

전기차 충전소 예약 시스템에서 효율적인 EV 배치 기법

Result Report
그래프이론과 머신러닝연구
2023.06.07
컴퓨터공학과
80221115
이 호 국

1. 서론

전 세계적으로 환경오염 및 기후변화 대응을 위한 친환경 에너지 사용 확대 추세에 따라 전기차 보급이 증가하고 있다. 현재 내연기관 차량 대비 높은 가격대인 전기차의 판매율을 높이기 위해서는 보다 저렴한 요금으로 이용할 수 있는 충전 인프라 구축이 필수적이다. 이를 위해서는 충전 사업자들을 위한 편리한 서비스 공급이 필요한 실정이지만, 현실은 플랫폼은 물론 인프라마저 열악한 상황이다. 따라서 본 연구에서는 전기차 충전소 사업자를 위한 스케줄링 알고리즘을 구현해 보고, 그 방법과 향후 과제를 제시하고자 한다.

2. 연구 배경

내연기관이나 수소연료 차량들에 비해서 전기차(EV)는 그 충전 시간이 훨씬 길다. 급속충전이라는 새로운 대안이 떠오르고 있지만, 금액적으로 부담이 크다는 단점이 있다. 이와 더불어 전기차의 배터리에 저장된 전기를 실시간으로 판매하는 V2V 사업 등을 고려해 볼 때, 장시간의 충전 시간 동안 EV를 효율적으로 사용하는 방법이 중요해졌다.

특히, 충전 서비스 사업자(Charging Service Operator, CSO)에게 충전소에서 전기차의 효율적인 충전 스케줄링은 매우 중요한 문제이며, 방법에 따라 충전 속도와 비용 등 사업체의 경쟁력을 결정짓는 중요한 기준이 될 수 있다. 따라서 본 연구에서는 전기차 충전 서비스 사업자가 본인들의 충전소에 적용할 수 있는 스케줄링 알고리즘을 구현해보고, 예시 데이터를 이용해 검증하며, 한계점과 향후 목표를 알아보하고자 한다.

3. 연구 목표

전기차 충전을 위해서 고려해야할 기준은 시간, 돈, 거리, 충전 속도, 부가 서비스 등 여러가지가 있다. 이 중 전기차 충전 사업자의 입장에서 서비스 경쟁력 및 높은 회전율을 통한 이익의 극대화를 위한 가장 중요한 요소는 시간이다.

본 연구에서는 특별한 상황을 가정한다. 어떤 전기차 소유주가 전기차 충전소에 미리 다음 날 충전 예약을 걸어 두었고, 충전소에서는 해당 시간 안에 원하는 용량의 충전을 모두 완료해야 한다. 이런 방식으로 존재하는 내일의 예약이 여러 건이 존재하며, 알고리즘 구현의 편의를 위해 특정 시간 범주동안 예약을 해 두고, 그 예약 시간 안에 특정 시간동안 충전소에서 충전이 이루어지는 상황을 고려했다. 이 때 전기차 충전소에는 N 개의 충전기가 있다고 가정하며, 이는 곧 동시에 최대 N 대의 차량을 충전할 수 있다는 의미이다. 즉, 하나의 충전소 내부에 N 개의 충전기가 있을 때, 전체 예약에 대해서 예약 시간 안에 최대한 많은 충전이 가능하도록 알고리즘을 제안하고자 한다. 이를 Python의 Module 중 NetworkX에서 제공하는 이분 그래프(Bipartite Graph)를 이용해 위 알고리즘을 구현한다.

4. 연구 방법

이전의 연구 목표에서 제안한 목표를 달성하기 위해서, 왼쪽 노드는 예약을 한 차, 오른쪽 노드는 시간을 범주화 한 시간 블록(time block)을 배치해, 서로 연결하면 전체 예약에 대한 이분 그래프를 만들 수 있다. 이 때 전기차에서는 원하는 용량의 전기를 충전해야 하기 때문에 오른쪽 노드에서 연결되는 간선의 개수가 정해져 있고, 오른쪽 노드인 시간 블록에서 연결되는 간선은 최대 개수가 N 으로 고정되어 있다. 가장 먼저 고려해 본 것은 충전소에 있는 충전기가 1대 일 때($N=1$), 사용할 수 있는 NetworkX에서 자체적으로 제공하는 Max-Matching 알고리즘이다. 하지만, 이번 연구에서는 $N=1$ 이외의 상황을 고려해야 하므로, 새로운 알고리즘의 구현이 필요하다.

따라서 본 연구에서는 Max-N-Condition-Matching 알고리즘을 제안한다. 이 알고리즘은 기존의 Max-Matching 문제에서 두 가지의 조건이 추가된 것으로 [1] 왼쪽 노드에서 연결되는 간선의 최소 개수 조건(condition)이 있고, [2] 오른쪽 노드에서 연결되는 간선의 최대 개수 조건(N)이 존재한다. 기존의 Max-Matching 문제보다 제한 조건이 구체적이어서, NetworkX에서 구현한 Hopcroft-Karp Algorithm을 매핑한 함수가 아닌 DFS를 이용한 새로운 방법을 제시한다.

시간 구분 No.	시작 시간	종료 시간
1	7:00	7:30
2	7:30	8:00
3	8:00	8:30
4	8:30	9:00
5	9:00	9:30
6	9:30	10:00
7	10:00	10:30
8	10:30	11:00
9	11:00	11:30
10	11:30	12:00

그림 1 시간 블록 범주화

id	start	end	real
a	1	3	2
b	2	3	2
c	2	5	3
d	3	5	2

그림 2 입력 데이터

사용자 ID	충전 시작 시간	충전 종료 시간	총 충전 시간	필요 충전 시간
a	1	3	3	2
b	2	3	2	2
c	2	5	4	3
d	3	5	3	2

그림 3 입력 데이터의 의미

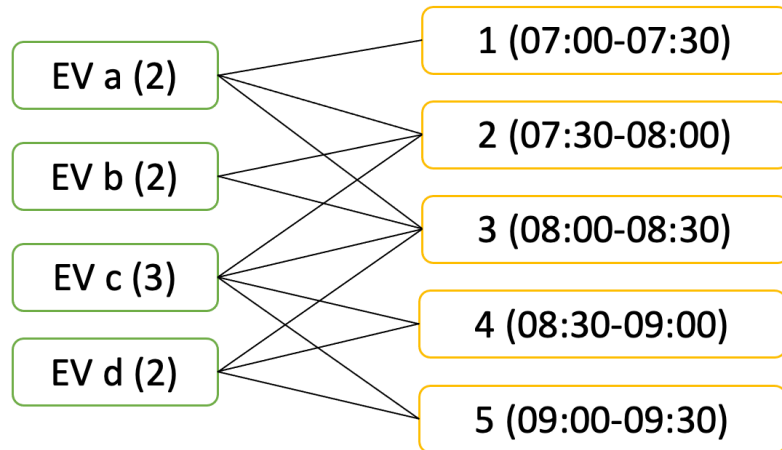


그림 4 입력 데이터 이분 그래프

가장 먼저, 입력 데이터 처리를 진행한다. 입력 데이터는 예약 정보이며, 그림 2 과 같은 형태로 가공한다. 그림 2에서 id는 EV의 예약 세션의 id를 지정하며, