대기 시간 감소를 위한 대기 시간 추정 모델을 정의함에 있어서 도로 네트워크는 방문 지점을 설명한 관심 지역 정보(POI; Point Of Interest)를 포함할 수 있고, A\* 알고리즘을 이용하여 두 지점 사이의 운행 거리 및 운행 시간을 추정할 수 있다. Figure 5와 같이 2개의 관광지만을 대상으로 한 경우의 예를 들면 먼저, 전기 자동차가 임의의 지점에서 Spot1 또는 Spot2에 도착했을 때 초기 배터리 잔량의 거리 지수는 10 km이고, Spot1과 Spot2 사이의 거리는 30 km라고 가정한다. 또한 1시간 충전으로 15 km의 주행 거리가 증가한다고 가정하고 Spot1에서의 예정된 체류 시간은 1시간, Spot2에서 예정된 체류 시간은 3시간이라고 가정한다. 거리 지수(distance credit)는 전기 자동차가 특정 배터리 량으로 갈 수 있는 가용 거리를 의미한다. 거리 지수는 전기 자동차가 충전 시 증가하고, 전기 자동차가 운행한 거리에 따라 감소한다.



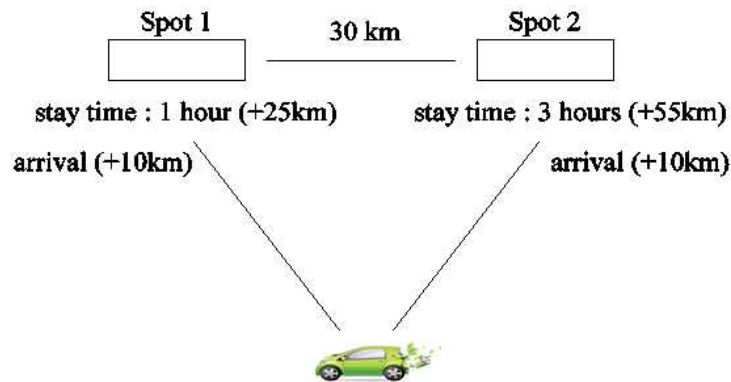


Figure 5. 대기 시간의 추정

이 때 Spot1 → Spot2의 순서로 방문한다면 Spot1에서 관광시간은 1시간으로 15 km 주행거리만큼의 충전량이 증가하게 되어 사용 가능한 가능한 배터리 용량의 거리 지수는 25 km가 된다. 즉 Spot2까지 가는데 필요한 30 km를 확보하려면 5 km 주행거리에 필요한 양만큼 대기하여야 한다. 따라서 20분의 대기 시간이 발생한다. 반면 Spot2 → Spot1의 순서로 방문하게 되면 Spot2에서 관광시간은 3시간으로 45 km 주행 거리 만큼의 충전량이 증가하게 되어 거리지수는 55 km가 된다. 따라서 Spot1까지 가는데 필요한 30 km를 이미 확보하고 있으므로 추가적인 대기 시간이 없게 된다. 그러므로 Spot2 → Spot1의 순서로 가는 것이 Spot1 → Spot2의 순서로 가는 것보다 더 효율적일 수 있다 [Lee et al. 2012f].

관광객은 인터넷과 관광 추천 시스템 등에서의 정보를 통해 자신들의 원하는 목적지를 결정한 후에 관광 스케줄을 작성하게 된다. 만일 단지 우리가 관광지간의 이동 거리만을 고려한다면, 관광 스케줄은 TSP solver에 의해 생성될 수 있다. TSP는 모든 노드를 방문하고 출발지점으로 되돌아 오는데 가장 최소의 비용이 드는 방법을 찾는 것이다. 즉, 방문 목적지가  $\{V_1, V_2, \dots, V_n\}$  일 때 수식 (1)의 비용함수를 최소화하는 방문 순서를 찾아야 한다. TSP는 대표적인  $O(n!)$  문제로 시간복잡도가 높은 NP-problem에 해당한다.

$$F(V_1, V_2, \dots, V_n) = \sum D(V_i, V_{i+1}) + D(V_{i+1}, V_1) \quad (1)$$

대기 시간은 TSP의 수식 (1)과 유사한 방법으로 추정되어질 수 있다. 수식 (2)에서  $B_{in}^i$  은 관광지  $V_i$ 에 도착할 때 배터리 량이고,  $B_{av}^i$  는  $V_i$  도착할 때  $V_{i+1}$ 로 갈 수 있는 남은 배터리 량이다.  $W_i$ 는 관광지  $V_i$  방문시의 대기 시간이며, 총 대기 시간은  $\sum W_i$  이다.  $W_i$ 는 수식 (2)에 의해 구해질 수 있다 [Lee et al. 2012j].

- $B_{av}^i$  : 관광지  $i$ 에 도착할 때 가용해질 수 있는 배터리 량
- $B_{in}^i$  : 관광지  $i$ 에 도착할 때 배터리 량
- $B_{out}^i$  : 관광지  $i$ 를 출발할 때의 배터리 량
- $T(V_i)$  : 관광지  $i$ 에서의 관광시간
- $D(V_i, V_{i+1})$  : 관광지  $i$ 와 관광지  $i+1$  사이의 거리
- $W_i$  : 관광지  $i$ 를 방문시의 대기 시간
- 최종 비용 함수: Cost (order) =  $\sum W_i$

$$\begin{aligned}
 B_{av}^i &= \min(B_{\max}, B_{in}^i + T(V_i)) \\
 W_i &= W^{i-1} - \min(0, B_{av}^i - D(V_i, V_{i+1})) \\
 B_{out}^i &= \max(0, B_{av}^i - D(V_i, V_{i+1}))
 \end{aligned}
 \tag{2}$$

이 때,  $B_{\max}$  는 전기 자동차의 최대 배터리 충전량이고,  $T(V_i)$ 는 관광지  $V_i$ 에서의 체류시간이다. 만약  $V_1$ 이 호텔이고 운전자가 전기 자동차를 완충하여 관광을 시작한다면,  $B_{in}^1 = B_{\max}$  이고  $T(V_1) = 0$ 이다.  $B_{av}^i$  는 관광지  $V_i$ 에서의 체류시간동안 얻은 충전량을 더하게 되는데 그때의 상한 값이  $B_{\max}$ 가 된다. 또한  $B_{av}^i$  가  $D(V_i, V_{i+1})$ 보다 크다면 관광지  $V_{i+1}$ 까지 가는데 배터리 량은 충분히 남아 있다는 것이고  $\min(0, B_{av}^i - D(V_i, V_{i+1}))$ 이 0이 되므로 이는 추가적인 충전 시간과 대

기 시간을 필요로 하지 않기 때문에  $W_i$ 의 값이 0된다. 반면에  $B_{av}^i - D(V_i, V_{i+1})$ 가 음의 값이 된다면 가용 배터리 량이 다음 관광지로 가는데의 비용보다 부족하다는 의미가 된다. 즉,  $\min(0, B_{av}^i - D(V_i, V_{i+1}))$ 도 음의 값으로 곱해지기 때문에 대기 시간  $W_i$ 은 0보다 큰 값이며  $W^{i-1}$ 에서  $-\min(0, B_{av}^i - D(V_i, V_{i+1}))$ 만큼 합한 것이 된다. 결국 관광지  $V_i$ 에서의 추가적인 충전의 필수 여부에 따라 다음 관광지에 도착했을 때의 배터리 량  $B_{out}^i$  즉,  $B_{in}^{i+1}$ 은 이 될 수도 있고  $B_{av}^i - D(V_i, V_{i+1})$ 가 될 수도 있다.

Figure 6과 같이 TSP solver가 비용함수를 최소화 하는 방문 순서를 찾아준다면 제안된 스케줄러는 백트래킹 방식을 통해 대기 시간 추정 모델에서 거리정보에 대기 시간을 포함하는  $\sum W_i$ 을 최소화 하는 최적의 스케줄을 찾는다.

따라서 가능한 방문 스케줄을 모두 검색하고 recursive하게 순서들을 생성한다. 리프 노드를 만나게 되면 비용 함수를 호출하여 전역 최소값 변경 여부를 조사하며, 중간 노드 확장 시 이미 최소값을 넘어서면 중지시키게 된다.

일반차량	TSP solver : visiting order
EV 차량	EV Scheduler : visiting order + waiting time

Figure 6. TSP와 전기 자동차 스케줄러의 비교

### 3) 유전 알고리즘

유전 알고리즘(Genetic Algorithm)은 수 없이 많은 해가 존재하는 임의의 해 집단(population)을 생성하고 이들을 점차 변화시킴으로서 가장 품질이 우수한 해를 찾는데 유용한 알고리즘이다. 유전 알고리즘은 선택, 교차, 변이, 대치 연산이라는 진화의 과정을 기반으로 한 기계학습의 탐색 방법이다. 즉, 부모 세대가 평가를 받아 자식 세대를 생성하고, 그 자식 세대가 다시 부모 세대가 되어 평가를 받아 자식 세대를 생성하는 것을 반복하게 되면 그 집단에 있는 해는 대부분 높은 평가를 받게 된다. 즉 최적의 해를 찾을 수 있다. 유전 알고리즘은 수식으로 탐색하기 어려운 문제, 서치 스페이스가 넓어 최고의 해를 찾기가 쉽지 않은 경우 그에 버금가는 해 정도만 찾아도 되는 경우에 사용되는 탐색 방법으로 NP 문제의 최적화에 사용된다 [Lee et al. 2011d].

Figure 7은 일반적인 유전 알고리즘의 과정을, Figure 8은 유전 알고리즘의 전형적인 구조를 보여준다. 먼저 초기 해 집단을 랜덤하게 생성한 후, 우수한 유전자를 선택하기 위한 적합도 함수를 구한다. 이 후 적합도 계산에 따라 선택 연산에 의해 부모 해(parent)를 선택한 후 교차와 변이 연산을 통해 자식 해(offspring)를 생성한다. 이를 통해 생성된 자식 해는 대치 연산을 통해 새로운 부모 세대로 대치되고 정지 조건을 만족하면 알고리즘을 종료하고 그렇지 않으면 계속 반복해 나간다. 유전 알고리즘은 이러한 반복 과정을 거쳐 세대가 진화함으로써 문제에 최적 해에 접근하게 된다. 하나의 반복을 세대(generation)라 하고, 세대 전체를 하나의 실행이라 하며, 실행이 끝날 때까지 적합도가 높은 염색체 한 개 이상을 찾는다. 유전 알고리즘에서는 세대수가 보통 50 ~ 500 또는 그 이상일 수도 있다.

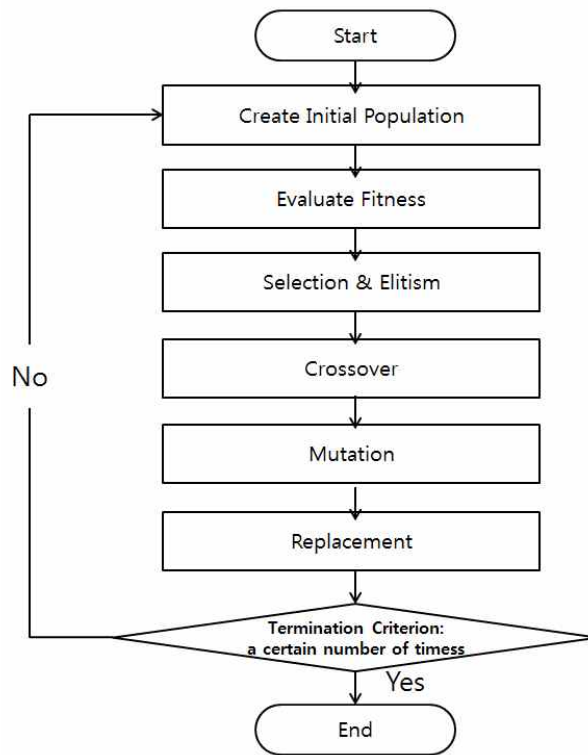


Figure 7. 유전 알고리즘의 순서도

```

n개의 초기 염색체 생성;
repeat {
  for = i to k {
    두 염색체  $p_1, p_2$  선택;
     $offspring_i = crossover(p_1, p_2)$ ;
     $offspring_i = mutation(offspring_i)$ ;
  }
   $offspring_1, \dots, offspring_k$ 를 population 내의 k개의 염색체와 대치;
} until(정지 조건 만족 또는 지정된 횟수);
남은 염색체 중 최상의 염색체를 return;
  
```

Figure 8. 유전 알고리즘의 과정

유전 알고리즘에서는 우선 해의 집단을 랜덤하게 채워 초기 해 집단을 생성한다. John Holland 유전 알고리즘은 인코딩을 통해 0과 1로 표현된 인공적인 염색체(chromosome)를 생성한다. 염색체는 유전자(gene) 여러 개로 이루어지고, 유전자는 0 또는 1로 표현된다. 주어진 문제에 맞추어 다양한 인코딩 방식이 제안되었다.

Figure 9는 전력 스케줄링을 위한 유전 알고리즘의 인코딩 방법의 한 예를 보여준다. 행은 전력 수요들을 나타내고, 열은 0부터 24시까지를 표현한 후, 0과 1의 값을 가지는 행렬로 구성하였다. 0은 전력 수요가 해당 시간에 없는 상태(Off)를 1은 해당 시간에 전력 수요가 있는 상태(On)를 나타낸다 [Lee et al. 2014].

	0시	24시
전력수요1	0101000100 ...	1
전력수요2	1000001001 ...	0
...		
전력수요n	0000001001 ...	0

Figure 9. 전력 데이터의 인코딩 예시

진화의 목표는 적합도가 증가하는 개체를 생성하는 것이다. 자연 환경에 더 적합한 개체는 살아남고 적응하지 못한 개체는 도태되어진다. 환경에의 적응 정도를 수량적으로 나타내는 적합도에 의해 개체가 선택된다. 따라서 랜덤하게 채운 해의 집단을 적합도 함수(fitness function)를 통해 풀려는 문제에 대한 염색체의 성능, 즉 적합도를 검증한다. 적합도 함수는 현재 상태에 유전자가 얼마나 잘 적용하는지를 점수화 시키는 함수이고, 점수를 통해 현재 해의 우수한 정도를 알 수 있는 평가 기준이다. 적합도 함수를 통해 적합도가 구해지면 이를 통해 좋은

유전자를 선택하기도 하고, 선택되지 않은 유전자는 버리기도 한다. 즉, 선택의 목표는 적합도가 높은 개체에 더 많은 기회를 주는 것이다. 이렇게 선택된 해를 부모 해라 하고 이를 복제와 일부 개량과 같은 다양한 연산으로 자식 해를 재생산한다. 유전 알고리즘의 연산은 선택(selection), 교차(crossover), 돌연변이(mutation)가 있다.

먼저 선택은 한 세대에서 다음 세대로 전해지는 해의 후보가 되는 해들을 선택하는 과정이다. 선택에는 룰렛 휠(roulette wheel) 방식, 순위(ranking) 방식, 토너먼트(tournament) 방식 등 다양한 방법이 있다.

룰렛 휠 방식은 유전자 적응도에 따른 확률로 선택하는 방법으로 일반적으로 가장 많이 사용되는 방법으로 적응도가 높은 유전자는 확률을 높게 설정하여 선택되기 쉽게 하고, 적응도가 낮은 유전자는 확률을 낮게 설정하여 도태되기 쉽게 한다. 룰렛 휠은 원형 회전판을 몇 개의 구역으로 나누고, 이 회전판이 돌아가고 있는 상태에서 다트를 던져 맞추는 방식으로 다트가 회전판의 어느 구역에 맞춰지는 것은 랜덤하나, 확률로 보면 넓은 구역에 맞춰질 가능성이 높다. 룰렛 휠은 모든 개체의 적합도 함수 값을 구해서 이를 더하고, 각각의 개체가 차지하는 영역의 비율을 통해 룰렛 판의 영역을 결정한다. 당연히 개체의 적합도가 높아 좋은 품질이라고 생각되는 개체는 많은 영역을 차지하게 되고, 적합도가 낮아 나쁜 품질이라고 생각되는 개체는 작은 영역을 차지하게 되므로 좋은 품질의 개체가 부모로 선택될 확률이 커지게 된다.

룰렛 휠 방식에서 해 집단 해  $i$ 의 적합도  $f_i$  다음 수식 (3)과 같다.

$$f(i) = (C_w - C_i) + \frac{(C_w - C_b)}{(k-1)}, k > 1 \quad (3)$$

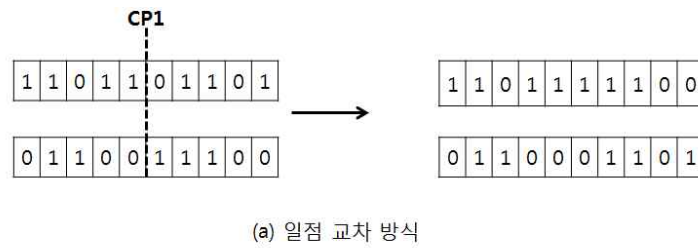
$C_w$ 는 해 집단 내에서 가장 나쁜 품질의 개체가 가지는 비용,  $C_b$ 는 해 집단 내에서 가장 좋은 품질의 개체가 가지는 비용,  $C_i$ 는  $i$ 번째 개체가 가지는 비용으로

좋은 품질의 개체일수록 적합도  $f_i$ 의 값은 커진다. 일반적으로  $k$ 의 값은 3 ~ 4의 값을 갖는다. 만일 해 집단이 7개의 개체로 이루어져 있다면  $f(1) \sim f(7)$ 의 값을 모두 더하는 값을 전체 면적으로 하는 룰렛 휠을 만들고 각 적합도 만큼 해당 해를 배정한다.

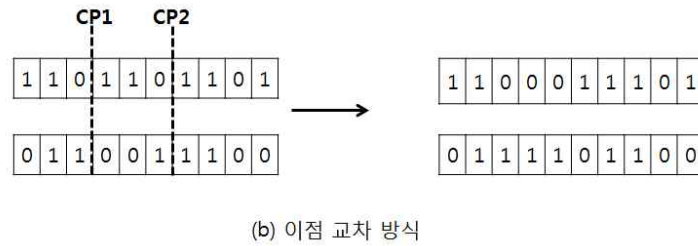
또한 순위 방식은 유전자를 적응도 순서로 나열해 순위에 따라 선택 확률을 부여하는 방법이고, 토너먼트 방식은 유전자 개수를 적당히 랜덤하게 선택하고 그 중에서 가장 적응도가 높은 유전자를 선택하는 방식이다. 이러한 해의 선택 방식은 유전 알고리즘 성능에 큰 영향을 줄 수 있다. 어떤 방식을 쓰느냐에 따라 최적의 해를 찾는 속도가 느려지거나 또는 찾은 해가 일부 특정 집단에서만 최적일 수 있기 때문이다.

이렇게 선택되어진 해는 교차를 통해 유전자를 재조합하면 다양한 다음 세대를 생성하게 된다. 즉, 교차는 두 개의 부모 해의 유전자를 서로 교차시켜서 자식 해를 만들어낸다. Figure 10은 가장 대표적인 교차 연산인 일점 교차와 이점 교차이다. 일점 교차와 이점 교차는 임의의 교차점을 선택하여 각각의 기준으로 두 유전자로부터 번갈아 정보를 받아오는 방법이다. Figure 14에서 보듯이, 일점 교차 방식은 하나의 교차점(Cross Point, CP)을 기준으로 부모 1과 부모 2를 교차하는 방법이다. 반면 이점 교차 방식은 2개의 교차점을 기준으로 교차가 일어난다. 선택되는 교차점은 랜덤이므로 어느 부모에게서 더 많은 유전자를 물려받게 될지는 예측하기 어렵다. 이 외에도 균등 교차, 사이클 교차, PMX(Partially Matched Crossover), 산술적 교차, 휴리스틱 교차, 간선 재결합 등의 다양한 방법이 있다. 다양한 교차 방식 중 어떻게 교차할 것인가에 대해서도 충분한 고려가 필요하다. 만일 좋은 품질의 부모 해가 선택되었을 때 다점 교차를 적용하게 되면 품질이 좋은 유전자는 여러 부분으로 나뉘어져 자식에게 전달되어지기 때문에 일점 교차보다 품질이 떨어지는 자식 해가 생성될 가능성이 높다.





(a) 일점 교차 방식



(b) 이점 교차 방식

Figure 10. 유전 알고리즘의 교차 연산

유전학에서 돌연변이가 발생하듯이 유전 알고리즘에서도 변이 연산을 통해 유사한 과정을 거친다. 변이 연산은 교차 연산을 통해 물려받은 유전자에 작은 확률로 변화를 준다. 적합도가 높은 부모 해는 다음 자식을 생성할 확률이 높지만 100%는 아니며 적합도가 낮은 부모 해는 절대 자식을 생성할 수 없는 것도 아니다. 적합도가 높은 해들끼리만 계속 교차를 시켜 자식 세대를 생성하면 국소 최저점에 빠지게 된다. 돌연변이는 국소 최저점에 빠지는 것을 방지하고 해 집단의 다양성을 높여주는 유전자 조작 방법이다. 변이는 특정 확률로 유전자 상의 정보를 랜덤하게 바꿔주는 조작이다. 예를 들어 염색체의 가장 왼쪽 유전자로부터 시작하여 각각의 유전자에 대해 [0,1] 범위의 난수를 만들어 내고, 미리 정한 임의의 임계값 미만의 수가 나오면 해당 유전자를 임의로 변형시킨다. 변이의 확률을 높이면 탐색 공간을 넓힘으로써 다양한 해를 탐색하고 최적 해를 찾을 확률을 높이지만 수행 시간이 느려진다. 따라서 보통 0.01 또는 0.001의 작은 확률로 변이를 발생시킨다.

교차와 변이 연산을 통해 새롭게 생성된 자식 해를 해 집단에 추가하고, 기존 해 중 열등한 해를 가려내서 제외시키기 위해 대체(Replacement) 연산을 사용한다. 가장 쉬운 대체 방법은 해 집단 내에서 가장 품질이 낮은 해를 제거하고 새로운

해로 대체해 주는 것이다. 이외에도 부모 해를 자식해로 모두 대체하는 방법과 생성한 자식 해와 부모 해의 품질을 비교하여 품질이 더 좋은 해로 대체하는 방법이 있다. 유전 알고리즘은 현재 아래와 같이 폭넓고 다양한 분야에서 스케줄링에 관한 문제에 사용되어지고 있다 [Montana et al. 1998].

- jobshop / flowshop scheduling
- urban transit systems
- supply chain management
- exam timetabling
- scheduling computing tasks
- scheduling laboratory equipment
- crew scheduling
- maintenance / rehabilitation scheduling
- talent / project scheduling

[Zaho et al. 2013]은 가정에서의 전력 기기를 OMA(Operation Manually Appliances)와 OAA(Operation Automatically Appliances)로 구분하고 거주자의 선호도와 RTEP(Real-Time Electricity Price)를 통해 모든 OAA의 전력 사용을 효과적으로 관리할 수 있는 스케줄링 기법을 제안한다.

OMAs는 TV와 PC와 같이 스위치를 수동적으로 On/off 해야 하기 때문에 OAA를 스케줄링한다. 가정에서의 OAA는 식기 세척기와 같은 선점형과과 전기포트와 같은 비선점형으로 구분하고, HAN system에 의해 HG(Home Gateway)는 실시간 전력 요금 등의 DR 정보를 받아 EMC(Energy Management Controller)로 전송한다. EMC는 최적의 전력 스케줄링을 완성하고 이는 HG에 의해 각각의 전력 기기로 전달됨으로써 모든 전력 기기들은 효율적인 스케줄링에 따라 자동으로 작동한다. RTP 가격 모델만을 적용하게 되면 많은 OAA는 전력 요금이 가장 저렴한 시간으로 사용 시간을 변경하기 때문에 그 시점의 PAR의 상승으로 전력 시스템의 마비를 초래할 수도 있다.

따라서 RTP만을 스케줄링 기법만을 사용하지 않고, IBR(Inclining Block Rate)와

RTP를 결합한 요금 모델을 적용한다. 이로써 전력 요금을 최소화하고 PAR(Peak Average Rate)를 완화시키며 동시에 전력 시스템의 안정화를 가져올 수 있다. 최적화 문제는 보통 비선형이기 때문에 GA를 사용하여 문제를 해결하였다. 선택 연산은 룰렛 휠 방식을 적용하였다. 일반적으로 GA의 시간 복잡도는  $O(\text{세대수} * (\text{변이 복잡도} + \text{교차 복잡도} + \text{선택 복잡도}))$ 로 나타내어진다. 최대 세대수, 해 집단의 크기, 개체들의 수를 각각  $g, N, n_a$ 라고 나타냄을 가정하고 시간 복잡도를  $O(gNn_a)$ 라고 표현한다. 이 경우에 시간의 비용은 세 개의 파라미터가 증가할수록 증가함에 따라 대체적으로 GA 시간의 비용은 만족하기 어렵다. 그러나 제안한 전력 소비 스케줄링 프로세스는 하루의 시작에 시행되어지므로 시간 파라미터들이 결정되어진 후에 전력 스케줄링을 위한 충분한 시간이 있다. 또한 수행 시간 문제는 그렇게 중요한 것은 아니다. GA에서 해 집단의 크기는 200, 교차와 변이의 확률은 각각 90%와 2%이다. 결국 세대수가 1,000에 도달하게 되면 진화 프로세스는 종료된다.

[Lee et al. 2014]은 스마트 전력망계가 전면적으로 사용되는 차세대 전력망 환경에서 실시간 전력 요금제에 적합한 스마트 빌딩 내 전력 스케줄링 기법을 제안한다. 전력 요금의 변화에 따라 각각의 전력 기기들의 전력 모드가 동적으로 바뀐다. 제안하는 기법은 유전 알고리즘을 이용하여 실시간 전력 요금 정보에 따라 스마트 빌딩 내의 전력 수요를 스케줄링하여 전력 요금을 최소화한다.

각 전력 수요별 최소 수행 주기, 꼭 필요한 수행인지 여부 등을 유전 알고리즘을 이용해 최소의 전력 요금으로 스마트 빌딩 내 모든 전력 수요들이 안정적으로 수행될 수 있는 최적의 스케줄링 기법을 탐색하여 해당 스케줄에 맞게 모든 시스템을 제어한다. 특히, 개별 시스템의 전력 모드 변경을 협력적으로 수행하는 시분할 스케줄링을 통해 각 타임 구간별로 전력 사용량의 합이 고비용 누진 임계치를 넘지 않도록 스케줄링을 최적화한다. 각각의 전력 수요를 행으로, 하루의 24시간을 열로 표현한 후, 0과 1의 값을 가지는 행렬로 해를 구성한다. 초기 해 집단은 1,000개의 해를 랜덤하게 발생시켜 구성한다. 0은 해당 전력 수요가 해당 시간에 꺼진 상태를, 1은 해당 전력 수요가 해당 시간에 꺼진 상태를 의미한다. 생성된 해가 전력 수요별로 존재하는 시간 제약 조건을 만족하지 못할 경우는

각 전력 수요별로 보정 연산을 수행하여 모든 해가 제약 조건을 만족하도록 보정한다. 부모해의 선택 연산은 룰렛 휠 방식을 사용한다. 또한 우수한 해가 선택될 확률이 높음으로 인해 해 집단의 다양성을 해칠 수 있기 때문에 해 집단에서 가장 우수한 해가 선택될 확률이 가장 열등한 해가 선택될 확률의 4배가 되도록 적합도 값을 정규화한다. 교차 연산은 가장 많이 사용되는 일점 교차를 행렬에 적용하는 방식을 사용한다. 변이 연산 따로 시행하지 않았고 대치 연산은 강한 수렴성을 추구하기 위해 품질이 가장 나쁜 해를 제거하고 새로운 해로 대치해주는 방법을 사용한다.

#### 4) 유전 알고리즘을 이용한 전기 자동차 재배치 스케줄링 기법

전기 자동차 셰어링 시스템은 사용자가 전기 자동차를 픽업한 스테이션에 반납을 하도록하는 모델도 있지만 사용자가 자신이 전기 자동차를 픽업한 스테이션이 아닌 다른 셰어링 스테이션에 전기 자동차를 반납할 수 있는 단방향 렌탈 모델을 사용할 수 있다. 단방향 렌탈 모델은 픽업한 곳에서 반납을 하지 않아도 되기 때문에 사용자의 관점에서는 매우 편리한 서비스 이지만 복잡한 관리 문제를 가질 수 있다.

사용자가 전기 자동차를 픽업하기 위해서는 픽업하려는 스테이션에 적어도 하나의 전기 자동차가 있어야 하지만 스테이션들마다 픽업 요청량과 반납되는 전기 자동차 수가 다를 수 있기 때문에 일부 스테이션은 서비스할 차량이 부족하거나 심지어 없게 될 수 있는 스테이션들의 서비스 차량 불균형 문제가 발생하게 된다 [Lee et al. 2012c]. 이러한 전기 자동차 차량 불균형 문제는 재배치 전략(relocation strategy)으로 재배치 벡터(relocation vector)를 생성하고, 유전 알고리즘을 이용하여 재배치를 스케줄링한다.

재배치 전략은 이븐 재배치 스킴(even relocation scheme), 사용률 기반 재배치 스킴(utilization-based relocation scheme), 모닝 중점 스킴(morning-focused relocation scheme)의 3가지 방법으로 이를 통해 재배치 목표 전기 자동차 분포 즉, 재배치 벡터(relocation vector)를 계산할 수 있다.

이른 재배치 스킴은 모든 스테이션에 동일한 수의 전기 자동차를 할당하는 방법으로 수요패턴은 고려하지 않지만 장기적인 관점에서 유리한 방법이다. 사용자 기반 재배치 스킴은 각 스테이션에서의 수요 비율에 따라 차량을 할당하는 방법이다. 이는 픽업을 원하는 사용자가 많은 스테이션에 더 많은 차량을 할당하는 것이다. 모닝 중점 스킴은 운영 시작 시간 이후 일정 시간 동안의 사용자들의 픽업 요청 비율에 따라 차량을 할당하는 방법이다. 즉, 사용자 기반과 모닝 중점 스킴은 각각 전체 픽업 비율, 오전시간대의 픽업 비율에 따라 전기 자동차를 할당하는 것이다.

· Even relocation  $V_i = \left\lfloor \frac{N}{N_s} \right\rfloor$  (4)

· Utilization-based relocation  $V_i = N \times \frac{P_i}{\sum P_i}$  (5)

· Morning-focused relocation  $V_i = N \times \frac{M_i}{\sum M_i}$  (6)

Table 2. Relocation weights

Weights	$S_1$	$S_2$	$S_3$	$S_4$	$S_5$
Even	0.20	0.20	0.20	0.20	0.20
Utilization-based	0.10	0.46	0.28	0.02	0.14
Morning-focused	0.17	0.43	0.23	0.04	0.13

각각의 재배치 기법의 재배치 목표량은 전기 자동차의 총 개수와 가중치의 곱으로 계산할 수 있다. 이 때, 가중치는 Table 2에서와 같이 각 스테이션에서의 전기 자동차 할당비율을 의미한다. 예를 들어 사용자 기반 재배치 스킴인 경우 스

테이션이 5곳이고, 전기 자동차의 총 개수가 50대일 때 50과 가중치를 각각 곱하면 재배치 목표량은 (5, 23, 14, 1, 7)이 된다.

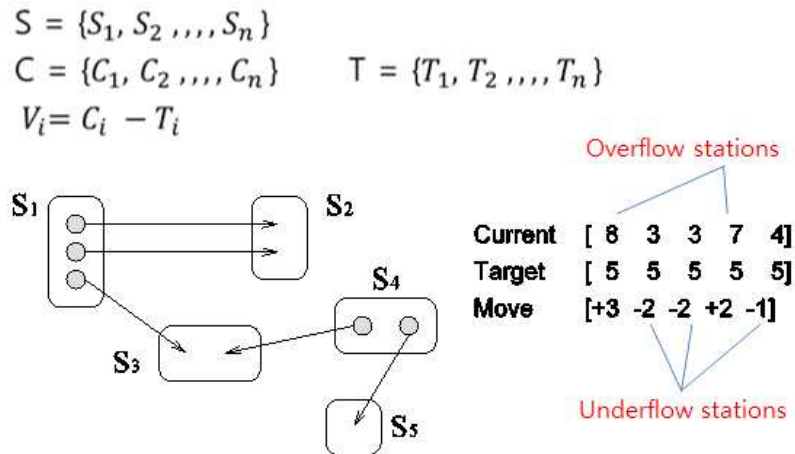


Figure 11. 전기 자동차-충전소 매칭

S는 스테이션들의 집합을 나타내고, C는 현재 분포량을 T는 목표 분포량을 나타낸다. 그리고  $V_i$ 는 재배치 벡터로서 현재 분포량과 목표 분포량의 차이로 계산할 수 있고, 이는 스테이션의 과부족 정보를 나타낸다. 즉,  $V_i$ 가 +이면 목표량보다 현재 분포량이 초과 상태를 의미하는 오버플로우 스테이션이 되고,  $V_i$ 가 -이면 목표량보다 현재 분포량이 부족한 상태를 의미하는 언더플로우 스테이션이 된다. Figure 11에서 오버플로우 스테이션인  $S_1$ 과  $S_4$ 의 전기 자동차는 언더플로우 스테이션인  $S_2, S_3, S_5$ 로 이동시켜 전기 자동차와 스테이션간의 매칭을 시켜주게 된다 [Lee et al. 2012h]. 재배치 스케줄링은 전기 자동차의 수가 크게 증가할수록 매칭에 대한 조합이 늘어나기 때문에 시간복잡도가 커지게 된다. 따라서 가능한 시간 내에서 재배치에 대한 합리적인 솔루션을 얻을 수 있는 유전 알고리즘을 이용하여 재배치를 스케줄링 한다.

유전 알고리즘은 염색체의 해들로 구성된 해 집단을 갖는다. 먼저 n개의 해를 임의로 생성하는 초기화 단계로 시작되며, 이 해 집단으로부터 선택, 교차, 변이의

단계를 거쳐 k개의 새로운 해를 만들어낸다. 생성된 k개의 해는 해 집단의 k개의 해와 대치된다. 이러한 과정을 임의의 정지 조건이 만족될 때까지 수행한 후 해 집단에 남은 해 중 가장 좋은 해를 답으로 정하게 된다. 정지 조건은 repeat-until 루프를 일정 횟수만큼 수행한 다음 정지시키는 방법과 해 집단에 있는 해들의 다양성이 어느 정도 이하로 떨어지는 시점에 정지시키는 방법이 있다. repeat-until 루프를 정지할 경우 그 정도이면 해들의 어느 정도 수렴할 것이라는 경험적 짐작이 있어야 한다. 또한 다양성이 떨어지는 것을 판단하기 위해서는 해 집단 내의 염색체들 중 대부분이 똑같은지를 확인해야 한다 [Lee et al. 2012e]. 유전 알고리즘은 염색체의 해들로 구성된 해 집단을 갖는다. 유전 알고리즘을 이용해 재배치를 스케줄링하기 위해서는 Figure 12와 같은 인코딩 기법을 통해 정수 값을 가지는 벡터(integer-valued vector)로 인코딩해야 한다.

Relocation vector : [3, -2, -2, +2, -1]

A vector-encoded schedule: [1, 0, 2, -1, 4, 3]

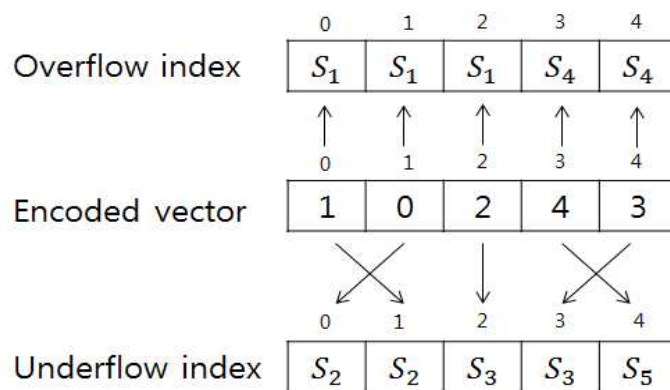


Figure 12. Encoding scheme

재배치 전략으로부터 재배치 벡터 [3, -2, -2, +2, -1]가 생성되면 오버플로 스테이션에 속하는 차량과 언더플로우 스테이션의 차량으로 구성되는 재배치 페어들

의 집합을 생성할 수 있다. 이를 위해 오버플로 스테이션과 언더플로 스테이션을 포함하는 인덱스 테이블을 생성할 수 있다. 인코딩 벡터의 정수 엘리먼트들의 인덱스 위치는 오버플로 스테이션의 인덱스 정보를 나타내고, 정수 엘리먼트 값은 언더플로 스테이션의 인덱스 정보를 나타낸다. 만약 인코딩 벡터가 [1, 0, 2, -1, 4, 3]이라면 정수 1의 위치가 0이므로 오버플로 스테이션 인덱스 0의  $S_1$ , 언더플로 스테이션 인덱스 1의  $S_2$ 이므로 재배치 페어는  $(S_1, S_2)$ 가 되며, 정수 0의 위치가 1이므로 오버플로 스테이션 인덱스 1의  $S_1$ , 언더플로 스테이션 인덱스 0의  $S_2$ 가 된다. 동일한 방법을 통해 결국 염색체 [1, 0, 2, -1, 4, 3]은  $\{(S_1, S_2), (S_1, S_2), (S_1, S_3), (S_4, S_5), (S_4, S_3)\}$ 의 재배치 페어들이 생성된다. 벡터의 음수 값은 재배치 팀을 분리하기 위해 사용된다. 즉, (1, 0, 2)와 (4, 3)은 2개의 팀을 위한 서브벡터가 된다.

이러한 인코딩 벡터는 랜덤하게 생성될 수 있으며 생성된 여러 개의 벡터들의 집합은 최초의 해 집합이 된다. 해 집합은 선택, 교차, 변이 및 생산을 통해 진화하며 세대들을 발전시킬 수 있다. 유전 오퍼레이션은 임의의 정지 조건이 만족될 때까지 수행된다. 정지 조건이 만족되면 해 집단에 남은 해 중 재배치 거리가 가장 짧은 해 집단이 가장 바람직한 해 집단이 된다. 이로써 유전 알고리즘을 통해 염색체에 포함된 재배치 페어들 중 더 작은 이동 거리를 갖는 염색체를 최적해로 선택할 수 있게 된다.

적합도 함수는 유전 알고리즘에 의해 산출된 개개의 해가 알고리즘의 궁극적 목적에 얼마나 근접한 것인지에 대해 평가에 사용되는 목적함수(objective function)이다. 재배치 스케줄링에서의 적합도 함수는 재배치되어야 하는 차량들의 재배치 거리를 기준으로 한다. 적합도 함수는 동일한 오버플로 스테이션과 동일한 언더플로 스테이션을 갖는 재배치 쌍들을  $2(S_1, S_2)$ 와 같이 병합한다 [Kim et al. 2014].



### 3. 전력 수요 예측

전력 사용이 증가는 전력 계통을 위협하고 특히 이로 인해 발생하는 블랙아웃은 국가 경제에도 큰 손실을 가져온다. 따라서 정전사태를 예방하고 효율적이고 안정적인 전력 계통을 운용하기 위해서는 전력 수요에 대한 보다 정확한 예측이 무엇보다 중요하다.

이에 따라 일별, 주별, 계절별 전력 수요 예측에 다양한 통계적 방법을 적용한 많은 연구가 이루어지고 있다. [Erodogdu 2006]은 공적분 분석(Cointegration Analysis)과 ARIMA 모형을 이용해 터키의 연도별 전력 수요를 예측하였고, [Beccali et al. 2004]은 인공 신경망 기법을 사용하여 이탈리아 Palermo 지역의 하루 24시간의 전력 수요를 예측하였다.

[Shin et al. 2016]은 이중 계절성 Holt-Winters 모형과 TBATS 모형을 이용하여 외부 요인의 영향을 가장 적게 받는 특정 시간대의 일 단위 전력 수요량을 예측하고 모형의 성능을 비교 분석하였다.

[Park et al. 2014]은 자귀회귀모형과 뉴로-퍼지 모델로 구성된 하이브리드형태의 일별 최대 전력수요 예측 알고리즘을 개발하였다.

전력 수요는 계절, 주간, 일간 주기와 같은 시계열 데이터에 해당한다. 시계열 데이터는 종속성 즉, 시간에 따라 관측된 자료들이 서로 의존적인 관계가 존재한다. 따라서 시계열 분석을 통해 과거 관측 값을 토대로 미래에 관측될 값을 추측하여 전력 수요에 대한 의미 있는 예측할 수 있다 [Barak et al. 2016]. 미래의 수요를 예측하는 시계열 분석에는 회귀분석(regression analysis), AR(Autoregressive), MA(Moving Average), ARMA(Autoregressive moving average), 지수평활법(exponential smoothing), 인공 신경망 기법(Artificial Neural Network) 등이 있다.

비정상성 시계열은 추세나 계절성을 갖고 있다. 추세는 시간이 지남에 따라 증가, 감소, 유지 등의 형태로 변화하는 경향이고, 계절성은 연간, 일간, 시간당 주기적으로 반복되는 패턴이다. 이러한 비정상성 시계열은 예측의 정확도가 떨어지기 때문에 정상성의 시계열로 변환하여 예측의 정확도를 높여야 한다.

Box-Jenkins(1976)의 ARIMA 모형과 계절 ARIMA 모형은 AR과 MA 모형을 결합한 형태로 차분을 취해 안정적 시계열로 변환시킨다. ARIMA 모형은 ARIMA(p, d, q)로 표기하는데 이는 시계열이 d번 차분을 하고 난 후 안정적인 ARIMA(p, q)모형이 됨을 의미한다. 여기서 p는 AR의 차수, d는 차분의 수, q는 MA의 차수를 나타낸다.

시계열의 목적은 t기 시점에 대응되는  $Z_t$  확률 변수의 실현 값들을 가지고 그것을 생성시키는 시계열 모형을 추정하고, 그 결과를 이용해 예측하는데 있다.

Figure 13은 Box-Jenkins의 ARIMA 모형의 절차를 보여준다.

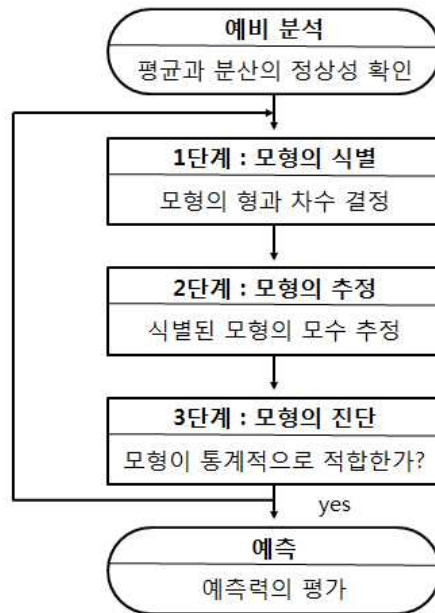


Figure 13. ARIMA 구축의 흐름도

(1) 예비 분석

시계열 분석을 위해 ARIMA 모형을 구축하고자 할 때는 먼저 시계열 도표와 자기상관함수(ACF; AutoCorrelation Function)의 그래프를 통해 시계열 자료의 정상성 여부를 확인하여 추세는 차분, 계절성은 계절 차분을 통해 안정 시계열로 변환한다. 만약 분산이 시간의 변화에 따라 변한다면 대수변환과 제곱근 변환을

통해 분산의 불안정성을 제거할 수 있다.

## (2) 모형의 식별

모형의 식별이란 시계열의 여러 모형 중에서 분석하려는 시계열 데이터를 생성시켰다고 생각되는 가장 적합한 모형의 형과 그것의 차수를 결정하는 것을 말한다. 즉 이 단계에서는 차분이나 대수변환을 통해 구해진 안정 시계열에 대해 AR과 MA의 차수  $p, q$ 를 정하여 여러 개의 ARIMA 모형 중 적당하다고 판단되는 모형을 임시적으로 선택한다.  $p$ 와  $q$ 의 차수는 자기상관함수(Autocorrelation function, ACF)와 편자기상관함수(PACF; Partial AutoCorrelation Function)를 비교·분석하여 결정하게 된다. 이 때 절약의 원리에 의해 가능한 낮은 차수를 선정하며 보조적인 역할로 Akaike의 AIC와 Schwarz의 BIC가 적은 모형을 선택한다.

## (3) 모형의 추정

모형 식별에서 선택된 모형의 파라미터를 추정하는 단계로 추정된 모수가 통계적으로 유의한지 여부를 결정할 수 있다.

## (4) 모형의 진단

모형의 진단은 추정된 모형이 타당한 모형인지 아닌지를 판단하는 과정으로 선택된 모형을 최종적으로 선택하는 단계이다. 만일 타당한 모형이 아니라고 판단될 경우 식별과 추정, 진단을 반복적으로 시행하여 최종 모형을 선택한다.

## (5) 예측

시계열 모형에 의한 분석의 목적은 예측에 있는데 예측능력이 떨어지면 유용한 모형이 될 수 없다. 예측치와 실제 관측치의 차이를 비교하여 예측 오차(forecast error)가 작을수록 예측력이 높다고 볼 수 있다.

예측력의 평가 기준은 평균제곱오차(MSE), 제곱근 평균제곱오차(RMSE), 퍼센트 오차(PE), 평균 퍼센트오차(MPE), 평균 절대퍼센트오차(MAPE), 평균절대오차(MAE)등이 있지만 일반적으로 MSE와 RMSE, MAPE를 사용한다.

### III. 전력 수요 예측

전력 수요 예측을 위해 본 논문에서 사용된 데이터는 Figure 14와 같이 한국전력으로부터 다운로드한 제주대학교의 전력 사용량으로 2008년 1월 1일부터 2016년 6월 1일까지 15분 단위로 측정된 시계열 자료(단위: kwh)이다. 이 자료에 대한 분석은 ARIMA 모형을 사용하였으며 전력 사용량의 월별 예측과 15분별 1일, 70일 예측으로 나누어 예측값을 구하고 모형의 성능을 분석하였다.

시	사용량 (kWh)	최대수요 (kW)	무요전력 (kWh/h)	Cb (10%)	역률 (%)	시	사용량 (kWh)	최대수요 (kW)	무요전력 (kWh/h)	Cb (10%)	역률 (%)				
0015	475.56	1,902.24	185.84	0.00	0.22	97.61	100.00	1215	672.84	2,691.36	148.68	0.00	0.31	97.64	100.00
0030	464.40	1,857.60	181.16	0.00	0.21	97.71	100.00	1230	673.92	2,695.68	149.04	0.00	0.31	97.64	100.00
0045	468.36	1,873.44	186.20	0.00	0.21	97.52	100.00	1245	695.52	2,762.08	149.40	0.00	0.32	97.77	100.00
0100	460.80	1,843.20	182.60	0.00	0.21	97.61	100.00	1300	690.12	2,760.48	150.48	0.00	0.32	97.70	100.00
0115	473.04	1,892.16	186.92	0.00	0.22	97.54	100.00	1315	710.64	2,842.56	155.52	0.00	0.33	97.69	100.00
0130	457.92	1,831.68	184.76	0.00	0.21	97.48	100.00	1330	730.08	2,920.32	157.32	0.00	0.33	97.76	100.00
0145	459.76	1,837.44	182.60	0.00	0.21	97.60	100.00	1345	719.28	2,877.12	144.36	0.00	0.33	98.04	100.00
0200	455.76	1,823.04	99.72	0.00	0.21	97.69	100.00	1400	711.36	2,845.44	149.76	0.00	0.33	97.85	100.00
0215	466.20	1,864.80	187.64	0.00	0.21	97.44	100.00	1415	723.24	2,892.96	150.84	0.00	0.33	97.89	100.00
0230	447.84	1,791.36	181.88	0.00	0.21	97.51	100.00	1430	717.48	2,869.92	154.80	0.00	0.33	97.75	100.00
0245	448.92	1,795.68	183.68	0.00	0.21	97.44	100.00	1445	723.60	2,894.40	154.80	0.00	0.33	97.79	100.00
0300	462.08	1,768.32	182.24	0.00	0.20	97.43	100.00	1500	737.64	2,950.56	156.24	0.00	0.34	97.83	100.00
0315	444.60	1,778.40	186.20	0.00	0.20	97.26	100.00	1515	746.64	2,986.56	157.32	0.00	0.34	97.85	100.00
0330	442.80	1,771.20	183.32	0.00	0.20	97.28	100.00	1530	736.92	2,947.88	153.36	0.00	0.34	97.90	100.00
0345	446.04	1,784.16	185.84	0.00	0.20	97.30	100.00	1545	725.76	2,903.04	144.36	0.00	0.33	98.08	100.00
0400	435.24	1,740.96	185.48	0.00	0.20	97.19	100.00	1600	701.64	2,806.56	139.32	0.00	0.32	98.09	100.00
0415	441.36	1,765.44	182.96	0.00	0.20	97.39	100.00	1615	692.64	2,770.56	134.28	0.00	0.32	98.17	100.00
0430	430.92	1,723.68	181.52	0.00	0.20	97.34	100.00	1630	714.60	2,888.40	141.84	0.00	0.33	98.09	100.00
0445	439.20	1,756.80	183.32	0.00	0.20	97.34	100.00	1645	696.60	2,786.40	138.24	0.00	0.32	98.09	100.00
0500	429.84	1,719.36	98.28	0.00	0.20	97.48	100.00	1700	700.92	2,803.68	138.96	0.00	0.32	98.09	100.00
0515	440.64	1,762.56	182.96	0.00	0.20	97.38	100.00	1715	685.08	2,740.32	137.52	0.00	0.31	98.04	100.00
0530	429.48	1,717.92	97.56	0.00	0.20	97.52	100.00	1730	675.00	2,700.00	132.12	0.00	0.31	98.14	100.00
0545	434.88	1,738.52	99.36	0.00	0.20	97.49	100.00	1745	666.36	2,665.44	126.72	0.00	0.31	98.24	100.00
0600	437.40	1,749.60	98.64	0.00	0.20	97.55	100.00	1800	641.16	2,564.64	122.04	0.00	0.29	98.24	100.00
0615	430.92	1,723.68	181.16	0.00	0.20	97.35	100.00	1815	615.24	2,460.96	122.40	0.00	0.28	98.08	100.00
0630	434.88	1,738.52	99.00	0.00	0.20	97.51	100.00	1830	590.04	2,360.16	122.40	0.00	0.27	97.92	100.00
0645	438.12	1,752.48	99.72	0.00	0.20	97.51	100.00	1845	583.92	2,335.68	118.44	0.00	0.27	98.00	100.00
0700	443.16	1,772.64	98.48	0.00	0.20	97.71	100.00	1900	590.40	2,351.60	124.56	0.00	0.27	97.85	100.00

Figure 14. 한국전력의 iSMART가 제공하는 전력 수요 데이터

첫 번째 분석에는 2008년 1월부터 2014년 12월까지의 자료를 사용하여 훈련을 위한 모형을 구축하고 ARIMA 모형을 통해 2015년 1월부터 2015년 12월까지의 전력 사용량을 예측하고 모형의 적합도를 평가하였다.

두 번째 분석에는 2016년 1월 1일부터 2016년 6월 1일까지의 전력 사용량 14,688 개를 가지고 모형 구축에 쓰이는 데이터를 변화시켜 가면서 15분 후, 1시간 후,

2시간 후의 전력 사용량을 1일과 70일치, 요일에 따른 전력 사용량을 예측하고 모형의 적합도를 평가하였다.

## 1. 전력 사용량의 월별 예측

2008년 1월부터 2015년 12월까지의 96개의 월별 전력 사용량 자료를 이용하였다. 자료는 R의 ts함수를 통해 데이터를 시계열 객체로 저장한 후, 시계열 자료의 부분을 선택하게 하는 window함수를 통해 2008년 1월부터 2014년 12월까지의 84개의 자료는 모형 적합에 이용하고, 나머지 2015년 1월부터 2015년 12월까지의 12개의 자료는 모형을 평가하는데 사용하였다.

2008년 1월부터 2014년 12월까지의 전력 수요 자료에 대한 시계열 그래프는 Figure 15와 같다. 일반적으로 arima함수를 통해 모의 모형들을 구한 후 ACF와 PACF를 통해 모형을 식별하지만 auto.arima() 함수는 이를 생략하고 적합한 모형의 모수를 자동적으로 추정해준다. 따라서 auto.arima() 함수를 통해 ARIMA(p, d, q) 모형을 추정한다.

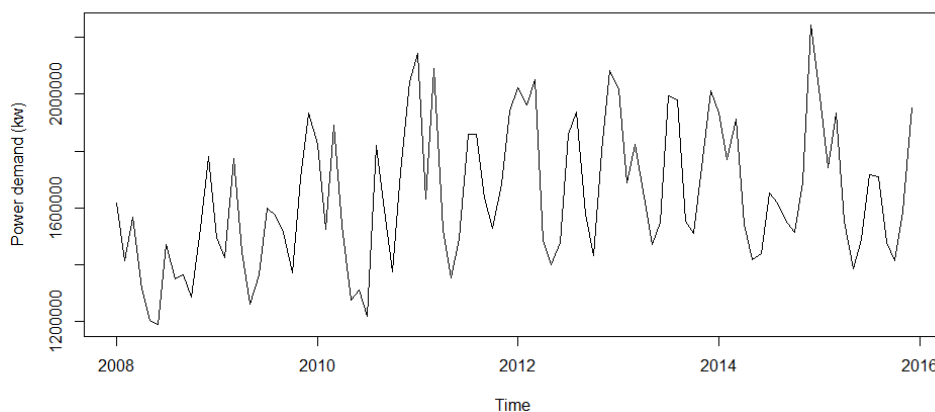


Figure 15. 2008년 1월부터 2014년 12월까지의 월별 데이터

auto.arima() 함수를 실행한 결과 Figure 16과 같이 ARIMA(0, 1, 1) 모형의 대수 우도(log likelihood) 및 AIC 식별 통계량은 각각 -945.62, 1897.24로 나타났으며 예측의 적합척도인 RMSE는 131832.9, MAPE는 5.392902로 나타났다.

```
ARIMA(0,1,1)(1,1,0)[12]
Coefficients:
      ma1      sar1
    -0.8666  -0.4099
s.e.    0.0634   0.1102

sigma^2 estimated as 2.116e+10: log likelihood=-945.62
AIC=1897.24  AICC=1897.6  BIC=1904.03

Training set error measures:
      ME      RMSE      MAE      MPE      MAPE      MASE      ACF1
Training set -8669.945 131832.9 89902.22 -0.8827408 5.392902 0.7285504 0.1359191
```

Figure 16. auto.arima() 함수의 결과

auto.arima() 함수는 최선의 차수가 (0, 1, 1)라고 결정하였다. 즉 MA 계수 1개 (q=1)가 있는 모형을 선택하였고, 그전에 데이터를 한번 차분(d=1)했다는 것이다. 이후, 선택된 모형인 ARIMA(0, 1, 1)을 가지고 predict() 함수를 통해 예측값을 구한다. auto.arima() 함수는 ARIMA 모형을 만들 때 가장 적당한 p, d, q를 갖는 조합을 일일이 찾지 않고 바로 적합한 모수를 결정해 주기 때문에 굳이 선택된 모형의 적합도를 검정하지 않아도 된다. 그러나 추가적으로 tsdiag() 함수를 통해 추정된 시계열 모형에 대한 적합도를 검정하였다. Figure 17과 같이 자기상관 함수가 잔차들 사이에 유의미한 자기상관을 보이지 않고 있으며 Ljung-Box 카이제곱 검정 결과 p-값이 모두 커서 잔차들에 패턴이 없다는 사실을 나타낸다.

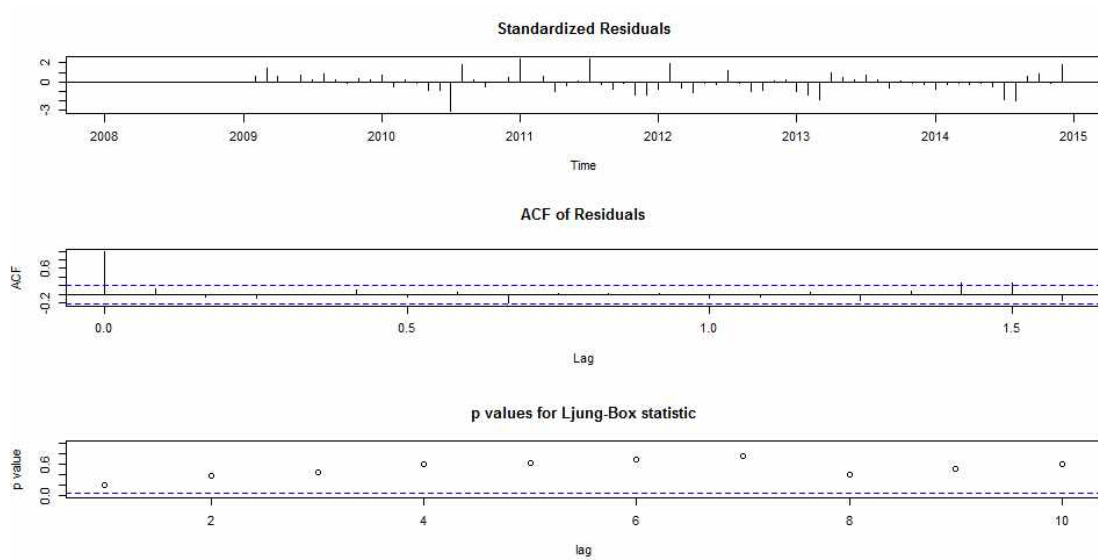


Figure 17. Diagnostic Display for the ARIMA(0, 1, 1)

Table 3은 2015년 1월부터 12월까지의 실제값, 예측값, 실제값과 예측값의 차이를 보여주며 Figure 18은 실제값과 예측값을 보여주는 그래프이고, Figure 19와 Figure 20은 predict()함수와 forecast() 함수를 통해 예측한 값과 이에 대한 결과 그래프이다. predict() 함수의 결과값 pred와 forecast() 함수의 결과값으로 도출된 mean 변수는 어차피 동일하지만 forecast() 함수는 80%와 95% 신뢰구간에 해당하는 예측값을 추가로 보여준다. 이 때 예측의 정확도는 2015년 4월의 경우 최고 99.8%로 실제값과 예측값이 거의 일치함을 보이고 있다.

Table 3. 2015년도 월별 전력 수요량 (단위: kwh)

월별 (time)	실제값 (Actual)	예측값 (Forecast)	실제값-예측값 (Actual-Forecast)
2015.01	1989680	1935460	54220
2015.02	1740667	1705168	35499
2015.03	1934683	1844266	90417
2015.04	1554235	1551229	3006
2015.05	1384444	1409572	-25128
2015.06	1490066	1453312	36754
2015.07	1716393	1762967	-46574
2015.08	1710887	1735470	-24583
2015.09	1478107	1525070	-46963
2015.10	1415564	1481541	-65977
2015.11	1558808	1694497	-135689
2015.12	1952571	2116884	-164313

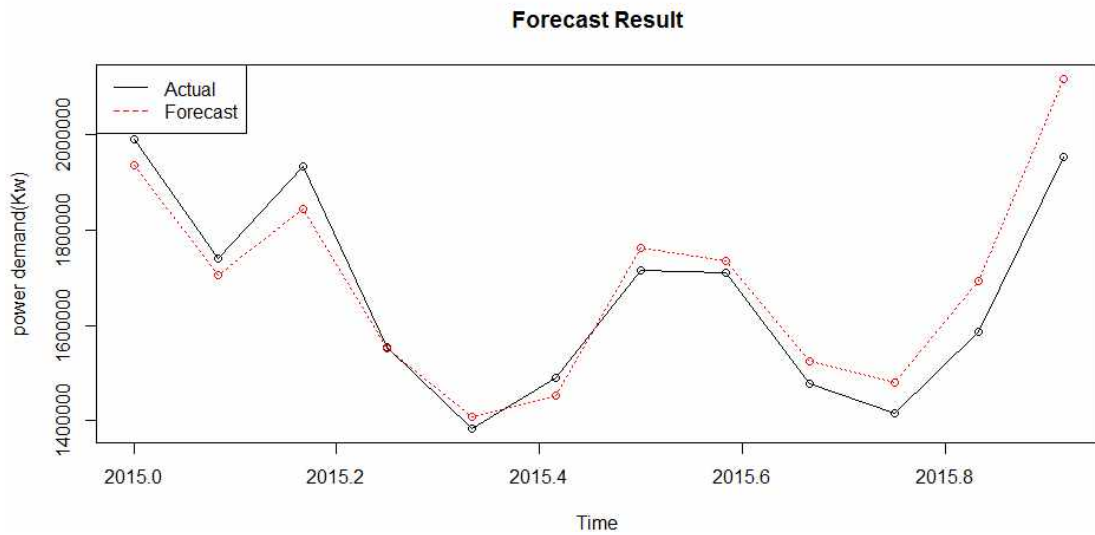


Figure 18. 2015년도 월별 전력 수요량 예측 결과 그래프



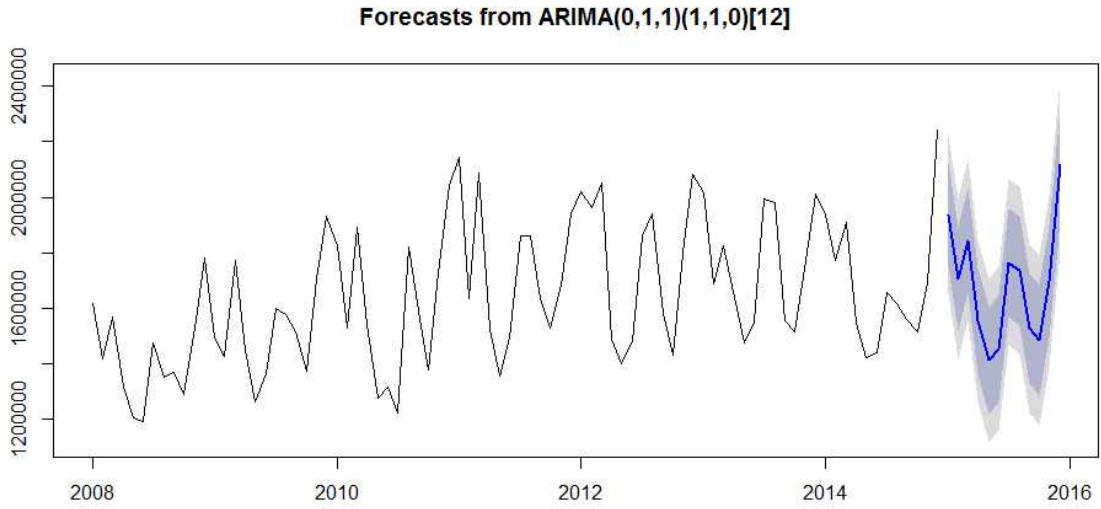


Figure 19. forecast() 함수를 통한 예측 결과 그래프

```

RStudio
File Edit Code View Plots Session Build Debug Tools Help
Go to file/function Addins Project: (None)
Source
Console
> mon<-read.csv("d:a2.csv", header=F)
> mon.ts<-ts(mon$V1, start=c(2008, 1), frequency=12)
> str(mon.ts)
Time-Series [1:96] from 2008 to 2016: 1615917 1414500 1566855 1313203 1203576 ...
> mon.ts
      Jan  Feb   Mar   Apr   May   Jun   Jul   Aug   Sep   Oct   Nov   Dec
2008 1615917 1414500 1566855 1313203 1203576 1191096 1471794 1350903 1364811 1287207 1527822 1781061
2009 1494893 1425583 1772507 1446926 1260569 1364002 1598240 1573390 1513751 1373289 1703561 1932325
2010 1823721 1524485 1890586 1539263 1274725 1311520 1219655 1818101 1590610 1374232 1728256 2042742
2011 2144118 1632282 2088874 1523974 1354751 1493063 1859704 1859256 1635539 1527060 1683377 1939288
2012 2020618 1963087 2051238 1486405 1399656 1477151 1860198 1937798 1578452 1430758 1799785 2083414
2013 2013617 1687160 1823448 1638239 1471494 1549384 1994717 1980084 1553556 1511451 1772852 2011179
2014 1933371 1769880 1910930 1542989 1418760 1438778 1654181 1617748 1557485 1512966 1692270 2242516
2015 1989680 1740667 1934683 1554235 1384444 1490066 1716393 1710887 1478107 1415564 1588008 1952571
> zt<-window(mon.ts, c(2008,1), c(2014, 12))
> fit<-auto.arima(zt)
> pr<-forecast(fit, 12)
> pr
      Point Forecast   Lo 80   Hi 80   Lo 95   Hi 95
Jan 2015    1935460 1749048 2121873 1650367 2220554
Feb 2015    1705168 1517103 1893233 1417547 1992788
Mar 2015    1844266 1654563 2033969 1554140 2134391
Apr 2015    1551229 1359902 1742556 1258619 1843838
May 2015    1409572 1216635 1602509 1114500 1704644
Jun 2015    1453312 1258778 1647847 1155798 1750827
Jul 2015    1762967 1566848 1959085 1463029 2062904
Aug 2015    1735470 1537780 1933160 1433129 2037810
Sep 2015    1525070 1325821 1724319 1220345 1829795
Oct 2015    1481541 1280745 1682336 1174450 1788631
Nov 2015    1694497 1492166 1896828 1385059 2003935
Dec 2015    2116884 1913030 2320738 1805116 2428652
  
```

Figure 20. forecast() 함수를 통한 예측값

## 2. 전력 사용량의 15분별 1일 예측

Figure 21은 2016년 4월 1일부터 2016년 6월 1일까지의 전력 사용량 그래프이다.

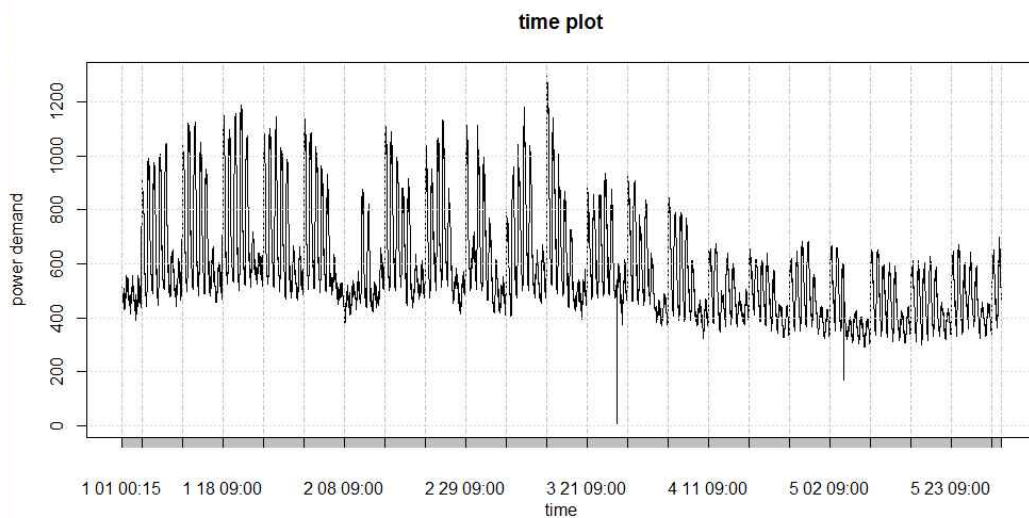


Figure 21. 2016년 4월 1일부터 2016년 6월 1일까지의 전력 사용량

2016년 4월 1일부터 2016년 6월 1일까지의 전력 사용량 8,640개를 가지고 15분 후, 1시간 후, 2시간 후의 전력 수요를 예측하였다.

Figure 22는 훈련데이터의 윈도우를 이동해 가며 `auto.arima()` 함수를 통해 적절한 모수를 갖는 모형을 찾고, 여기서 얻어진 그 모형을 `forecast()` 함수로 예측값을 생성해주는 R 프로그램의 코드를 보여 준다 [Lee et al. 2016].

```

RStudio
File Edit Code View Plots Session
Source
Console ~/
> for(i in 1:96){
+ st<-i
+ ed<-i+5759
+ mv<-v[st:ed]
+ myts<-ts(mv)
+ fit<-auto.arima(myts)
+ pr<-forecast(fit, 12)
+ hq[i]<-pr$mean[1]
+ h1[i+3]<-pr$mean[4]
+ h2[i+7]<-pr$mean[8]
+ h3[i+11]<-pr$mean[12]
+ }

```

Figure 22. 모형 구축을 위한 코드

학습 데이터는 5,760개 즉, 2개월의 데이터를 가지고 다음날 하루의 데이터를 예측하고자  $i$ 는 for문을 통해 1부터 96까지의 반복한다. `forecast()` 함수의 파라미터의 값이 12이기 때문에 원래는 `mean[1]`부터 `mean[12]`까지 12개의 예측값이 생성되는데 for문으로 인해 `mean[1]`부터 `mean[12]` 각각 96개씩 생성된다. 전력 데이터의 주기는 15분이므로 `mean[1]`은 15분 후 예측값, `mean[4]`는 60분(1시간) 후 예측값, `mean[8]`은 120분(2시간) 후 예측값, `mean[12]`는 180분(3시간) 후 예측값이 된다.

즉,  $i$ 가 1부터 96까지이므로 모형에 쓰이는 훈련 데이터 `myts`의 윈도우가 하나씩 이동하면서 `mean[1]`부터 `mean[12]`까지의 예측 값의 세트도 96개씩 이동하며 생성된다. 따라서 `hq`, `h1`, `h2`, `h3` 각각에는 96개의 예측 결과값이 존재하게 된다.

이를 실제 데이터 `act`와 비교하여 예측이 잘 되었는지 그래프를 생성하여 시각화한다. 또한 아래의 예측 오차 수식을 통해 최대 예측 오차와 평균 예측 오차를 구하여 비교한다.

$$prediction\ error = \frac{|actual\ value - predicted\ value|}{actual\ value} \quad (7)$$

Table 4. 2016년 6월 1일 전력 수요량 예측 (단위: kwh)

15분별 (time)	실제값 (Actual)	예측값 (Forecast)	실제값-예측값 (Actual-Forecast)	예측 오차 (Prediction error)
00:00:00	460.44	456.0137	4.426259	0.009613
00:15:00	445.32	435.8708	9.44921	0.021219
00:30:00	433.08	425.9838	7.09619	0.016385
00:45:00	434.16	429.0371	5.122872	0.0118
01:00:00	419.76	415.8374	3.922573	0.009345
01:15:00	421.2	416.5868	4.613223	0.010953
01:30:00	421.92	421.778	0.141972	0.000336
1:45:00	415.08	411.682	3.398014	0.008186
2:00:00	410.4	408.3907	2.009268	0.004896
2:15:00	408.96	405.2541	3.705862	0.009062
2:30:00	403.56	400.5914	2.968595	0.007356
2:45:00	402.12	399.3979	2.722078	0.006769
3:00:00	399.6	398.6714	0.928638	0.002324
3:15:00	401.76	400.641	1.119032	0.002785
3:30:00	392.04	390.2662	1.773775	0.004524
3:45:00	393.48	390.8032	2.676848	0.006803
4:00:00	396	397.4941	-1.49408	0.003773
4:15:00	389.16	386.9085	2.251539	0.005786
4:30:00	386.28	384.7049	1.575141	0.004078
4:45:00	386.64	384.4785	2.161536	0.005591
5:00:00	385.2	385.1514	0.048596	0.000126
5:15:00	374.4	369.9751	4.424854	0.011819
5:30:00	364.32	358.6527	5.667337	0.015556
5:45:00	377.28	377.2378	0.042199	0.000112

Table 4는 2016년 6월 1일 전력 수요량 예측의 일부를 나타낸다. 실제값과 예측값의 차이는 -18.5867kwh부터 15.06012kwh까지 존재하며 하루 96개 중에서 89개의 경우가 실제값과 예측값의 차이가 거의 10kwh 이내의 차이를 보인다. 즉

모형을 통해 예측한 값의 약 90%는 실제값과 거의 10kwh밖에 나지 않는다고 볼 수 있다. 5kwh 이내 차이도 약 74%에 해당한다. 수식에 의해 구한 예측 오차는 Figure 23과 같으며 최대 예측 오차는 0.03484이고, 평균 예측 오차는 0.00907이다.

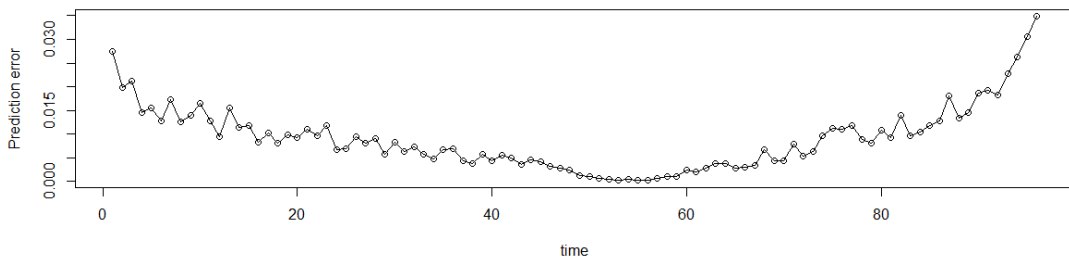


Figure 23. 15분 후 전력 수요 예측 오차 그래프

전력 응용 서비스마다 전력 수요 예측 모델을 통해 특정 시간대의 전력 수요를 예측을 하려는 목적이 다를 수 있다. 예를 들어 전력을 사용할 것인지의 여부, 전기 자동차를 충전할 것인지의 여부를 결정하는데에는 15분 후의 전력 수요 예측 값이 필요로 되어질 수 있다. 또한 V2G 스케줄러의 경우는 가능한 많은 전기 자동차와 접촉하기 위해서는 시간적 여유가 있기 때문에 15분 후의 전력 수요 예측보다 2 ~ 3시간 후의 전력 수요 예측값을 선호할 수 있다.

Figure 24는 다음 날의 15분, 1시간, 2시간, 3시간 후의 전력 수요 예측을 그래프로 나타낸 것이다. 15분, 1시간, 2시간, 3시간 후 4개의 경우에 대한 실제값-예측값을 비교해 보면 최대가 18.6kwh, 79.6kwh, 114.1kwh, 159.8kwh이고, 최소가 4.6kwh, 15.1kwh, 33.4kwh, 50.9kwh이다. 예측 범위가 넓어질수록 실제값과 예측값의 차가 점점 증가하고 있지만 예측 데이터는 실제 전력 소비 데이터를 어느 정도 잘 따라가고 있다.

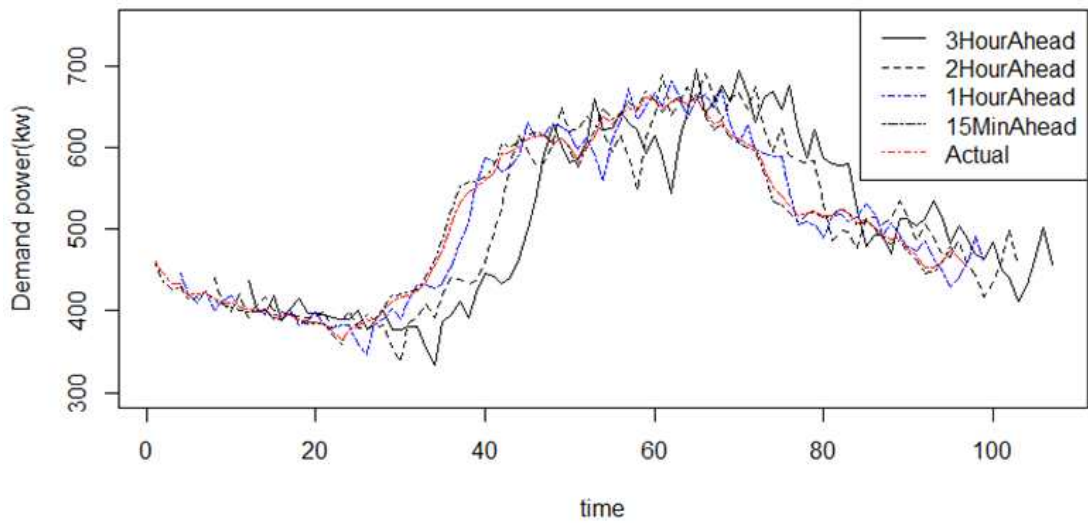


Figure 24. 15분별 1일치 전력 수요 예측

Figure 25와 Table 5는 15분별 1일치 전력 수요 예측 오차에 대한 그래프와 표이다. 예측 오차는 예측 범위가 넓어질수록 커지고 있으며, 15분 후는 최대 0.035, 3시간 후는 최대 0.281이다.

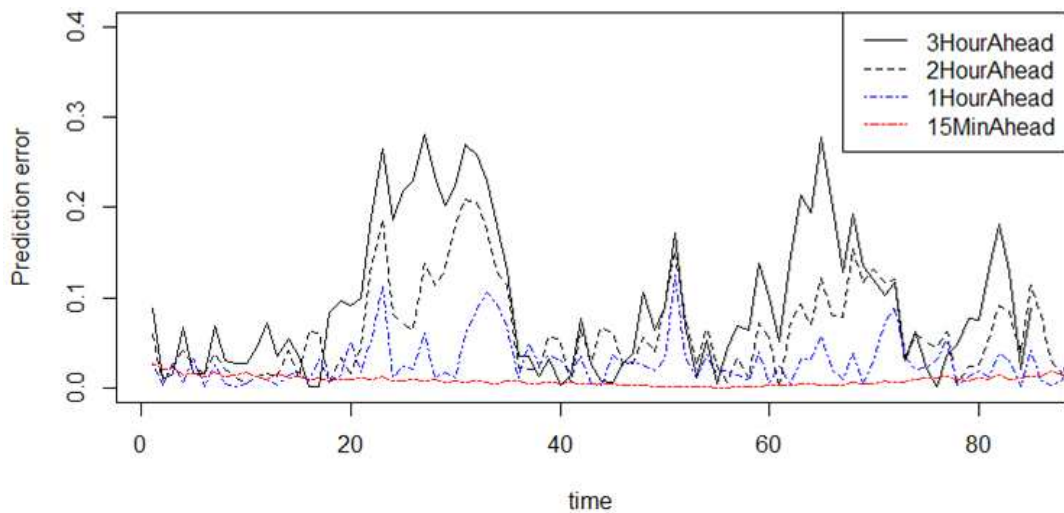


Figure 25. 15분별 1일치 전력 수요 예측 오차

Table 5. 전력 수요 예측 오차 (15분별)

	최소	최대
15분 후	0.009	0.035
1시간 후	0.029	0.125
2시간 후	0.063	0.209
3시간 후	0.096	0.281

### 3. 전력 사용량의 15분별 70일 예측

이번 실험에서는 전력 사용량의 1일 기반 예측을 조금 더 확장하여 70기반에 대한 최대 전력 수요와 평균 전력 수요를 예측해보고, 이에 대한 예측오차들을 비교 분석하였다.

Figure 26부터 Figure 29까지는 15분 후, 1시간 후, 2시간 후, 3시간 후의 예측을 70일치(96\*70) 즉 6720분까지 나타낸 그래프들이다. Figure 26을 보면 15분 후의 전력 수요 예측은 실제와 거의 일치함을 보이고 있고, 1시간 후 정도까지는 예측이 실제값(Actual)을 거의 따라가고 있으나 2시간 후에서 3시간 후로 갈수록 예측의 결과가 실제값을 점점 따라가지 못하고 있음을 보인다.

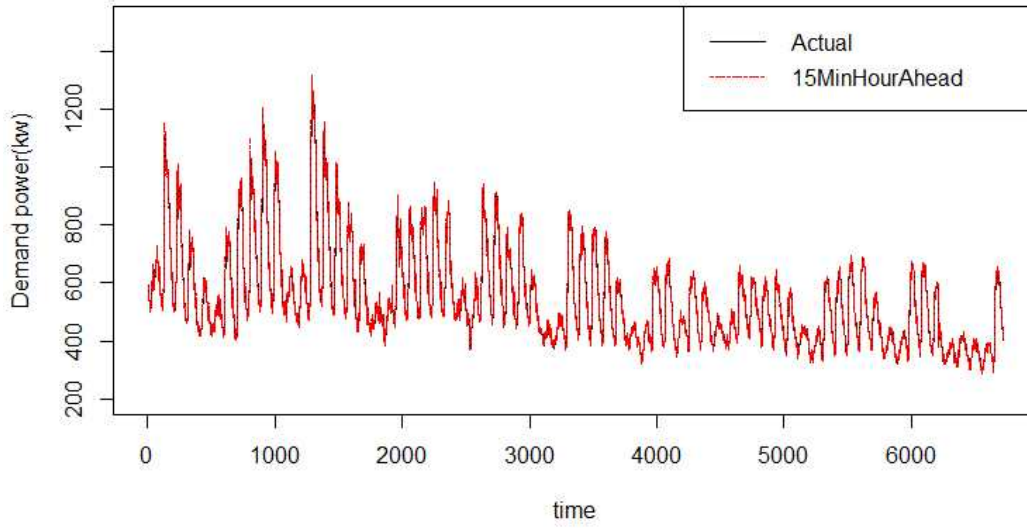


Figure 26. 70일치 15분 후 전력 수요 예측

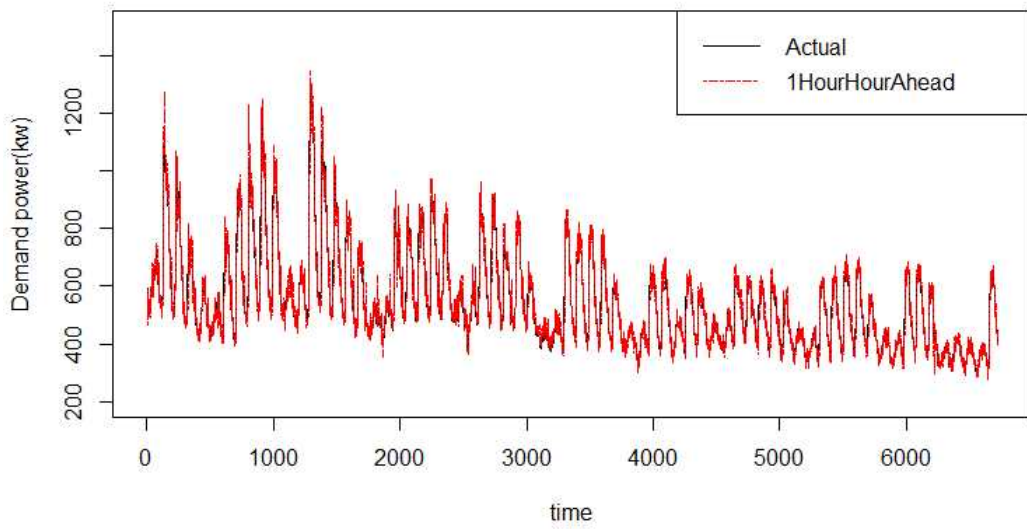


Figure 27. 70일치 1시간 후 전력 수요 예측



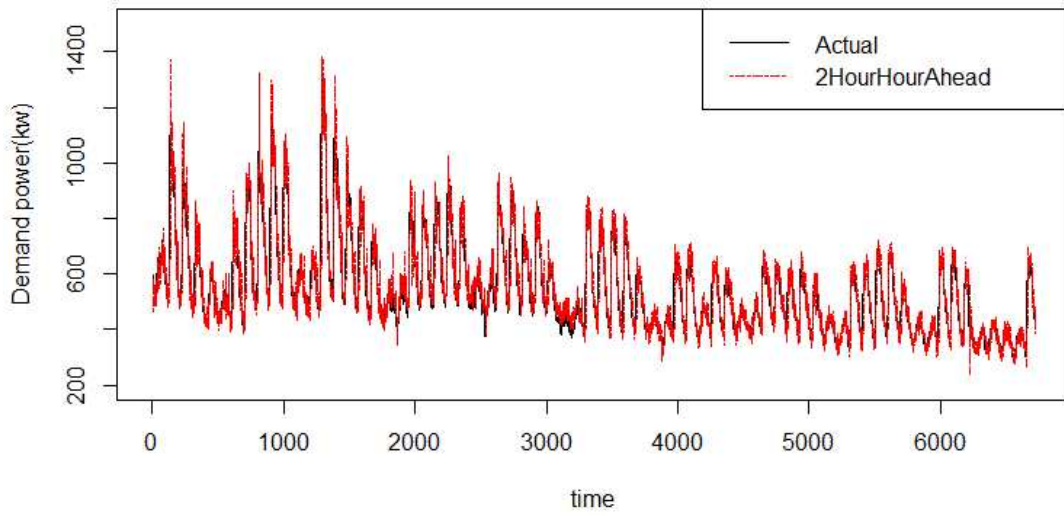


Figure 28. 70일치 2시간 후 전력 수요 예측

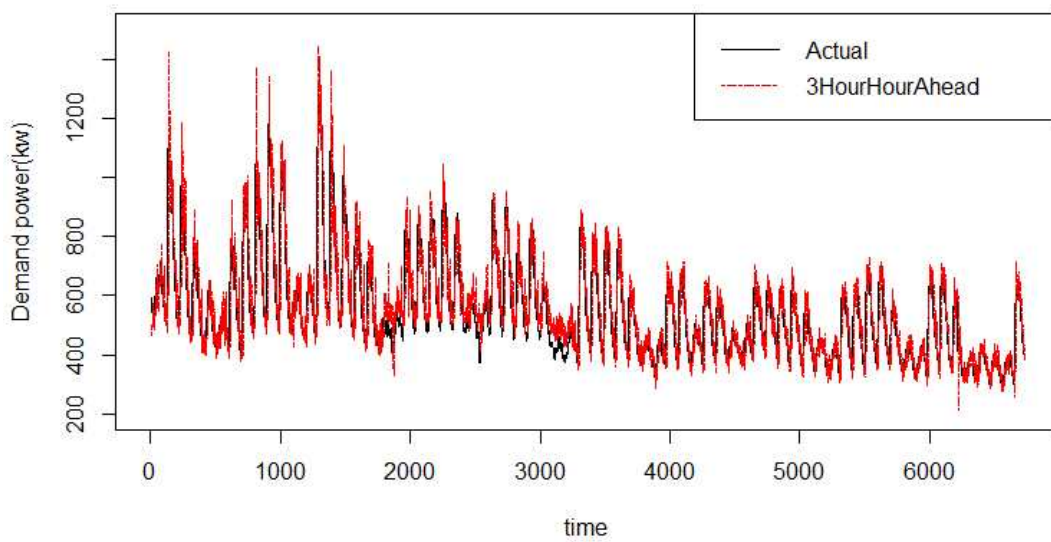


Figure 29. 70일치 3시간 후 전력 수요 예측

Table 6은 15분 후, 1시간 후, 2시간후, 3시간 후의 실제값과 예측값의 최소와 최대 차이를 비교한 것이다.

Table 6. 전력 수요의 실제값과 예측값의 차(70일치 15분별)

	최소	최대
15분 후	0.0008kwh	237kwh
1시간 후	0.006kwh	195kwh
2시간 후	0.003kwh	459kwh
3시간 후	0.007kwh	610kwh

Figure 30과 Table 7은 각각 70일치의 최대 전력 수요 예측에 대한 그래프와 최대 전력 수요의 실제값과 예측값의 차이를 보여준다. 70일치 최대 전력 수요 예측 결과 15분 후가 최소 1.6kwh에서 최대 54.5kwh로 나타나고, 3시간 후는 최소 14.0kwh에서 최대 325.8kwh이다. 예측의 범위가 넓어질수록 예측값은 실제값의 그래프 형태와의 차이를 보인다.

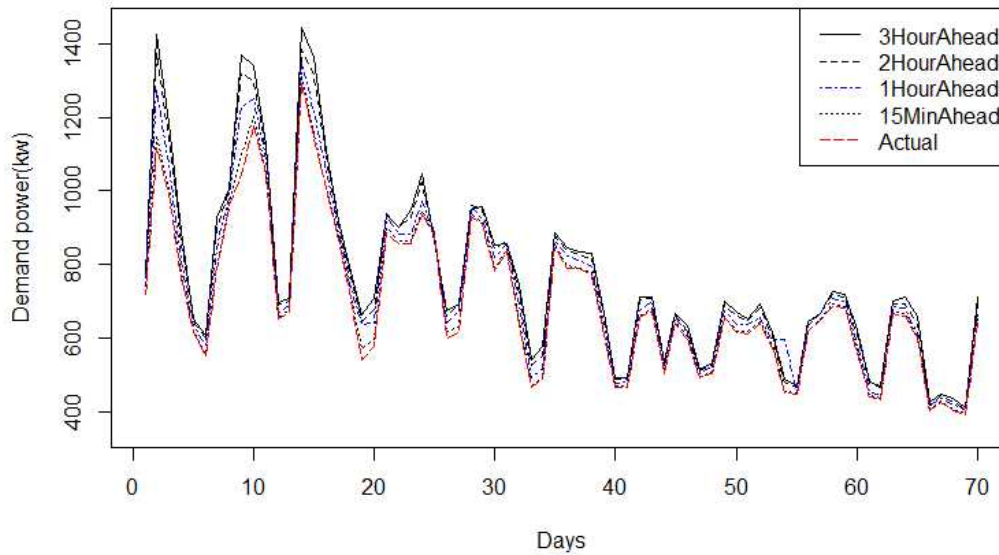


Figure 30. 70일 최대 전력 수요 예측

Table 7. 최대 전력 수요의 실제값과 예측값의 차 (70일치 15분별)

	최소	최대
15분 후	1.6kwh	54.5kwh
1시간 후	8.7kwh	184.9kwh
2시간 후	2.9kwh	278.5kwh
3시간 후	14.0kwh	325.8kwh

Figure 31은 70일치 평균 전력 수요 예측을 보여준다. 약 15일부터 35일까지만을 제외하고는 15분 후, 1시간 후, 2시간 후, 3시간 후의 그래프는 실제값의 그래프와 거의 일치함으로 보인다. 앞의 최대 전력 수요 예측과 비교해 볼 때, 평균 전력 수요 예측은 상당히 잘되어 Table 8에서 보듯이 실제값과의 차이가 최대 61kwh와 같이 비교적 적게 나타난다.

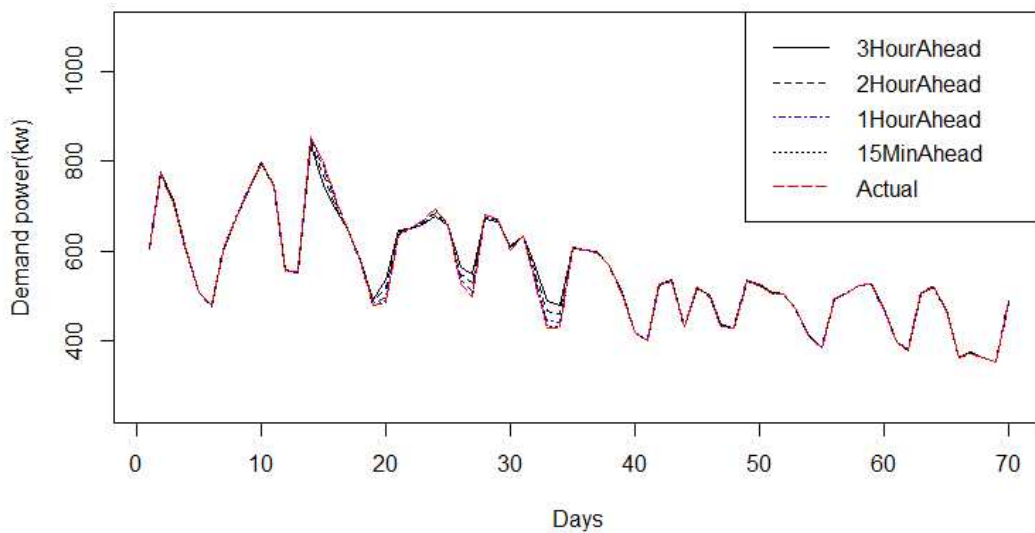


Figure 31. 70일 평균 전력 수요 예측

Table 8. 평균 전력 수요의 실제값과 예측값의 차 (70일치 15분별)

	최소	최대
15분 후	0.008kwh	3.1kwh
1시간 후	0.004kwh	14.5kwh
2시간 후	0.011kwh	37.0kwh
3시간 후	0.010kwh	61.0kwh

Figure 32와 Figure 33은 각각 요일별(7일)에 따른 최대 전력 수요 예측과 요일별(7일)에 따른 평균 전력 수요 예측에 대한 그래프이다. 두 개의 그래프에서 보듯이 토요일과 일요일은 평일에 비해 전력 수요가 적게 나타난다. 이는 실험에 쓰인 전력 데이터가 대학교이기 때문에 당연히 나타날 수 있는 특성이다.

Figure 32는 70일치 요일별에 따른 최대 전력 수요 예측으로서 요일에 따른 예측도 15분 후에서 3시간 후로 갈수록 예측력이 떨어지고 있다. 또한 그래프에서 보면 주말에 비해 평일이 훨씬 예측을 못하고 있다. Table 9에서 보듯이 실제값과 예측값과의 최대 차이도 모두 수요일에 나타나며 최소 차이는 주말에 나타나고 있다.

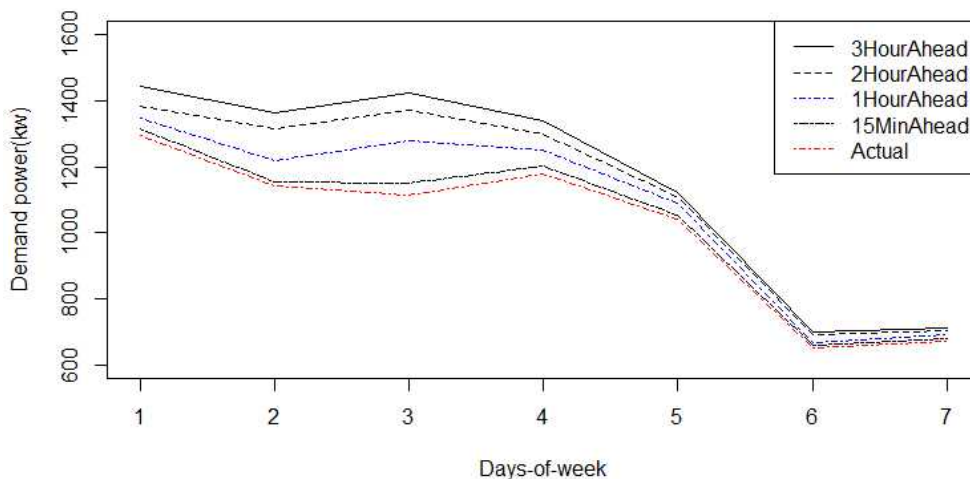


Figure 32. 요일별 최대 전력 수요 예측

Table 9. 최대 전력 수요의 실제값과 예측값의 차 (70일치 요일별)

	최소	최대
15분 후	4.69kwh(토)	34.78kwh(수)
1시간 후	14.96kwh(토)	161.30kwh(수)
2시간 후	31.17kwh(일)	256.42kwh(수)
3시간 후	39.30kwh(일)	307.42kwh(수)

Figure 33은 70일치 요일별에 따른 평균 전력 수요 예측에 대한 그래프를 나타내고, Table 10은 요일별 평균 전력 수요 예측의 실제값과 예측값의 차이를 보여준다. 요일별 평균 전력 사용량은 고르게 나타나고 있으며 요일별 최대 전력 수요 예측과는 반대로 주말의 예측력이 더 좋지 않게 나타난다. 그러나 모든 시간 범위의 그래프는 실제값의 그래프를 거의 잘 따라가고 있다.

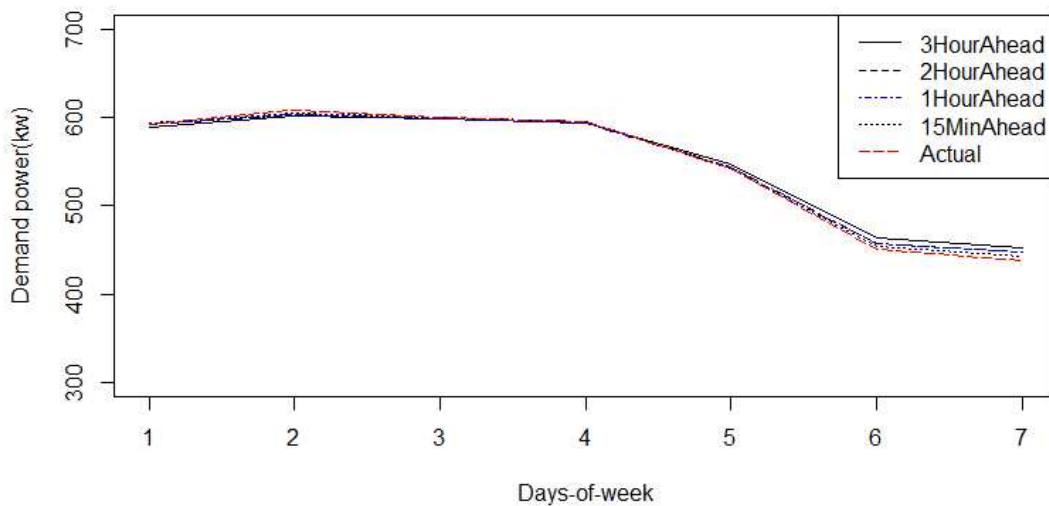


Figure 33. 요일별 평균 전력 수요 예측

Table 10. 평균 전력 수요의 실제값과 예측값의 차 (70일치 요일별)

	최소	최대
15분 후	0.02kwh(화)	2.61kwh(월)
1시간 후	0.09kwh(금)	4.24kwh(일)
2시간 후	0.54kwh(수)	9.18kwh(일)
3시간 후	1.03kwh(수)	14.90kwh(일)

지금까지의 표들이 70일치에 대한 실제값과 예측값의 차라면 Figure 34부터 Figure 37까지의 그래프는 70일치에 대한 예측 오차들과 표들을 나타낸다. 예측 오차는 실제값 대비 실제값과 예측값의 차이를 비율로 나타낸 것이다.

Figure 34와 Table 11은 각각 15분 후, 1시간 후, 2시간 후, 3시간 후의 70일치 최대 전력 수요의 예측 오차와 표를 보여준다. 15분 후의 예측 오차는 70일 동안 단지 최소 0.2%에서 최대 5.2%까지 예측력이 5% 범위 내에 존재하며 1시간 후는 최대 17.7%, 2시간 후는 최대 26.7%, 3시간 후는 최대 31.3%이다.

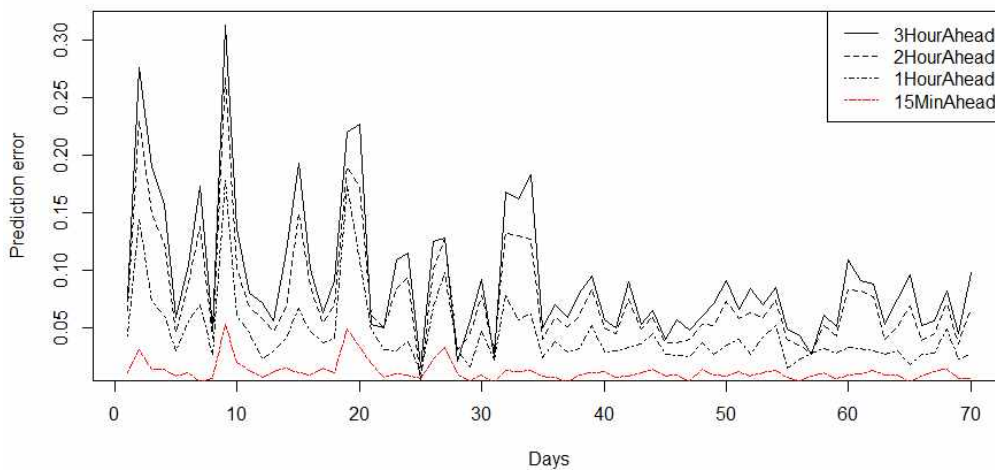


Figure 34. 일별 최대 전력 예측 오차

Table 11. 최대 전력 수요의 예측 오차 (70일치 15분별)

	최소	최대
15분 후	0.0023	0.052
1시간 후	0.0099	0.177
2시간 후	0.0034	0.267
3시간 후	0.0160	0.313

또한 Figure 35와 Table 12는 각각 15분 후, 1시간 후, 2시간 후, 3시간 후의 70일치 평균 전력 수요의 예측 오차와 표를 보여준다. 15일부터 35일까지만을 제외하고는 15분 후, 1시간 후, 2시간 후, 3시간 후의 예측 오차는 거의 0에 가깝다. 가장 높은 예측 오차는 3시간 후의 예측 오차 14.2%이다.

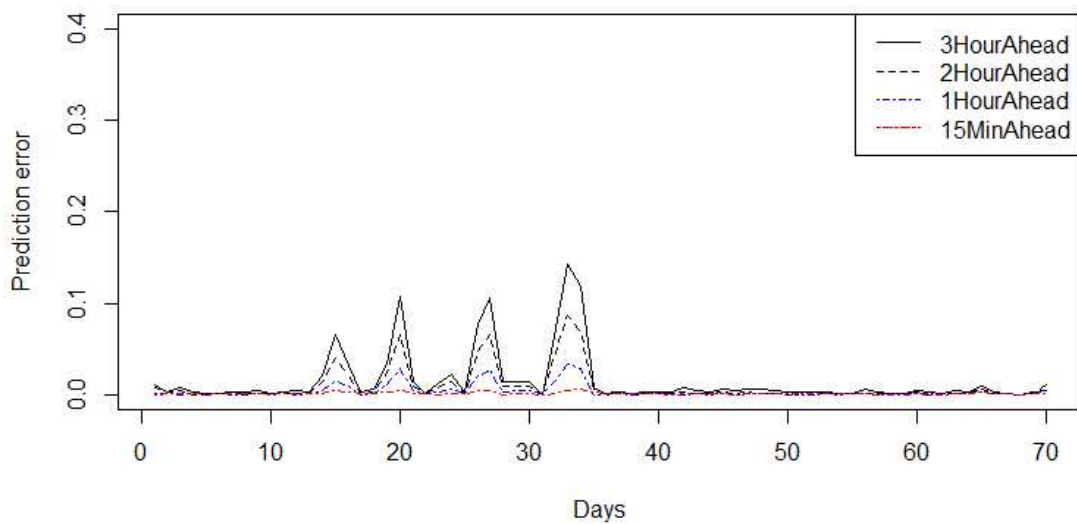


Figure 35. 일별 평균 전력 예측 오차

Table 12. 평균 전력 수요의 예측 오차 (70일치 15분별)

	최소	최대
15분 후	0	0.0056
1시간 후	0	0.0339
2시간 후	0	0.0867
3시간 후	0	0.1427

Figure 36과 Figure 37은 15분 후, 1시간 후, 2시간 후, 3시간 후의 70일치에 대한 각각 요일별 최대 전력 수요의 예측 오차와 요일별 평균 전력 수요의 예측 오차를 보여주며, Table 13과 Table 14는 각각 이에 대한 표를 나타낸다.

요일별 최대 전력 수요의 예측 오차는 15분 후가 가장 낮으며 3시간 후의 예측 오차도 27.6%로 비교적 낮게 나타난다.

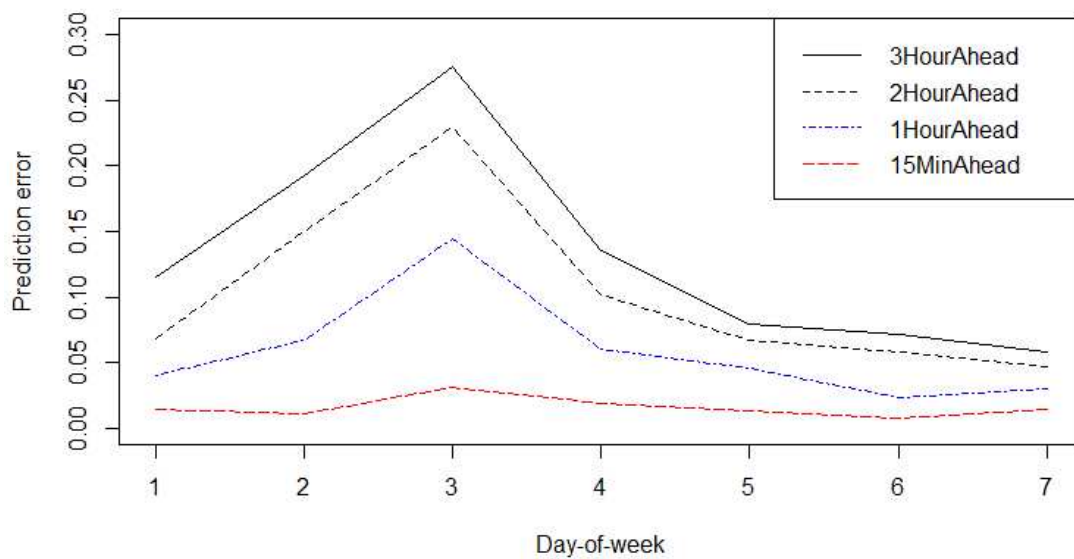


Figure 36. 요일별 최대 전력 예측 오차



Table 13. 최대 전력 수요의 예측 오차 (70일치 요일별)

	최소	최대
15분 후	0.0072(토)	0.031(수)
1시간 후	0.0229(토)	0.145(수)
2시간 후	0.0464(일)	0.230(수)
3시간 후	0.0590(일)	0.276(수)

요일별 평균 전력 수요의 예측 오차는 15분 후가 최대 0.2%로 실제값과 거의 차이가 없으며, 1시간 후가 최대 1.0%, 2시간 후가 최대 2.2%, 3시간 후가 최대 3.4%로 매우 작게 나타난다.

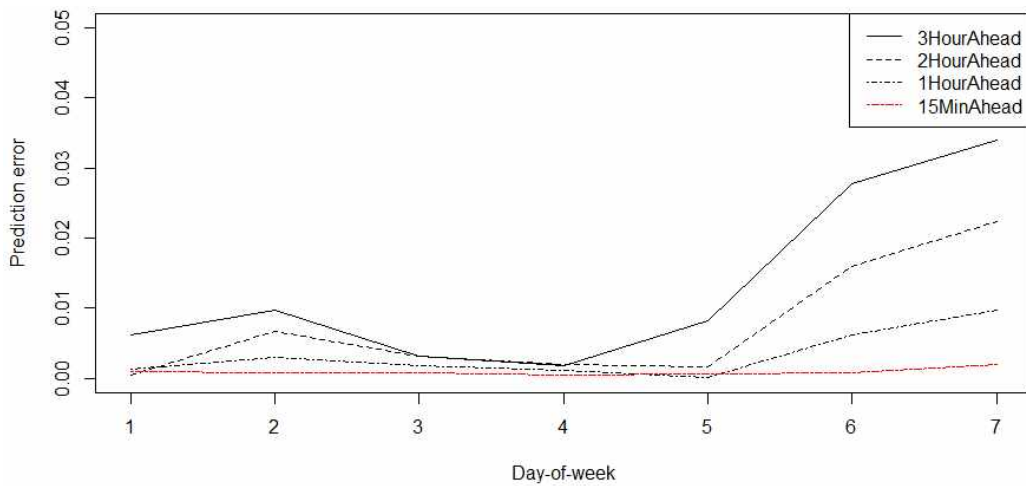


Figure 37. 요일별 평균 전력 예측 오차

Table 14. 평균 전력 수요의 예측 오차 (70일치 요일별)

	최소	최대
15분 후	0.0003(목)	0.002(일)
1시간 후	0.0001(금)	0.010(일)
2시간 후	0.0003(월)	0.022(일)
3시간 후	0.0017(목)	0.034(일)

## IV. 백트래킹 기반의 전력 소비 스케줄링 기법

### 1. 태스크 모델

인 홈 네트워크에서 발생하는 전력 소비 태스크에 대한 정보를 획득할 수 있다. 사용자들은 IHD나 PC등이 제공하는 유저 인터페이스를 통해 인 홈 네트워크에서 발생하는 전력 소비 태스크 정보를 입력할 수 있다. 또한 사용자에게 의한 입력이 아니라 스마트 홈 시스템 내의 스마트 가전 기기로부터 입력될 수도 있다. 전력 소비 태스크 정보는 각종 전력 기기가 언제 작동해야 하는지, 언제까지 작동해야 하는지, 얼마 동안 작동해야 하는지 등이 될 수 있다 [Lee et al. 2011h].

각각의 태스크는 태스크들이 갖는 고유의 전력 소비 특징이 있으며, 태스크는 3가지의 클래스로 분류될 수 있다. Class 1의 태스크들은 그 태스크가 활성화 되면 즉시 시작해야하고 끝날 때까지 중지되어질 수 없다. 예를 들어 헤어 드라이기는 사용자가 사용을 원하는 그 순간에 시작되어진다. 이러한 태스크들은 전력 소비가 고정되고 작동을 조절할 수 있는 다른 옵션을 가지고 있지 않기 때문에 스케줄링이 불가능하다. Class 2의 태스크들은 활성화 되면 즉시 시작해야만 하는 것은 아니며 스케줄링이 가능한 비선점형(nonpreemptive)의 태스크이다. 식기 세척기 또는 세탁기는 특정 종료 시한 안에 작동이 완료되어지기만 하면 언제든지 시작할 수 있다. 이러한 태스크들은 한번 시작되어지면 중지 없이 끝까지 수행되어야 한다. 마지막으로 Class 3의 태스크들은 Class 2의 태스크들이 갖는 비선점형을 제외하고는 비슷하다. Class 2의 태스크들과 같이 그 태스크들의 활성화된 후에 시작할 수 있다. 전기자동차의 충전은 Class 3의 태스크들에 속하며 이러한 태스크들은 종료 시한 내에서 중지되어질 수도 있고 다시 재개할 수도 있다 [Lee et al. 2011g].

Table 15. 스케줄링에 사용될 태스크의 분류

구분	Class 1	Class 2	Class 3
특징	<ul style="list-style-type: none"> <li>· 태스크가 활성화되면 바로 시작해야 함</li> <li>· 끝날 때까지 중단될 수 없음</li> <li>· 사용자가 원하는 그 순간에 시작해야 함</li> </ul>	<ul style="list-style-type: none"> <li>· 활성화되면 즉시 시작해야 하는 것은 아님</li> <li>· 특정 종료시한 안에 끝낼 수 있다면 언제든지 시작 가능</li> <li>· 한번 시작하면 중단없이 계속해서 작동해야 함</li> </ul>	<ul style="list-style-type: none"> <li>· Class2와 같이 시작시간 후에 시작해야 함</li> <li>· 종료시한 안에는 중단되고 재개될 수 있음</li> </ul>
스케줄링	<b>non-schedulable</b>	<b>schedulable (nonpreemptive)</b>	<b>schedulable (preemptive)</b>
예	헤어 드라이기 등	식기 세척기, 세탁기 등	전기자동차 등

본 논문은 스케줄링이 가능한 Class 2와 Class 3에 대한 스케줄링에 초점을 맞추는데, Class 2와 Class 3은 각각 비선점형과 선점형으로 나타내어질 것이다. 다른 복잡한 태스크들은 Class 2와 Class 3의 조합으로 지정되어진다.

전력 소비 프로파일은 각각의 전력 기기의 전력 소비 행동을 특징화하기에 적절하다. Figure 38은 5개의 태스크들의 전력 소비 프로파일 샘플을 나타낸다. 시간 축은 고정크기의 타임 슬롯으로 나누어져 있으며 각각의 전력 기기들의 전력 소비는 고정된 타임 슬롯에 정렬되기 때문에 전력 기기들의 스케줄링을 슬롯 할당 문제가 된다. 타임 슬롯의 길이는 스케줄의 정밀도와 컴퓨팅 시간에 대한 시스템 요구에 따라 조정되어질 수 있다. 그래프에서의 전력 스케줄링 슬롯의 길이는 20 분이며, 각각의 전력 기기들의 전력 소비 패턴은 고정된 크기의 타임 슬롯에 맞춰 조정되어진다. 실제로 각각의 전력 기기들은 전력 수요 행동 내에서 각자 자신만의 시간 크기를 갖는다.

그러나 본 논문은 전력 제어 용량을 고려하는 슬롯의 대표 값으로 피크와 평균 전력을 선택함으로써 모든 전력 기기는 전력 수요 내에서 같은 크기를 갖는다고 가정한다. 물론 더 작은 슬롯은 더 정확한 스케줄을 생성할 수 있으나 그러면 당

연히 복잡성과 정확성 사이의 상충적 관계(trade-off)가 존재하기 때문에 슬롯이 더 작을수록 더 정확한 스케줄을 생성하는 것은 당연하다. 전력 부하 프로파일의 정확성은 올바른 스케줄을 생성하는데 중요하나 본 논문의 스케줄링 기법에서는 충분히 정확한 프로파일은 이용할 만하다고 가정한다. 슬롯 당 총 부하를 나타내고, 슬롯 6에서는 피크가 10이 됨을 보이고 있다.

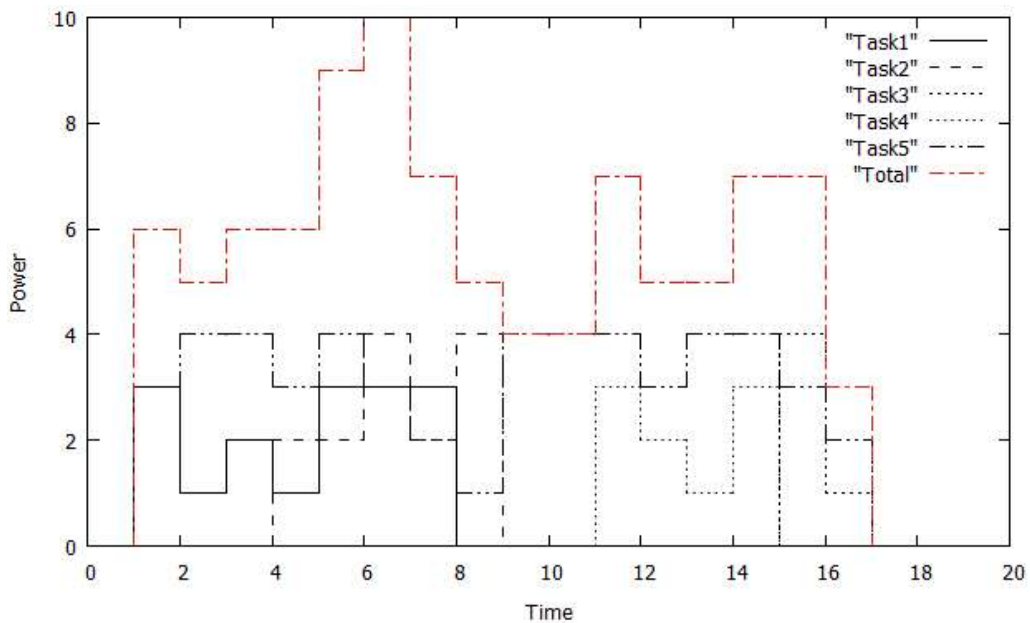


Figure 38. 스케줄링 전의 전력 소비 프로파일

전력 소비 프로파일은 Figure 39와 같이 슬롯 할당 테이블로 나타내어질 수 있다. 슬롯 할당 테이블은  $M \times N$ 으로 구성되며  $M$ 은 슬롯의 수를,  $N$ 은 태스크의 수를 나타낸다. 타임 슬롯안의 숫자는 전력 수요량을 나타낸다. 전력 단위는 가정, 빌딩, 농장, 센서 네트워크와 같은 스마트그리드의 유형에 의해 결정되는 상대적인 값을 지닌다 [Derin et al. 2010]. 따라서 본 논문에서는 전력의 단위를 명시하지 않는다. 단순화를 위해 모든 태스크의 종료 시한은 20으로 하며, 각각의 행에는 태스크 하나의 수행 스케줄이 슬롯 바이 슬롯으로 할당되어진다. 따라서 슬롯 당 각 태스크의 총합은 스케줄이 되어지는 전력 수요량이 되며, 총합 중 가

장 높은 값은 전력 소비 스케줄의 피크 부하가 된다.

Figure 39에서의 현재 슬롯은 20개이고, 태스크의 수는 5개이다. 이 때, 태스크 1부터 4까지는 비선점(N 표시), 태스크 5는 선점형(P 표시)의 태스크이다. 현재 태스크 1의 경우는 비선점형의 태스크로 타임 슬롯 1에서 시작하고 (3, 1, 2, 1, 3, 3, 3)의 연속적인 전력 소비 특성을 나타내고, 태스크 2 또한 비선점형의 태스크로 타임 슬롯 4에서 시작하고 (2, 2, 4, 2, 4)의 연속적인 전력 소비 특성을 나타낸다. 마찬가지로, 나머지 태스크들도 프로파일에 의해 설정된 타임 슬롯에서 동작이 시작되고, 미리 설정된 각각의 전력 소비 특성을 보인다.

슬롯 할당 테이블에서의 전체 전력 소비량은 (6, 5, 6, 6, 9, 10, 7, 5, 4, 4, 7, 5, 5, 7, 7, 3)의 값을 갖는다. 피크로드는 각 슬롯의 전력 소비의 합이 가장 높은 값을 나타내며, 현재 피크로드는 10이다. 전력 스케줄러는 각각의 타임 슬롯에서 전체 전력 소비량을 고려하여 스케줄링 함으로써 전력 소비가 집중되는 피크 시간대를 줄일 수 있고, 이는 곧 피크로드를 줄일 수 있게 된다.

Slot	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
Task 1(N)	0	3	1	2	1	3	3	3	0	0	0	0	0	0	0	0	0	0	0	0
Task 2(N)	0	0	0	0	2	2	4	2	4	0	0	0	0	0	0	0	0	0	0	0
Task 3(N)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	1	0	0	0
Task 4(N)	0	0	0	0	0	0	0	0	0	0	3	2	1	3	0	0	0	0	0	0
Task 5(P)	0	3	4	4	3	4	3	2	1	4	4	4	3	4	4	3	2	0	0	0
Total	0	6	5	6	6	9	10	7	5	4	4	7	5	5	7	7	3	0	0	0

Figure 39. 전력 소비 프로파일에 대한 슬롯 할당 테이블

Figure 40은 적절한 스케줄링 후의 슬롯 할당 테이블을 나타낸다. 태스크 2는 타임 슬롯 4에서 19사이에 동작될 수 있으면서 비선점형 태스크의 동작 정보를 갖고 있다면 태스크 2의 동작 시작 시간을 타임 슬롯 4에서 19사이에 언제든지 시작할 수 있으므로 동작 시간을 조정하고 연속적인 동작을 하도록 스케줄링할 수 있다. 따라서 태스크 2는 타임 슬롯 5에서 동작이 시작되어 (2, 2, 4, 2, 4)의 전

력 소비 특성을 연속적으로 보이며 동작할 것이다.

또한 태스크 5는 타임 슬롯 1에서 19사이에 동작될 수 있으면서 선점형 태스크의 동작 정보를 갖는다면 태스크 5는 타임 슬롯 1에서 19사이에 적어도 한번 이상은 중지, 재개할 수 있기 때문에 스케줄러는 피크 시간을 줄이기 위해 태스크 5의 동작을 타임 슬롯 5에서 일시적으로 중지시키고 타임 슬롯 8에서 다시 시작하도록 스케줄링할 수 있다. 이러한 스케줄링을 통해 피크로드는 10에서 7로 감소되어진다. 즉, 전력 기기들의 동작 시간과 연속성을 조절함으로써 스케줄링함으로써 타임 슬롯별 전체 전력 소비량이 달라지고, 이에 따라 몇몇의 피크 시간도 제거될 수 있다.

Slot	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
Task 1(N)	0	3	1	2	1	3	3	3	0	0	0	0	0	0	0	0	0	0	0	0
Task 2(N)	0	0	0	0	0	2	2	4	2	4	0	0	0	0	0	0	0	0	0	0
Task 3(N)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	1	0	0	0
Task 4(N)	0	0	0	0	0	0	0	0	0	0	0	3	2	1	3	0	0	0	0	0
Task 5(P)	0	3	4	4	3	0	0	0	4	3	2	1	4	4	4	3	4	4	3	2
Total	0	6	5	6	4	5	5	7	6	7	2	4	6	5	7	7	5	4	3	2

Figure 40. 스케줄링 후의 슬롯 할당 테이블

태스크  $T_i$ 는  $\langle F_i, A_i, D_i, U_i \rangle$ 의 튜플로 모델링되어질 수 있다.  $F_i$ 는  $T_i$ 의 종류가 선점형인지 비선점형인지를 나타내고,  $A_i$ 는  $T_i$ 의 활성화 시간(activation time),  $D_i$ 는 종료 시한(deadline),  $U_i$ 는 전력 소비 시간에 따른  $T_i$ 의 수행 길이(operation length)를 나타낸다. 활성화 시간은 각각의 태스크들이 수행을 시작할 수 있는 가장 이른 시간을 나타내고, 종료 시한은 각각의 태스크들이 수행을 완료해야 하는 가장 늦은 시간을 말하며, 수행 길이는 각각의 태스크들이 수행을 완료하는데 최소의 시간을 말한다 [Lee et al. 2012b].

비선점형 태스크는 활성화 시간부터 가장 늦은 시작 시간 사이에 즉,  $A_i$ 에서부터  $(D_i - U_i)$  사이의 기간 동안에 시작할 수 있다. 비선점형 태스크는 언제든지 시작

이 가능하지만 예를 들어  $U_i$ 가 3이고  $D_i$ 가 10이면 최소한 7에는 시작을 해야만 3만큼을 수행하고 10까지 끝낼 수가 있다. 따라서 최소  $A_i$ 에서 시작하거나 최대  $(D_i - U_i)$ 에서 시작해야 한다. 비선점형 태스크의 경우는 태스크가 한번 시작되어지면 선점되어질 수 없기 때문에 시작 시간이 선택되어지면 프로파일 엔트리는 단지 할당 테이블로 하나씩 차례차례 복사되어진다. 또한 최대  $M$ 개의 선택 옵션을 갖게 되기 때문에 비선점형 태스크의 서치 스페이스 순회의 시간 복잡도는  $O(M)$ 이 된다 [Lee et al. 2011f].

반대로 선점형 태스크의 경우는 비연속적인 작업이 가능하기 때문에 다소 복잡할 수 있다. 이러한 시간 제약 조건을 만족하기 위해서  $(D_i - A_i)$  슬롯으로 부터  $U_i$ 만큼 타임 슬롯이 할당되어야 한다. 이 때의 서치 스페이스의 트리의 단말 수는  $(D_i - A_i)C_{U_i}$ 와 같고, 그 결과 시간 복잡도는  $O(M^{\frac{M}{2}})$ 가 된다. 또한 선점형 태스크가 갖게 되는 최악의 서치 스페이스의 크기는  ${}_M C_{M/2}$ 이고, 시간 복잡도는 다음의 수식과 같이 계산되어질 수 있다.

$$\begin{aligned}
 O({}_M C_{M/2}) &= O\left(\frac{M(M-1)\dots(\frac{M}{2}+1)(\frac{M}{2})(\frac{M}{2}-1)\dots 1}{(\frac{M}{2})(\frac{M}{2}-1)\dots 1(\frac{M}{2})(\frac{M}{2}-1)\dots 1}\right) \\
 &= O\left(\frac{(\frac{3}{4}M)(\frac{3}{4}M)\dots(\frac{3}{4}M)}{(\frac{1}{4}M)(\frac{1}{4}M)\dots(\frac{1}{4}M)}\right) \\
 &\cong O(3^{M/2})
 \end{aligned} \tag{7}$$

즉, 비선점형 태스크는 선형으로  $M$ 에 의존되어지는 반면, 선점형 태스크의 스페이스 바운드는  $M$ 으로 증가할 수 있는 조합의 수에 의존된다. 이런 이유로 스케줄링 시간은 근본적으로 선점형 태스크의 탐색 크기에 민감하다 [Lee et al. 2012g].



## 2. 제안 기법

본 논문에서 제안된 전력 소비 스케줄링 기법은 전력 소비 동작을 실시간 태스크 모델로 변환하여 컴퓨터 알고리즘 즉, 완전 탐색(exhaustive search)에 대한 서치 스페이스를 확장해 가며 피크를 줄일 수 있는 스케줄을 찾는 것이고, 스케줄링의 속도를 향상시키기 위해 제약 조건 설정에 의해 불필요한 확장을 줄인다. 또한 선점형 태스크의 경우는 가능한 모든 조합을 미리 구해 저장해놓고 하나씩 확장해가면서 스케줄링함으로써 수행 시간을 대폭 감소시킬 수 있다 [Lee et al. 2011a].

백트래킹 알고리즘은 점진적으로 탐색 트리를 생성한다. 가능성 있는 탐색 트리는 유효한 솔루션들을 포함하고 있는 모든 실현 가능한 솔루션들로 구성된다. 그리고 반복적으로 루트에서 단말로 하향하며 순회하는 깊이 우선 탐색을 하고, 부분적인 솔루션에 해당하는 각각의 중간 노드에서 그 노드가 유효한 솔루션이 되었는지를 체크하고, 그렇지 않으면 남은 서브 트리는 제거된다.

백트래킹 기반의 전력 소비 스케줄러는 먼저 입력된 프로파일에 대한 모든 가능한 슬롯 할당 테이블 즉, 태스크  $T_1 \sim T_n$  중 적어도 하나의 시작 시간, 중지 시간, 재개 시간들을 조절하면서 모든 가능한 스케줄링을 생성한다. 그리고 나서 각각의 스케줄링에 대한 최적의 피크로드를 검색하고, 검색된 최적 피크로드를 갖는 스케줄의 타임 슬롯으로 태스크를 할당한다.

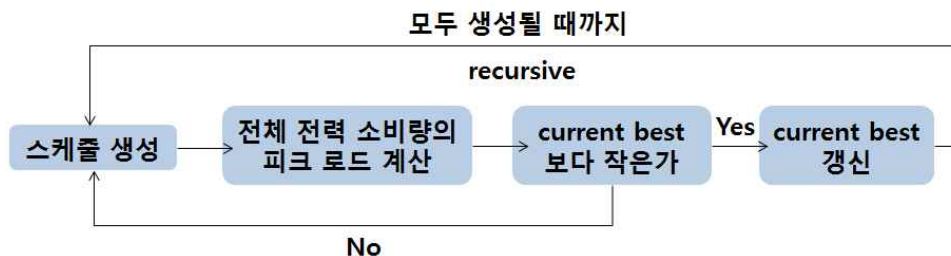


Figure 41. 백트래킹 기반의 전력 소비 스케줄링

피크로드는 생성된 슬롯 할당 테이블의 전체 전력 소비량을 계산하여 구할 수 있으며, 현재 최적 피크로드(이하 current best)는 처음으로 생성된 스케줄로 계산된 피크로드로 설정될 수 있다. 검색 트리를 확장해 가며 계산된 피크로드와 current best를 비교해 가며 계산된 피크로드가 current best보다 작으면 계산된 피크로드를 current best로 업데이트하고, 계산된 피크로드가 current best보다 작지 않으면 다른 스케줄을 생성하기 위해 현재의 슬롯 테이블 할당을 종료하고, 다음 할당을 실행한다. 이 과정은 스케줄이 모두 생성될 때까지 반복한다.

스케줄러는 모든 가능한 스케줄 중에서 어떤 스케줄이 최적인지 모든 가능한 스케줄링에 대한 검색 트리를 확장하게 되는데, 확장된 검색 트리 중에서 최적 피크로드보다 더 좋은 피크로드를 생성할 가능성이 없는 트리가 있으면 최적 피크로드 검색 대상에서 제외시킴으로써 검색 속도를 향상 시킬 수 있다. 즉, 서치 스페이스 확장 시 태스크를 위한 부분적이 할당의 피크로드 값이 current best를 초과할 경우 스케줄링을 종료한다.

스케줄링을 위해 설정한 가정은 다음과 같다.

- 가정/빌딩 내에는 여러 지능형 전기 장치가 있으며, 이들은 특정 종료 시한 내에 작업이 완료 되어야 한다.
- 이들의 동작은 제어기에 의해 작동/중지/재개될 수 있다.
- 전력 기기별로 전력 소비량은 사전에 알려져 있다.
- 전력 소비 패턴은 장치별로 선점형과 비선점형으로 나뉘며, 선점형은 중단 후 재개될 수 있고, 비선점형은 한 번 시작하면 중단될 수 없다.
- 시간들은 고정된 크기의 슬롯으로 구분되어진다
- 전력 소비량은 슬롯 단위로 정해져 있다.

Figure 42는 스케줄링을 위한 검색 트리를 나타낸다. 검색 트리는 모든 가능한 할당 테이블을 생성할 수 있다. 그리고 서치 스페이스를 확장하며 리프(leaf)에 도달했을 때 하나의 스케줄이 완성이 되고, 이 때의 피크로드를 계산한다. 계산되어진 이 스케줄의 피크로드와 current best를 비교해가며 current best를 업데이트한다. current best는 처음으로 생성된 스케줄로 계산된 피크로드로 설정될

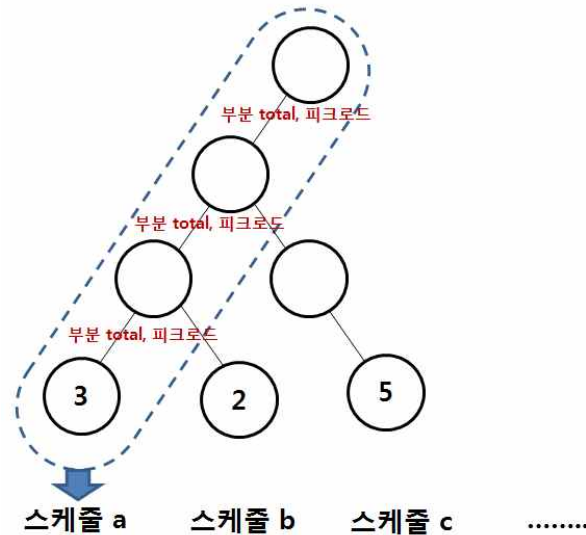


Figure 42. 스케줄링을 위한 Search Tree

백트래킹 알고리즘은 가능성 있는 탐색 트리는 유효한 솔루션들을 포함하고 있는 모든 실현 가능한 솔루션들로 구성되는데 그 솔루션들의 각각은  $M \times N$  할당 테이블로 표현된 전력 소비 스케줄이다.  $M$ 은 슬롯의 개수,  $N$ 은 태스크의 개수이다. 할당 프로시저는 첫 번째 행으로부터 할당 테이블을 채우게 되는데 이 때, 각각의 행은 태스크와 연관된다.

Figure 43은 상세한 스케줄링 알고리즘을 나타낸 것이다. 기본적으로 프로시저는 모든 가능성 있는 서치 스페이스를 생성한다. 또한 할당 테이블 하나당 태스크의 개수만큼 탐색을 해야 하고, 이렇게 생성되어지는 할당 테이블의 개수가 많아지면 탐색 시간이 오래 걸린다. 탐색 시간은 선점형이 1개, 비선점형이 4개라면  $O(M \times M \times M \times M \times {}_M C_{M/2})$ 인데, 태스크의 수가 많아진다면 이러한 테이블이 더 많이 생성되어지기 때문에 탐색 시간이 더 오래 걸린다. 따라서 백트래킹을 통해 수많은 할당 테이블 중 피크치가 작은 것을 선택한다. 이 때의 피크로드는 각각의 할당 테이블의 총합 중 가장 큰 값을 나타낸다. 즉, 할당 프로시저가 단말 노드에 이르렀을 때 *EvalAlloc* 함수는 현재의 할당 테이블의 피크로드와 current best와 비교하여 IF문이 참이면 즉, 현재의 할당 테이블의 피크치가 더 작다면

current best는 현재의 할당 값으로 대체하고, 단말 노드가 아니라면 그 할당은 계속 진행된다.

```

procedure AllocTab (i)
input : { $T_i \mid F_i, A_i, D_i, U_i$ }
  if i equals to N
    EvalAlloc( )
  end if
  if  $T_i$  is nonpreemptive
    for each start time from  $A_i$  to  $D_i - U_i$ 
      copy the profile to i th row of the allocation table
      if (! CheckConstraint( )) AllocTab (i + 1)
    else
      for each combination of  $(D_i - A_i)CU_i$ 
        map the profile to i th row of the allocation table
        if (! CheckConstraint( )) AllocTab (i + 1)
      end if
    end if
  end procedure

```

Figure 43. 전력 소비 스케줄링 알고리즘

$T_i$ 가 비선점형일 때,  $A_i$ 에서  $D_i - U_i$  까지의 모든 가능한 시작 시간에 대해 프로파일을 복사하며 ( $i + 1$ )번 째 행의 할당이 하나씩 차례차례 반복적으로 시도된다. 만일 전력 소비 프로파일의 비선점형 태스크가 (3, 4, 5, 2)이고, 시작시간이 2 라면 2부터 순차적으로 (0, 0, 3, 4, 5, 2, ...)가 할당될 것이다.

반대로  $T_i$ 가 선점형이라면 그 프로파일이 바로 복사되는 게 아니라 조합 벡터들을 이용하면서 할당 테이블로 사상된다. 예를 들면, 전력 프로파일의 태스크 셋이 (3, 4, 5, 2)일 때, 활성화 시간 2이고, 조합 벡터가 (1, 1, 0, 0, 1, 1)라면, 할당은 (0, 0, 3, 4, 0, 0, 5, 2)가 될 것이다. 따라서  $T_i$ 는 시간 4에서 중지되어질 것이고 시간 6에서 재개되어질 것이다.

*CheckConstraint* 함수는 불필요한 탐색 트리의 확장을 제거함으로써 탐색 프로시저의 속도를 높일 수 있다.  $T_0$ 에서  $T_1$ 까지의 할당 테이블의 일부분의 최대 전력 수요량이 이미 current best를 초과하였다면 남아 있는 단계는 수행되지 않는

다 [Lee et al. 2012d]. 결국 Figure 44는 주어진 태스크 셋을 백트래킹 기반의 스케줄러에 의해 생성된 전력 스케줄을 보여주고 있다. 선점형 태스크인  $T_5$ 는  ${}_{19}C_{16}$ 의 경우의 수가 있고 할당 결과에 따르면 time 1에서 시작되어지고, 5에서 중지되었다가 8에서 재개될 것이다. 이로써 최대 전력 요구량은 스케줄링 전 10에서 7이 된다.

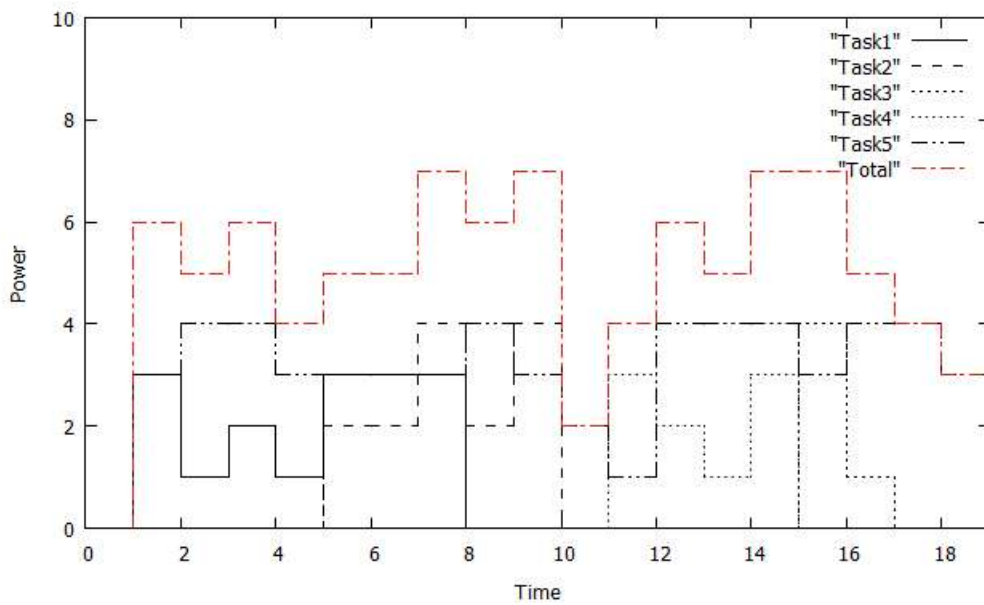


Figure 44. 스케줄링 후의 전력 소비 프로파일

태스크에 대한 모든 가능한 스케줄을 생성하면 최적성은 보장될 수 있지만 가정 뿐만 아니라 빌딩 등의 현실세계에서는  $M$ 과  $N$ 의 개수가 커지게 되고, 선점형과 비선점형이 섞인 경우 선점형 태스크로 인해 서치 속도는 폭증하게 된다. 이를 해결하기 위해 선점형 태스크의 경우는 먼저 가능한 조합 벡터를 미리 생성하고, 피크로드를 검사해가며 할당한다. 예를 들어, 선점형 태스크의 프로파일 개체를 할당 테이블의 행으로 사상시키하고자 할 때,  $(D_i - A_i)$  값이 4이고,  $U_i$  값이 2라면 가능한 경우의 수는  ${}_4C_2$  개가 되고, 가능한 조합 벡터들은 각각  $(0, 0, 1, 1)$ ,  $(0, 1, 1, 0)$ ,  $(0, 1, 0, 1)$ ,  $(1, 0, 0, 1)$ ,  $(1, 0, 1, 0)$ ,  $(1, 1, 0, 0)$ 이 된다.

### 3. 실험 및 결과

성능 평가를 위해 Visual C++ 6.0을 이용하여 제안한 할당 기법의 프로토타입을 구현하였으며, 이것은 Intel Core 2의 CPU와 3.0 GB의 메모리, Window Vista 운영 체제를 갖춘 개인용 컴퓨터에서 동작시켰다.

실험에서의 스케줄 길이는  $M$ 이라고 하며, 20개의 시간 단위로 설정한다. 만일 하나의 시간 단위가 20분이라면 스케줄 길이의 합은 400분, 약 6.7시간이 되고, 이는 가정의 전력 기기를 스케줄링 하는데 충분히 큰 시간이라 할 수 있다. 이 때 하나의 태스크의 시작 시간은 0과  $M$ 사이에서 랜덤하게 선택되어진다. 또한 수행되는 길이도 랜덤하게 분포되며 시작 시간과 수행 길이의 합은 종료 시간  $M$ 을 초과할 수 없다. 이 때, 모든 태스크들은 보통 종료 시간  $M$ 을 갖는데, 이것은 명령을 내린 작업이 소비자가 집에 돌아오기 전까지 수행을 끝내야 하는 시간을 의미한다. 각각의 타임 슬롯의 전력 레벨은 1부터 5까지의 값을 가지며, 이 때  $w$ 와  $kw$ 와 같은 전력의 단위는 명시하지 않았다. 각 전력기기의 소비량에 맞추어 보정할 수 있다. 다음의 실험에서 각각의 태스크는 동일한 평균 전력 요구량을 갖기 때문에 태스크의 수는 전력 요구량의 합과 거의 비례한다.

Figure 45는 태스크를 제어하거나 스케줄 하지 않았을 경우 time 6에서 최대 전력 요구량이 10까지 도달할 수 있다는 것을 나타낸다. *Earliest* 기법은 스케줄링이 되지 않은 상태 즉, non-coordination으로 단지 전력 소비 프로파일에 대한 슬롯할당 스케줄링이라고 보며, 본 논문에서의 스케줄링 기법과 비교하기 위해 *Earliest* 라 명명한다. 전기 자동차의 충전과 같은 특별한 경우를 제외하고는 실제 가정에서의 대부분의 전력 기기의 동작들은 비선점형의 형태를 갖는다. 따라서 실험에서 각각의 태스크 셋은 태스크의 수 범위를 3부터 8까지 가지며, 각각의 태스크 셋에서 단지 하나만을 선점형의 태스크로 설정한다.

*Earliest* 기법과 제안 기법의 비교 및 Non-constrained와 Constrained의 비교에 대해 첫 번째 실험은 피크부하 감소와 수행 시간에 따라, 두 번째 실험은 전력 피크 부하에 따라 비교한다.

1) 피크 부하 감소와 수행 시간에 대한 실험

첫 번째 실험은 *Earliest* 기법과 제안된 기법의 태스크 수에 따른 피크 부하를 측정한다. 각각의 태스크 셋들의 피크 전력 값을 측정하고 평균을 구한다. 이 후 전력의 최대치에 대해 *Earliest* 스케줄링 기법과 비교해 보면 Figure 45와 같이 제안된 기법이 *Earliest* 스케줄링보다 23.1 % 까지 피크 부하를 감소시킬 수 있음을 나타낸다. 또한 태스크의 수가 8이 되면 성능의 차이는 더 크게 증가하며, 결국 개별적인 전력 요구량에 따른 성능의 향상은 더 커질 것으로 예상된다.

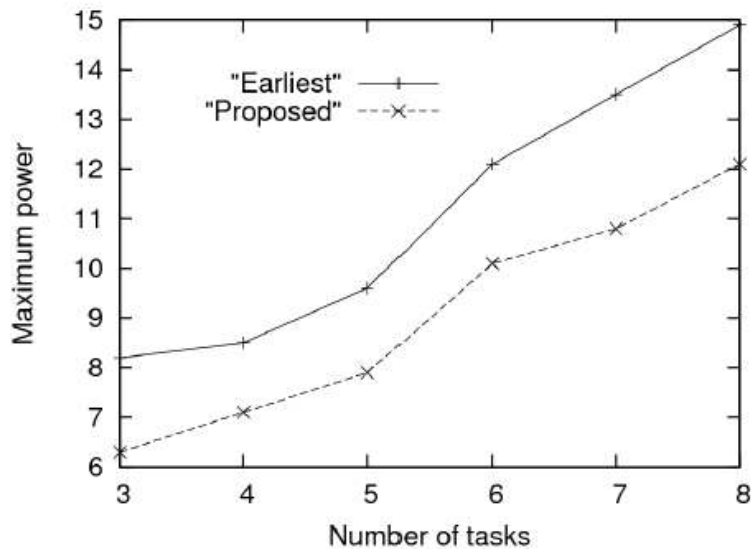


Figure 45. 태스크 수에 따른 최대 전력 요구량 비교

두 번째 실험은 제안된 스케줄링 기법의 서치 스페이스 가지치기 효과에 따른 수행 시간을 측정하여 각각의 기법을 비교한다. 이 때 수행 시간 측정은 *ms* 단위로 경과한 시간을 체크해주는 Microsoft Windows API의 *GetTickCount* 시간 함수를 사용한다.

Figure 46과 Figure 47은 각각 1개의 선점형 태스크와 2개의 선점형 태스크일 때의 측정 결과를 나타낸다. 각 그래프에서 보이는 두 개의 선은 서치 스페이스

의 가지치기 여부에 따라 실선은 Non-Constrained, 점선은 Constrained를 가리킨다. Non-Constrained의 경우 Figure 46에서의 수행 시간은 최고 140ms이고, Figure 47에서의 수행 시간은 최고 30,000ms이다. 즉, 선점형 태스크가 2개 이상이 되면 Non-Constrained의 경우는 수행 시간이 길어지고 있으나 반면 Constrained 서치의 경우는 Non-Constrained의 경우와 비교해 볼 때 상당히 짧은 수행 시간과 안정적인 분포를 보인다. 그런 점에서 단지 두 경우의 비교를 통해서도 가지치기의 효과는 충분히 드러난다. 실제로 수행 시간은 선점형 태스크의 서치 스페이스의 사이즈에 많이 좌우된다. 비록 다수의 태스크 셋의 평균화를 통해 이러한 효과를 어느 정도 줄일 수는 있지만 사이즈가 큰 서치 스페이스를 갖는 태스크 셋은 전체적인 수행 시간이 커질 수밖에 없다.

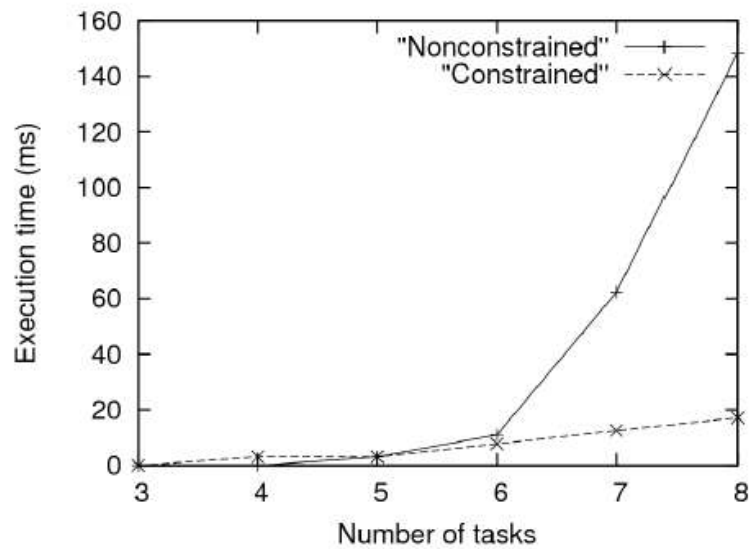


Figure 46 태스크 수에 따른 수행 시간 비교 : 1개의 선점형 태스크

Constrained 서치는 가지치기를 통해 불필요한 서치 스페이스 순회를 제거하고 수행 시간을 향상시킬 수 있다. 이는 스케줄링 기법의 현장 응용에 효과적인 휴리스틱의 분명한 중요성을 말해준다. 더 나아가 만일 선점형과 비선점형의 태스크 수로  $N$ 을 각각  $N_p$ 와  $N_{np}$ 로 나눈다면 서치 스페이스의 최악의 시간 복잡도는



$O(M^{N_p} | (3^{\frac{M}{2}})^{N_p})$ 에 근사될 수 있다.

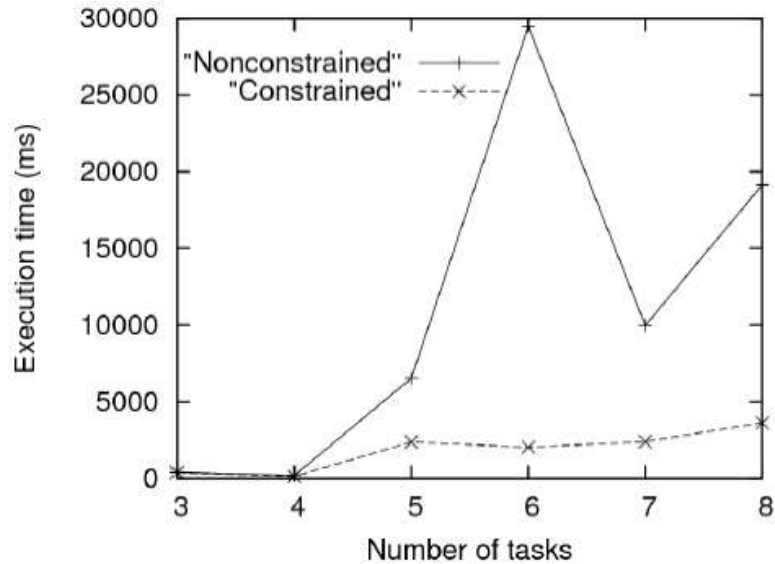


Figure 47. 태스크 수에 따른 수행 시간 비교 : 2개의 선점형 태스크

일반적인 컴퓨터 시스템에서의 태스크 스케줄러와 달리 본 논문의 전력 스케줄러는 몇 초의 수행 시간을 제공하는 것을 목표로 삼고 있다. 소비자가 태스크 프로파일을 제공하고 스케줄러를 시작한다면 가정 내의 전력 컨트롤러는 스케줄을 자동으로 생성할 것이고 이에 대한 작동을 관리한다. 따라서 소비자는 스케줄이 완료되는 것을 기다릴 필요가 없다.

## 2) 전역(global) 피크의 효과에 대한 실험

다수의 로컬 스케줄의 전체적인 조정이 유용하지 않은 경우에는 각각의 로컬 피크를 줄이는 것이 중요한 부분이다. 바람직하지 않게 각 가정에서 같은 타임 슬롯에서 피크를 만들게 되어 각 가정의 스케줄러가 같은 최적화 목표를 가진다면 많은 가정의 태스크는 같은 타임 슬롯에서 스케줄되어질 수 있다. 이렇게 몇몇의

로컬 스케줄이 같은 타임 슬롯에서 피크치를 가질 경우 전역 피크는 아주 커질 수가 있다. 따라서 이번 실험에서는 전역 피크 감소에 대한 제안된 스케줄의 효과를 측정한다. 전역의 전력 소비를 계산하기 위해 로컬 스케줄들의 그룹을 합하는 것이 필요하다. 본 논문에서는 단순화를 위해 각 로컬 스케줄의 슬롯 바운드는 조정되어진다고 가정하며, 슬롯 당 전력 소비량을 생성하기 위해 각 로컬 스케줄마다의 각각의 슬롯에 대한 전력 요구량은 더해진다. 이 때 각각의 로컬 스케줄들이 같은 슬롯에서 피크를 갖게 된다면 전력 피크는 매우 커질 수 있다.

첫 번째 실험은 빌딩과 같은 일반적인 전역 스케줄 그룹 안에 속하는 로컬 스케줄링 개수에 따른 평균 피크치를 측정한다. 먼저 20개의 태스크 셋을 생성하고 전역 피크 부하를 측정한다. 각각의 태스크 셋은 특정 관측 구간 동안의 전력 기기의 작동의 변화를 설명한다.

Figure 48은 로컬 스케줄링 단위에 따른 평균 피크의 변화를 나타낸다. 각각의 스케줄링 단위는 1개의 선점형 태스크를 포함한 5개의 태스크를 가지고 있다. 스케줄링의 단위의 수가 4 이하인 경우는 제안된 기법의 평균 피크 값이 *Earliest* 방식보다 조금 작게 나타난다. 그러나 4이후부터는 *Earliest* 할당 방식이 더 작은 평균 피크 값이 보여준다. 이는 스케줄이 종료 시한 근처에서 태스크의 작동을 더 많이 할당하기 때문이다. 반면에 *Earliest* 스케줄링 기법은 태스크의 작동을 스케줄 구간에 랜덤하게 분배하며 시작하게 된다. 따라서 비록 로컬 스케줄의 부분 조정으로 로컬 스케줄마다의 피크값이 더 작더라도 같은 슬롯 상에 집중하게 되면 *Earliest* 방식보다 전역 피크가 더 높게 나타난다. 그러나 제안된 기법은 스케줄링 그룹 내에서 다양한 전력 요구 패턴 변화에 대한 간헐적인 전력 부하의 급격한 폭증을 방지할 수 있다.

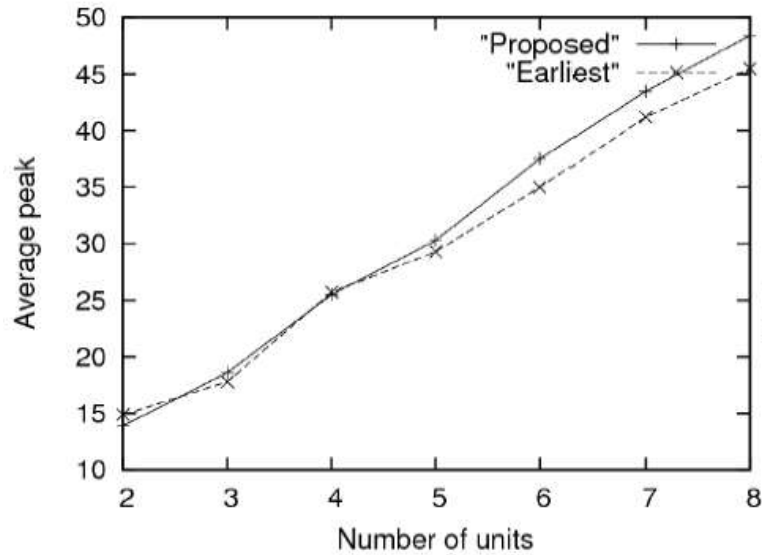


Figure 48. 로컬 스케줄링 수에 따른 평균 피크 비교

Figure 49는 로컬 스케줄링 수에 따른 전역 최대 피크 부하의 비교를 보여준다. 최대 피크는 실제 전력 공급 시스템과 전력 배터리의 용량을 결정하기 요소이기 때문에 성능 평가의 더욱 중요한 기준이다. 그래프는 제안된 스케줄링 기법이 전체적인 전역 조정을 하지 않고 각 로컬의 조정만으로도 *Earliest* 기법보다 피크 부하를 줄일 수 있음을 보여준다. 특히 로컬 스케줄링의 수가 6인 경우 피크 값을 16%까지 감소시킨다. 또한 로컬 스케줄링의 수가 8이상부터는 두 개의 기법의 피크 값은 거의 같아진다.

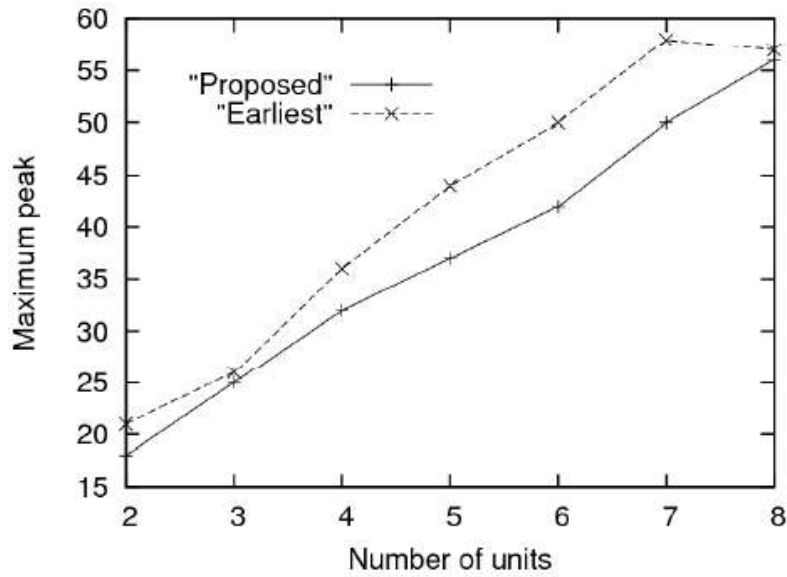


Figure 49. 로컬 스케줄링 수에 따른 최대 피크 비교

Figure 50은 5개의 로컬 스케줄링 그룹에서의 태스크 수에 따른 최대 전력 피크의 비교를 보여준다. 이때, 각 그룹은 단지 1개의 선점형 태스크를 갖는다. 앞에서 언급한 바와 같이 현실 세계에서 선점형 태스크의 예는 상대적으로 적은편이다.

제안된 스케줄링 기법은 *Earliest* 기법보다 안정적인 그래프의 모양을 나타낸다. 또한 태스크의 수가 4일 때, 피크를 20.4%까지 줄인다. 이런 예측 가능한 패턴은 전력시스템의 용량을 계획하는데 도움이 될 수 있다. 태스크의 수가 8 이상인 경우는 두 기법 모두 피크 값이 비슷해지며, 로컬 스케줄링 사이의 전역 조정의 중요성이 나타난다. 적어도 하나의 로컬 스케줄링 단위에서 단 하나의 할당이 최적의 피크를 가질 수 없다. 따라서 각각의 스케줄링 단위는 할당에 대해 피크를 가질 수 있는 다수의 선택권이 존재한다. 각각의 스케줄링 단위의 다수의 최적 할당 없이 하나를 적당하게 선택하는 것이 전역 피크를 줄일 수 있다. 그러나 가정 규모의 태스크의 수와 스케줄링 그룹의 합리적인 수에 대해서 제안된 스케줄링 기법은 로컬 단위와 전역 단위 모두 효과적으로 동작할 수 있다.

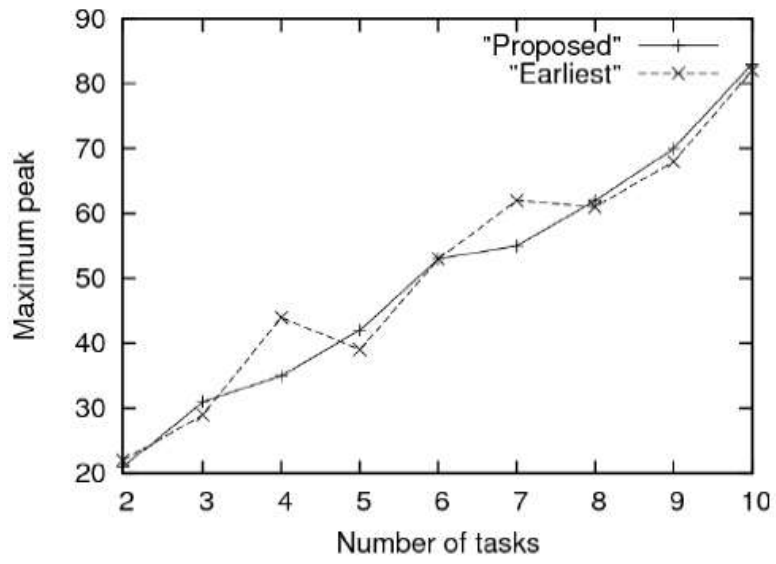


Figure 50. 태스크 수에 따른 최대 전력 피크의 비교 (5 units)

## V. 유전 알고리즘 기반의 멀티쓰레드 전력 소비 스케줄링 기법

### 1. 제안 기법

스마트그리드에서의 피크 부하를 줄이면서 백트래킹 기반의 전력 소비 스케줄링 기법보다 수행 시간을 현저히 개선시킬 수 있는 유전 알고리즘 기반의 멀티 쓰레드 전력 소비 스케줄링 기법을 제안한다. 유전 알고리즘에서 초기 해 집단의 다양성은 곧 해결하고자 하는 솔루션의 다양성을 만들 수 있기 때문에 초기 해 집단을 어떻게 선택하느냐는 중요한 부분이 된다. 그러나 초기 해 집단이 다양하면 수행 시간이 커질 수 있어 다양성과 속도 사이에 상충적 관계가 존재하게 된다. 따라서 일단 초기 해 집단을 다양하게 한 후 각각의 쓰레드에 그 해 집단을 할당하게 하여 각각의 쓰레드가 유전 알고리즘 프로시저를 독립적으로 작동하게 만드는 것이 좀 더 유리할 수 있다. 멀티 쓰레드 전력 소비 스케줄러는 수행 시간을 상당히 향상시킬 수 있기 때문에 멀티 프로세서 구조의 병렬 처리에 유리할 수 있을 뿐만 아니라 소형기기의 컴퓨터 디바이스의 프로그램으로 유용할 수가 있다 [Lee et al. 2011e].

해를 나타내는 벡터는 각각의 태스크들의 할당 조합의 수의 곱만큼 생성될 수 있고, 이것은 유전 알고리즘에 있어서 하나의 해 집단이 될 수 있다. 유전 알고리즘에서의 모든 해는 염색체로 표현되어야 하며 염색체는 정수의 엘리먼트를 포함하는 벡터로 구성될 수 있다. 이를 위해 우선적으로 각 태스크 ( $T_1, T_2, \dots, T_n$ )의 전력 명세 모델이 필요하다. 이 모델에는 각각 선점형인지 비선점형인지를 나타내는  $F_i$  정보와 언제 태스크가 동작하는지를 나타내는 활성화 시간  $A_i$  정보, 언제 동작을 완료해야 하는지를 나타내는 종료 시한  $D_i$  정보, 태스크가 얼마만큼 작동해야 하는지를 나타내는 수행 길이  $U_i$  정보에 따라 슬롯을 할당 받게 된다. 할당에 대한 결과는 Figure 51과 같이 유전 알고리즘에서 정수 엘리먼트를 포함한 벡터로 표현된 해 염색체가 된다.

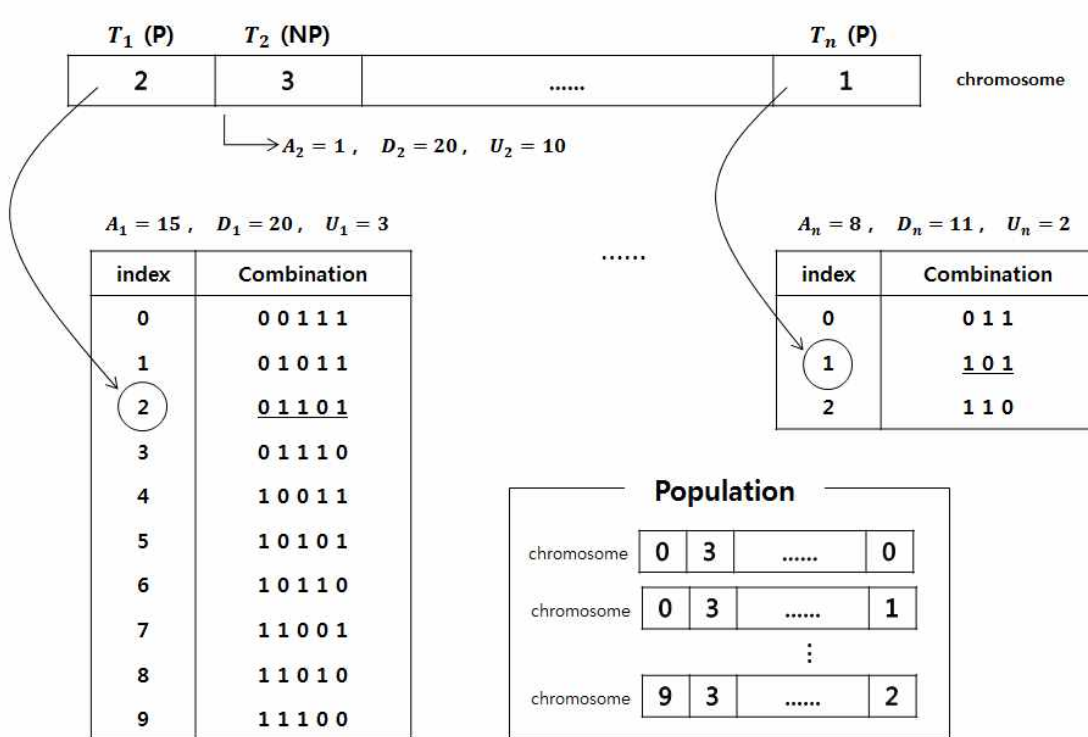


Figure 51. 전력 소비 프로파일에 대한 유전 알고리즘 탐색체

$T_1$ 과  $T_n$ 은 선점형 태스크이고,  $T_2$ 는 비선점형 태스크이다. 이처럼 하나의 탐색체에는 선점형과 비선점형 태스크들이 공존할 수 있다. 그러나 이들에 대응하는 정수 엘리먼트가 표면상 같은 값을 가지더라도 실제 의미하는 바는 다를 수 있다.

선점형 태스크의 조합들에는 각각 고유의 인덱스가 부여된다. 이 때, 벡터의 정수 엘리먼트는 선점형 태스크 조합의 인덱스를 나타내고, 다양한 할당 조합 결과들 중 어느 하나를 나타낼 수 있다. 예를 들어,  $T_1$ 에 대응하는 정수 엘리먼트 2는  $T_1$ 에 대응하는 선점형 태스크들의 조합들의 인덱스 중 2번 인덱스를 나타낸다. 만약  $T_1$ 의 활성화 시간이 15이고, 종료 시한이 20, 수행 길이가 3이라면 가용 타임 슬롯의 수는  $(D_1 - A_1)$ 의 값인 5가 되고, 이 때의 수행 길이가 3이기 때문에 타임 슬롯에 할당 할 수 있는 경우의 수는  ${}_5C_3$  가지가 되므로 5개의 타임 슬롯(15, 16, 17, 18, 19)에서 3개의 슬롯에 할당되어질 수 있고, 조합에 따른 인덱

스는 0 ~ 9까지가 된다.  $T_1$ 은 현재 인덱스가 2이므로 타임 슬롯 할당의 결과는 (0, 1, 1, 0, 1)이 된다. 따라서  $T_1$ 의 가용 슬롯의 범위가 15 ~ 19이므로 타임 슬롯 16, 17, 19에  $T_1$ 이 수행될 것이다. 마찬가지로  $T_n$ 의 가용 타임 슬롯의 수는  $(D_n - A_n)$ 의 값인 3이 되고, 이 때의 수행 길이가 2이기 때문에 3개의 가용 타임 슬롯에서 2개의 타임 슬롯에 할당 할 수 있는 경우의 수는  ${}_3C_2$  가지가 되므로 인덱스는 0 ~ 2까지 부여 된다. 현재  $T_n$ 의 인덱스가 1이기 때문에 타임 슬롯의 할당 결과는 (1, 0, 1)이 되고, 가용 슬롯의 범위가 8 ~ 10이므로  $T_n$ 은 타임 슬롯 8, 10에 수행될 것이다.

비선점형 태스크의 경우 벡터의 정수 엘리먼트는 비선점형 태스크의 수행 시작 시간을 나타낸다. 즉,  $T_2$ 에 대응 하는 정수 엘리먼트 3은  $T_2$ 가 3번 타임 슬롯에서 수행을 시작한다는 것이다. 만일 이 때  $T_2$ 의 활성화 시간이 1이고, 종료 시간이 20, 수행 길이가 10이라면  $T_2$ 는 타임 슬롯 3부터 12에서 수행된다 [Lee et al. 2011b].

벡터들로 구성된 해 집단이 생성되면 이 해 집단에 대해 선택, 교차, 대치 연산을 이용하여 새로운 자식 해와 다음 세대를 생성할 수 있다. 염색체의 각각의 태스크들에 대응하는 정수 엘리먼트들은 유전 알고리즘 연산이 수행되면 다양한 해들을 생성할 수는 있지만, 이것은 기존의 해에 대한 재조합이기 때문에 부모의 유전 정보와는 다른 해는 만들어지지 않는다. 즉, 유전 연산의 결과가 최초 해 집단에 포함되어 있는 값의 범위를 벗어나지는 않지만 돌연변이는 전혀 다른 유전 정보를 생성해 버리기 때문에 변이 연산은 제외한다. 우수한 해가 선택될 확률을 높일 수 있게 품질 비례 룰렛 휠 선택 연산을 사용하여 선택된 2개의 부모 해는 교차 연산을 수행함으로써 자식 해를 생성한다. 또한 교차 연산으로 생성한 자식해가 기존의 해 집단 내에 존재하는 염색체와 동일한 경우 대치 연산을 이용하여 새로운 염색체로 대체되어지고, 세대를 거듭하면서 반복하게 되면 점점 최적의 해에 가까워지게 된다 [Lee et al. 2012a]. 전력 소비 스케줄의 피크로드에 대한 평가를 위해 적합도 함수는 슬롯 당 전력 소비의 합을 계산하고 그 중 가장 큰 값 피크로드를 찾는다.



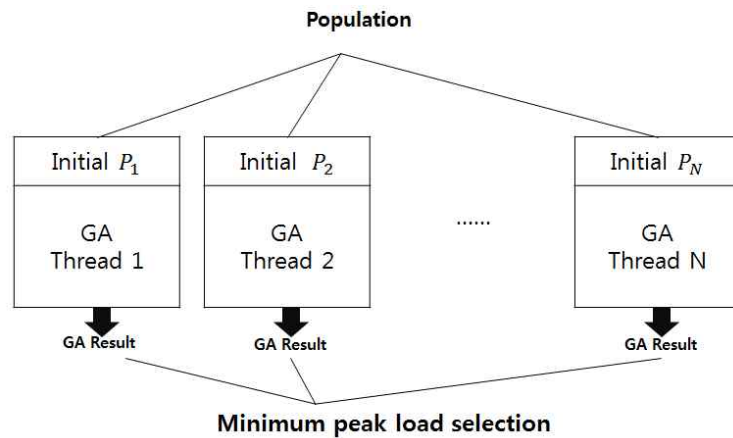


Figure 52. 유전 알고리즘을 이용한 멀티 쓰레드 전력 소비 스케줄링

최적화 문제에 이용되는 유전 알고리즘은 적은 시간에 만족할 만한 해를 도출해 낼 수 있다[Lee et al. 2013]. 따라서 유전 알고리즘은 슬롯 당 피크로드를 기반으로 하는 적합도 함수를 이용하여 전력 소비 태스크를 스케줄링 할 수 있다. 유전 알고리즘은 먼저 인 홈 네트워크를 통해 입력된 프로파일의 정보들을 만족할 수 있는 스케줄링의 결과들의 조합에 해당하는 해 집단을 생성할 수 있다. 이로부터 랜덤하게 초기 해 집단을 선택하게 되는데 이 때, 초기 해 집단을 어떻게 선택하느냐에 따라 결과의 품질이 달라질 수 있게 된다. 따라서 랜덤하게 초기 해 집단을 선택하는 것은 중요한 문제가 될 수 있다. 초기 해 집단을 선택한 후 선택과 교차 연산 후에 해에 대한 정렬이 이루어지게 되는데 초기 해 집단이 다양하게 되면 해 집단의 다양성은 보장되지만 수행 시간은 급격히 증가할 수 있다.

이런 이유로 초기 해 집단을 다양하게 생성하고 이를 각각의 쓰레드에 할당하여 각 쓰레드마다 유전 알고리즘 연산을 독립적으로 수행되게 한 후 이에 대한 결과를 수집한다. 이 때 각 쓰레드의 스케줄링 결과들 중 피크로드가 가장 적은 스케줄링을 선택한다. 더욱이 최근 멀티 코어 PC가 일반화되고 분산 처리 프레임워크의 사용이 보편화되면서 병렬처리가 가능하다. 쓰레드가 작업 할당의 단위가 되면서 쓰레드 기반 설계는 추가적인 하드웨어의 비용 없이 응용의 성능을 향상시키는 유용한 도구가 되고 있다.

## 2. 실험 및 결과

본 논문에서 제안된 기법은 Visual C++ 6.0을 통해 구현되어 Intel Core2 Duo CPU, 3.0GB 메모리, Window Vista 운영체제 기반의 개인용 컴퓨터에서 작동시켰다.  $M$ 은 스케줄링 윈도우의 타임 슬롯의 개수를 의미하는데, 이 때 하나의 태스크의 시작 시간은 0과  $M-1$  사이에서 랜덤하게 선택되어진다. 반면, 수행 시간은 지수적으로 분포하고, 태스크의 수행 길이는 스케줄링 윈도우 범위 안에서만 동작한다. 모든 태스크들은 보통의  $M-1$ 의 종료 시한을 갖는다.

실험은 쓰레드의 수에 따른 태스크들의 수, 해 집단의 크기, 피크로드에 대한 반복 횟수의 영향에 중점을 두며, 각각의 실험 파라미터의 설정을 위해 20개의 태스크 셋을 생성하고 그것들의 결과에 대해 평균을 낸다. 모든 실험들은 그들 고유의 태스크 수를 갖고 있고, 2개는 비선점 태스크이다.

첫 번째 실험은 해 집단의 크기는 50, 반복횟수는 100회로 설정하고, 쓰레드의 수를 1부터 10까지 변화시켜가면서 태스크의 수 5와 10인 경우에 대해 각각 피크로드를 측정한다.

선점형 태스크들은 작동이 시작된 후 태스크가 수행되고 동안에는 언제든지 중지되어질 수 있고 재개할 수도 있기 때문에 낮은 전력 소비를 가지고 있는 슬롯으로 유연하게 위치되어질 수 있다. 각각의 태스크는 동일한 평균 전력 소비량을 가지고 있기 때문에 태스크의 수가 증가함에 따라 피크로드도 함께 증가하게 된다. 하지만 스케줄러가 피크를 평준화함에 따라 피크로드는 태스크들의 수에 완전히 비례하지는 않는다. Figure 53과 같이 쓰레드의 수가 증가할수록 2개의 태스크의 경우 모두 피크로드는 감소한다. 또한 쓰레드의 수가 1일 때와 비교하여 쓰레드의 수가 10인 경우가 태스크의 수가 5개일 때, 피크로드가 약 4.5% 감소됨을 보인다.

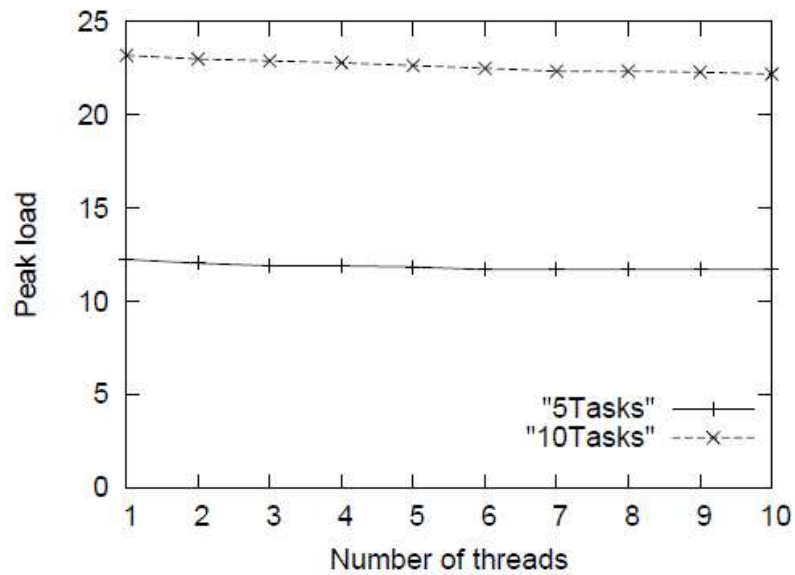


Figure 53. 피크로드에 대한 태스크 수의 영향

두 번째 실험은 쓰레드의 수가 1부터 10까지의 범위에 해당할 때 피크로드에 대한 해 집단의 크기의 영향을 알아보기 위해 반복횟수를 1,000회로 설정하여 측정한다. 첫 번째 실험과 같이 실험에 사용된 태스크 중 2개의 태스크는 선점형의 태스크이며 Figure 54와 같이 해 집단의 크기는 각각 20, 50, 100일 때의 피크로드 값을 비교 측정한다.

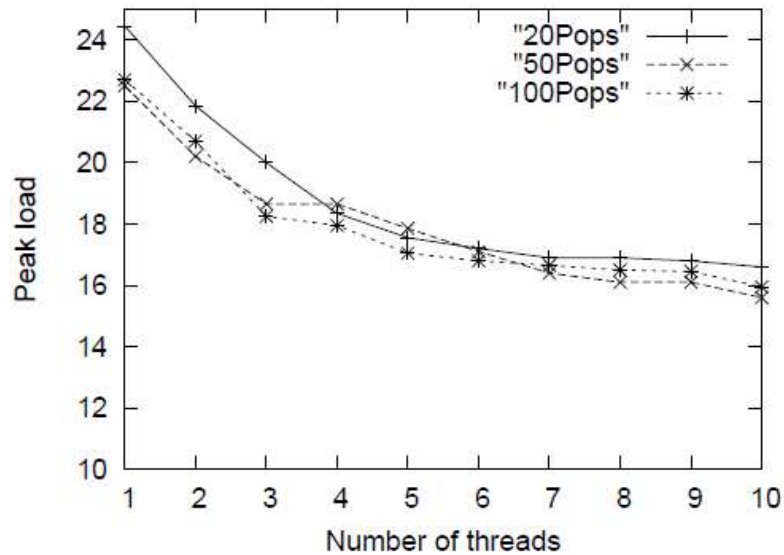


Figure 54. 피크로드에 대한 해 집단의 크기의 영향

그래프에서 보듯이 해 집단의 크기가 20인 경우 즉, 특히 해 집단의 크기가 작을 때 쓰레드의 실행이 피크로드를 줄이는데 상당히 기여하고 있음을 보여준다. 쓰레드가 1개일 때보다 10개일 때의 경우의 피크로드를 비교해보면, 해 집단의 크기가 50인 경우와 100인 경우가 각각 30.6%와 29.7%이지만 해 집단의 크기가 20인 경우는 32.1%로 피크로드를 감소시킨다. 일반적으로 해 집단의 크기가 작으면 실행 속도는 향상되지만 정확성은 떨어지게 된다. 그러나 멀티 쓰레드 컴퓨팅을 통해 실행 속도와 정확성 모두를 만족시킴으로써 이러한 문제를 극복할 수 있음을 나타내고 있다.

세 번째는 쓰레드의 수가 1부터 10까지일 때 피크로드에 대한 반복 횟수의 영향에 대한 실험이다. 10개의 태스크 중에서 2개의 선점형 태스크를 포함하고 있으며, 해 집단의 크기는 50이다.

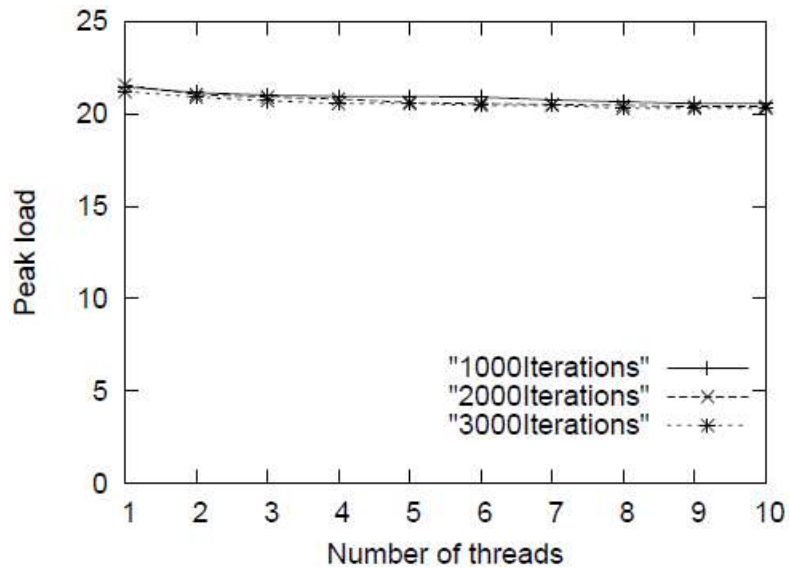


Figure 55. 피크로드에 대한 iteration의 영향

Figure 55는 반복 횟수를 각각 1,000, 2,000, 3,000회의 경우에 따라 쓰레드 수에 대한 피크로드를 나타낸다. 3,000회 반복하는 경우는 피크로드가 급격히 감소하지는 않지만 반복 횟수를 적게 하였을 때보다는 피크로드가 조금 더 감소한다. 평균 성능의 컴퓨터의 경우는 1,000회 반복하는데에 걸리는 시간은 1초 이하이다. 또한 근본적으로 반복 횟수는 태스크의 수와 전력 부하 프로파일에 따라 조정될 수 있는 파라미터이다. 두 번째 실험과 세 번째 실험의 결과로 보아 전력 스케줄링에서의 멀티쓰레드는 해 집단의 크기가 작고 반복 횟수가 1,000회일 때, 매우 효율적으로 동작한다.

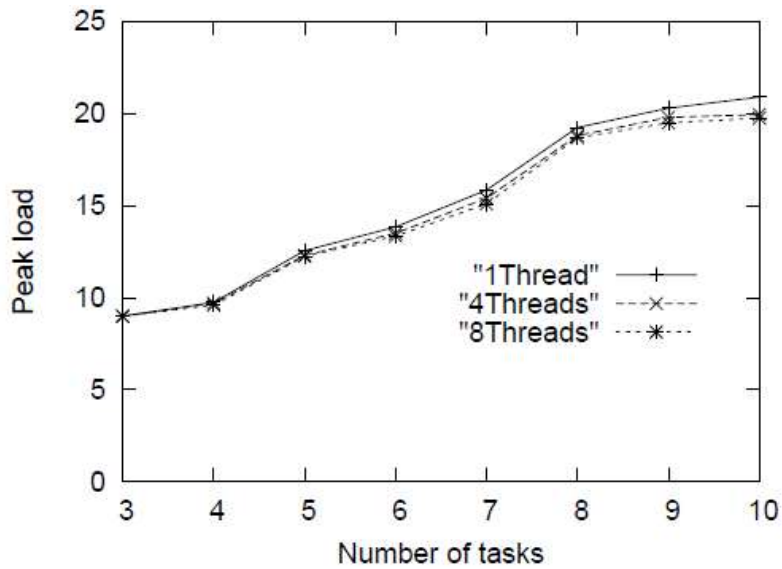


Figure 56. 피크로드에 대한 쓰레드 수의 영향

네 번째 실험은 태스크 수를 3부터 10까지 변화시키면서 쓰레드의 수가 1, 4, 8 일 경우의 피크로드를 비교 측정한다. 해 집단의 크기는 50이고, 반복 횟수는 2,000으로 설정한다. 실험 결과 Figure 56과 같이 태스크가 3인 경우에는 쓰레드의 수에 상관없이 동일한 피크로드를 나타내지만, 태스크의 수가 10인 경우에는 쓰레드가 많을수록 피크로드가 작아진다. 이 때의 피크로드 차이는 5.5%까지 나타난다. 즉 실험의 결과는 태스크의 수가 많은 경우는 쓰레드의 수가 많을수록 피크로드가 감소됨을 보인다.

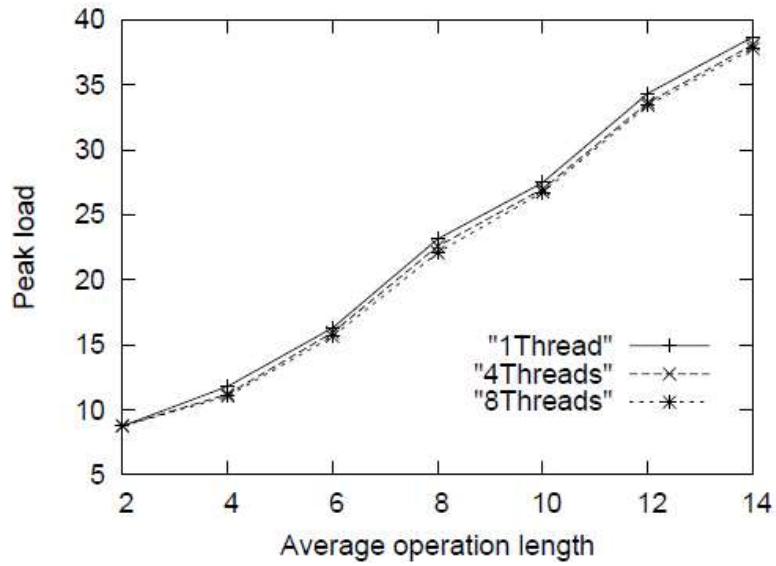


Figure 57. 피크로드에 대한 수행 길이의 영향

마지막 실험은 수행 길이에 따른 피크로드의 영향을 알아보는 실험으로 태스크의 수는 5, 해 집단의 크기는 50, iteration은 2,000회로 설정한다. Figure 57과 같이 수행 길이가 증가함에 따라 피크로드는 거의 비례하고 있으며 스레드가 8인 경우가 피크로드를 6.4% 감소시킨다. 그 차이는 수행 길이가 작을수록 더 커진다.

## VI. 결론

본 논문은 특정 시간대의 전력의 피크를 완화시키기 위해 마이크로그리드에서의 전력 수요 예측 및 백트래킹과 유전 알고리즘 기반의 효율적인 전력 소비 스케줄링 기법을 제안하고 그 성능을 분석하였다.

### A. 전력 수요 예측

월별 전력 수요 예측의 결과 2015년 4월의 경우 99.8%로 실제값과 예측값이 거의 일치함을 보였고, 15분별 전력 수요 예측의 결과 15분 후의 1일, 70일에 대한 수요 예측 오차는 단지 5% 범위 내로 다소 낮게 나타났다. 뿐만 아니라 요일별 평균 전력 수요의 예측 오차도 15분 후가 최대 0.2%로 실제값과 거의 차이가 없었다. 따라서 15분 후 예측에 대한 약간의 보정이 이루어진다면 마이크로그리드 발전 운용 계획을 위해 실제 전력 시스템에 적용해 볼 수 있다. 또한 V2G(Vehicle to Grid)의 전력 거래 장치에 전력 수요 예측 기법을 결합하여 전기 자동차로부터 전력 구입 계획을 유용하게 할 수 있다.

### B. 백트래킹 기반의 전력 소비 스케줄링

제안된 전력 소비 스케줄링 기법은 전력 소비 프로파일을 컴퓨터 알고리즘 즉, 백트래킹을 통해 사용자의 요구사항을 최대한 준수하면서 동시에 전력 소비 동작을 분산시키고, 이에 따라 피크로드를 23.1%까지 감소시켰다. 또한 가지치기를 통해 불필요한 확장을 줄임으로써 수행 시간을 약 2%까지 단축시켰으며, 전체적인 전력 조정을 하지 않고 각 로컬의 조정만으로도 피크 부하를 16%까지 감소시켰다. 뿐만 아니라 5개의 로컬 스케줄링 그룹에서의 태스크 수에 따른 최대 전력 피크의 측정된 결과 제안된 스케줄링 기법은 *Earliest* 기법보다 안정적인 그



래프의 모양을 나타냈으며, 태스크의 수가 4일 때, 피크를 20.4%까지 감소시켰다.

### C. 유전 알고리즘 기반의 멀티 쓰레드 전력 소비 스케줄링

유전 알고리즘 기반의 멀티 쓰레드 전력 소비 스케줄링 기법은 슬롯 당 피크로드를 기반으로 하는 적합도 함수를 이용하여 피크로드를 감소시키면서 수행시간을 향상시킬 수 있는 전력 소비 스케줄을 생성하였다. 또한 유전 알고리즘의 초기 해 집단을 다양하게 함으로써 솔루션의 다양성을 만족시켰고, 멀티 쓰레드를 통해 스케줄러의 수행 시간을 상당히 단축시킬 수 있었다. 실험 결과 쓰레드의 수가 증가할수록 태스크의 수가 5개인 경우가 태스크의 수가 10개인 경우보다 피크로드가 약 4.5% 감소됨을 보였고, 유전 알고리즘의 반복 횟수가 많을수록 피크로드가 조금 더 감소하였다. 태스크 수를 변화 시켜가며 쓰레드의 수에 따른 피크로드를 측정된 결과도 태스크의 수가 10인 경우에는 쓰레드가 많을수록 피크로드는 감소하였다. 이 때 피크로드 차이는 5.5% 까지 나타났다. 마지막으로 수행 길이에 따른 피크로드의 영향을 알아보는 실험의 결과는 수행 길이가 증가함에 따라 피크로드는 거의 비례하고, 쓰레드가 8인 경우가 피크로드를 6.4% 감소시켰다.

전력 수요 예측과 전력 소비 스케줄링 기법은 Figure 58과 같이 하나의 프로그램으로 구현되고, 서비스되어질 수 있다. 먼저 가정과 빌딩, 전력 시스템, 그리고 전기 자동차에서 생산된 전력 소비 프로파일은 IHD나 PC 등을 통해 사용자에게 의해 직접 입력되거나 스마트 전력 기기들에 의해 자동적으로 입력되어진다. 여기에 신재생 에너지 정보와 실시간 변화하는 가격 정보 또한 결합될 수도 있다. 구현된 프로그램은 입력된 프로파일을 토대로 전력 수요를 예측할 수 있으며 추가적으로 전력 소비에 대한 패턴 분석도 가능하다. 이를 통해 산출된 정보는 제안된 전력 소비 스케줄링의 기초 자료로도 활용이 되고, 전력 소비 스케줄러는 입력된 전력 소비 프로파일을 실시간 태스크 모델로 변환하여 피크 시 전력 사용량을 분산시키는 스케줄을 생성한다. 향후 프로그램은 웹서비스나 스마트 어플

리케이션과 연동되어 사용자들에게 전력 소비 패턴과 전력 수요 예측, 그리고 전력 소비 방식과 충전 방식 등의 서비스를 제공할 수 있다.

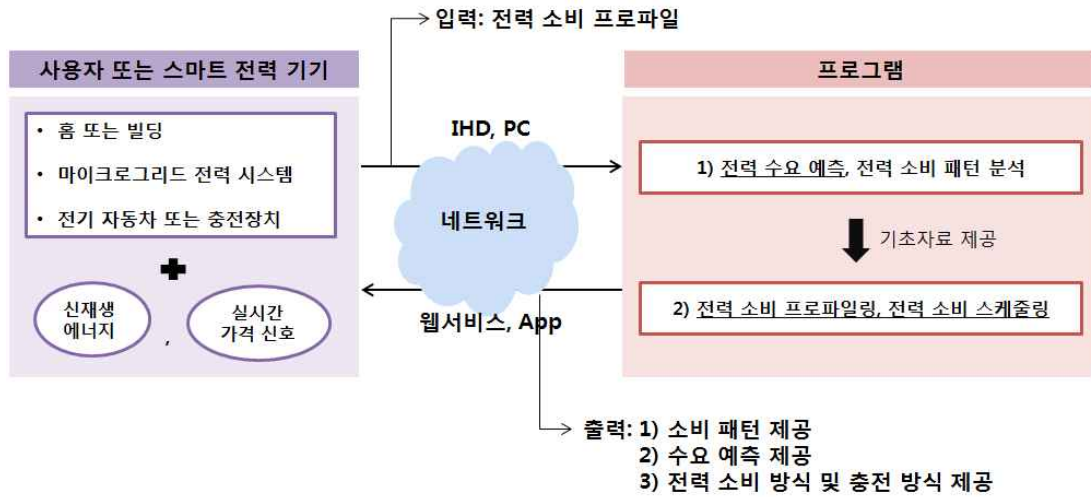


Figure 58. 제안된 기법의 서비스 구조

## VII. References

- [Arbaz et al. 2015] Arbaz, A., Waqas, M., Mahmood, A., 2015. Home Energy Management and Knapsack Technique in Smart grid Environment, International Conference on Engineering & Emerging Technologies (ICEET).
- [Barak et al. 2016] Barak, S., Sadegh, S., 2016. Forecasting Energy Consumption using Ensemble ARIMA-ANFIS Hybrid Algorithm, International Journal of Electrical Power & Energy Systems, pp. 92~104.
- [Beccali et al. 2004] Beccali, M., Cellura, M., 2004. Forecasting Daily Urban Electric Load Profiles using Artificial Neural Networks, Energy Conversion and Management, pp. 2789~2900.
- [Derin et al. 2010] Derin, O., Ferrante, A., 2010. Scheduling Energy Consumption with Local Renewable Micro-generation and Dynamic Electricity Prices, First Workshop on Green and Smart Embedded System Technology: Infrastructures, Methods, and Tools.
- [Erodogdu 2007] Erodogdu, E., 2007. Electricity Demand Analysis using Cointegration and ARIMA Modelling: A case study of Turkey, Energy Policy, 2007, pp. 1129~1146.
- [Facchinetti et al. 2010] Facchinetti, T., Bibi, E., Bertogna, M., 2010. Reducing the Peak Power through Real-time Scheduling Techniques in

Cyber-physical Energy Systems, First International Workshop on Energy Aware Design and Analysis of Cyber Physical Systems.

[Gellings 2009] Gellings, C., 2009. Enabling Energy Efficiency and Demand Response, The Fairmont Press.

[Gislason 2008] Gislason, D., 2008. ZIGBEE Wireless Networking, Newnes, Burlington, USA.

[Ipakchi et al. 2009] Ipakchi, A., Albuyeh, F., 2009. Grid of the Future, IEEE Power & Energy Magazine, pp. 52~62.

[Kim et al. 2010] Kim, T., Lee, S., Lee, S., 2010. Optimization of Home Loads scheduling in Demand Response, The Journal of The Korean Institute of Communication Sciences 2010, pp. 1407~1415.

[Kim et al. 2012] Kim, H., Lee, J., Park, G., 2012. Constraint-Based Charging Scheduler Design for Electric Vehicles, ACIIDS 2012, Part III, Lecture Notes in Artificial Intelligence(LNAI) 7198, pp. 266~275.

[Kim et al. 2014] Kim, H., Park, G., Lee, J., 2014. Incorporative Relocation Team Planning and Staff Member Allocation in Electric Vehicle Sharing Systems, International Journal of Multimedia and Ubiquitous Engineering(IJMUE), pp. 249~256.

[Lee et al. 2014] Lee, E., Bahn, H., 2014. A Genetic Algorithm Based Power Consumption Scheduling in Smart Grid Buildings, Information Networking (ICOIN), 2014 International Conference, pp. 469~474.

[Lee et al. 2011a] Lee, J., Park, G., Kang, M., Kwak, H., Lee, S., 2011. Design

of a Power Scheduler Based on the Heuristic for Preemptive Appliances, ACIIDS 2011, Lecture Notes in Artificial Intelligence(LNAI) 6591, pp. 396~405.

[Lee et al. 2011b] Lee, J., Park, G., Kwak, H., Jeon, H., 2011. Design of an Energy Consumption Scheduler Based on Genetic Algorithms in the Smart Grid, ICCCI 2011, Part I, Lecture Notes in Computer Science(LNCS) 6922, pp. 438~447.

[Lee et al. 2011c] Lee, J., Park, G., Kim, H., Jeon, H., 2011. Fast Scheduling Policy for Electric Vehicle Charging Stations in Smart Transportation, ACM RACS 2011, pp. 110~112.

[Lee et al. 2011d] Lee, J., Park, G., Kim, H., Kim, C., Kwak, H., Lee, S., Lee, S., 2011. Intelligent Management Message Routing in Ubiquitous Sensor Networks, ICCCI 2011, Part I, Lecture Notes in Computer Science(LNCS) 6922, pp. 537~545.

[Lee et al. 2011e] Lee, J., Park, G., Kim, H., 2011. Multithreaded Power Consumption Scheduler Based on a Genetic Algorithm, FGCN 2011, Part I, COMMUNICATIONS IN COMPUTER AND INFORMATION SCIENCE 265, pp. 47~52.

[Lee et al. 2011f] Lee, J., Kim, H., Park, G., 2011. Power Consumption Scheduler Integrating Renewable Energies for Smart Grid, International Conference on Internet (ICONI), pp. 353~357.

[Lee et al. 2011g] Lee, J., Park, G., Kim, S., Kim, H., Sung, C., 2011. Power Consumption Scheduling for Peak Load Reduction in Smart Grid

Homes, 2011 ACM Symposium on Applied Computing, pp. 584~588.

[Lee et al. 2011h] Lee, J., Park, G., Kim, H., Kang, M., Kim, E., 2011. Randomization Effect Measurement on the Fast Power Consumption Scheduler, Advances in Electrical Engineering & Electrical Machines, Lecture Notes in Electrical Engineering(LNEE) 134, pp. 137~152.

[Lee et al. 2011i] Lee, J., Park, G., Kim, H., 2011. Reservation-Based Charging Service for Electric Vehicles, ICA3PP 2011 Workshops, Part II, Lecture Notes in Computer Science(LNCS) 7017, pp. 186~195.

[Lee et al. 2012a] Lee, J., Kim, H., Park, G., 2012. A Genetic Scheduler for Electric Vehicle Charging, ACM Symposium on Applied Computing, pp. 1843~1844.

[Lee et al. 2012b] Lee, J., Kim, H., Park, G., Jeon, H., 2012. Design of a Heuristic-Based Charging Scheduler for Electric Vehicles, International Information Institute (Tokyo). Information 15.7, pp. 2913~2922.

[Lee et al. 2012c] Lee, J., Park, G., Kang, M., Kim, J., Kim, H., Kim, I., Ko, Y., 2012. Design of Efficient Matching-Based Relocation Scheme for Electric Vehicle Sharing Systems, ASNT 2012, Communication in Computer and Information Science(CCIS) 341, pp. 109~115.

[Lee et al. 2012d] Lee, J., Kim, H., Park, G., Kang M., 2012. Energy Consumption Scheduler Demand Response System in the Smart Grid, JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 28, pp. 955~969.

- [Lee et al. 2012e] Lee, J., Kim, H., Park, G., Jeon, H., 2012. Genetic Algorithm-Based Charging Task Scheduler for Electric Vehicles in Smart Transportation, ACIIDS 2012, Part I, Lecture Notes in Artificial Intelligence(LNAI) 7196, pp. 208~217.
- [Lee et al. 2012f] Lee, J., Kim, H., Park, G., 2012. Integration of Battery Charging to Tour Schedule Generation for an EV-based Rent-a-car Business, ICSI 2012, Part II, Lecture Notes in Computer Science(LNCS) 7332, pp. 399~406.
- [Lee et al. 2012g] Lee, J., Kim, H., Park, G.: Measurement of Global Peak Load reduction by Power Consumption Scheduling for Smart Places, Recent Researches in Applications of Electrical and Computer Engineering, pp. 167~171.
- [Lee et al. 2012h] Lee, J., Kim, H., Park, G., 2012. Relocation Action Planning in Electric Vehicle Sharing Systems, MIWAI 2012, Lecture Notes in Computer Science(LNCS) 7694, pp. 47~56.
- [Lee et al. 2012i] Lee, J., Kim H., Park, G., 2012. Renewable Energy and Power Management in Smart Transportation, ACIIDS 2012, Part III, Lecture Notes in Artificial Intelligence(LNAI) 7198, pp. 247~255.
- [Lee et al. 2012j] Lee, B., Lee, S., Im, D., Kim, H., Park, G., Lee, J., 2012. Tour and Charging Scheduler Development Based on Simulated Annealing for Electric Vehicles., IUrc 2012, Communication in Computer and Information Science(CCIS) 353, pp. 189~194.

- [Lee et al. 2012k] Lee, J., Kim, H., Park, G., Lee, B., Lee, S., Im, D., 2012. Tour Schedule Generation Integrating Restaurant Options for Electric Vehicles, CUTE 2012, Ubiquitous Information Technologies and Applications, Lecture Notes in Electrical Engineering 214, pp. 299~306.
- [Lee et al. 2013] Lee, J., Park, G., Kim, H., Lee, B., Lee, S., Im, D., 2013. Effect of Genetic Parameters in Tour Scheduling and Recommender Services for Electric Vehicles, GPC 2013, Lecture Notes in Computer Science(LNCS) 7861, pp. 556~562.
- [Lee et al. 2016] Lee, J., Lee, Y., Han, Y., Han, S., Ko, D., Ju D., Park, G., 2016. Power Consumption Model of a Microgrid Based on Utility-providing History Data, The IIER international Conference, pp. 29~31.
- [Lin et al. 2010] Lin, S., Guo, X., Chen, W., Zhang, W., Lin, Y., 2010. An Automation Model for the Building Energy Management Systems: A Theoretical Study, International Conference on Robotics, Control and Manufacturing Technology, pp. 41~46.
- [M. Jujano-Rojas et al. 2012] M. Lujano-Rojas, J., Monterio, C., Dufo - Lopez, R., L. Bernal-Agustin J., 2012. Optimum Residential Load Management Strategy for Real Time Pricing (RTP) Demand Response Programs, Energy Policy Volume 45, pp. 671~679.
- [Mady et al. 2010] Mady, A., Boubekeur, M., Provan, G., 2010. Optimised Embedded Distributed Controller for Automated Lighting Systems, First Workshop on Green and Smart Embedded System Technology: Infrastructures, Methods, and Tools.



- [Mehrshad et al. 2013] Mehrshad, M., Tafti, A., Effatnejad, R., 2013. Demand-side Management in the Smart Grid Based on Energy Consumption Scheduling by NSGA-II, International Journal of Engineering Practical Research (IJEPR), pp. 197~200.
- [Mohsenian-Rad et al. 2010] Mohsenian-Rad, A., Leon-Garcia, A., 2010. Optimal Residential Load Control With Price Prediction in Real-Time Electricity Pricing Environments, IEEE Transactions on Smart Grid 1, pp. 120~133.
- [Montana et al. 1998] Montana, D., Brinn, M., Moore, S., Bidwell, G., 1998. Genetic Algorithms for Complex Real-Time Scheduling, IEEE International Conference.
- [Park et al. 2012] Park, K., Lee, Y., Doh, G., and Yoo, J., 2012. Scheduling Problem for Energy Efficiency Optimization in Smart Grid, KIIE/KORMS, pp.1187~1206.
- [Park et al. 2014] Park, Y., Ji, P., 2012. Development of Daily Peak Power Demand Forecasting Algorithm with Hybrid Type composed of AR and Neuro-Fuzzy Model, The Transactions of the Korean Institute of Electrical Engineers, pp. 189~194.
- [Shin et al. 2016] Shin, Y., Yoon, S., 2016. Electricity forecasting model using specific time zone, Journal of the Korean Data & Information Science Society, pp. 275~284.
- [Spees et al. 2007] Spees, K., Lave, L., 2007. Demand response and electricity

market efficiency, The Electricity Journal, pp. 69~85.

[Yousuf et al. 2007] Yousuf, M., El-shafei, M., 2007. Power line communications: An overview part I, International Conference on Innovations and Information Technology(ICIIT), pp. 218~222.

[Zhao et al. 2013] Zhao, Z., Lee W., Shin Y., Song, K., 2013. An Optimal Power Scheduling Method Applied in Home Energy Management System Based on Demand Response, ETRI Journal, pp. 677~686.

## 감사의 글

대학을 졸업하고 대학원에 들어오며 늘 최선을 다하며 노력하자는 마음으로 정신없이 달려오니 어느덧 박사학위 논문의 결실을 맺게 되었습니다. 학업에 임하면서 가끔씩 힘이 들거나 나태해질 때마다 대학원 면접 때 했던 말을 되새기며 초심을 잃지 말고 열심히 하자고 자신을 채찍질할 때도 있었지만, 이제 이 논문을 끝으로 대학원을 마무리하려고 하니 시원하기도 하고 한편으로는 아쉬운 마음도 듭니다. 그러나 그 어떠한 마음보다 앞서 느끼는 것은 저를 항상 응원해주시고 도움을 주신 분들에 대한 고마움입니다. 이에 그동안 저를 아껴주신 모든 분들에게 이 논문의 마지막 장을 통해 감사의 말씀을 드립니다.

먼저 아버지같은 마음으로 저에게 항상 어려운 점을 물어주시고, 고충을 함께 나누어주시며 현명히 해결해 나갈 수 있도록 도와주신 박경린 교수님과 이정훈 교수님께 진심으로 감사드립니다. 부족한 저에게 더욱 발전할 수 있도록 많은 기회를 주시며 늘 세심하게 보살펴주셨기에 그동안의 모든 것들이 가능했습니다. 많은 관심과 따뜻한 격려로 그리고 아낌없는 조언으로 오랜 시간 늘 변함없이 저를 지도해주신 교수님 두 분의 고마움은 앞으로도 계속 가슴 깊이 남아있을 것이고, 제가 어느 곳에서 무엇을 하든 항상 같이 할 것입니다. 또한 학부시절부터 깊은 관심과 배려로 보살펴주셨던 김철수 교수님, 이봉규 교수님께 감사드리며, 바쁘신 중에도 미흡한 저의 논문 심사를 기꺼이 맡아주셨던 고봉운 교수님, 박찬정 교수님, 송준모 교수님께도 감사드립니다.

또한 학교생활 내내 지원을 아끼지 않아 주었던 학과 조교, 그리고 학과 후배들에게도 고마움을 전합니다. 더불어 제가 힘들 때마다 위로가 될 수 있었던 친구들과 언니들, 그리고 항상 가까이에서 저를 지켜봐주신 많은 분들께도 고마움을 전하며 기쁨을 함께 나누고 싶습니다.

무엇보다도 저를 늘 걱정해주고 곁에 있는 것만으로도 힘이 되어주신 사랑하는 어머니와 늘 내편이 되어주고 세심한 부분까지 외조에 힘써준 나의 남편 김용언 씨, 그리고 항상 저를 챙겨주시고 아껴주시는 이모와 친척 식구들을 비롯하여, 부족한 저에게 무한한 사랑을 주시며 아껴주시는 시부모님과 시댁 식구들에게 마음 깊이 감사드리고, 부족하지만 대학원 생활의 마지막 결실인 이 논문을 바칩니다.

끝으로 ‘나는 똑똑한 것이 아니라 단지 문제를 더 오래 연구할 뿐이다’라는 아인슈타인의 말을 가슴 깊이 새기며 저를 아껴주시는 모든 분들의 사랑과 기대에 어긋나지 않도록, 그리고 언젠간 제가 그 사랑에 보답할 수 있는 사람이 될 수 있도록 노력하고 또 노력하며 열심히 살겠습니다. 고맙습니다.

2016년 12월

김혜진 올림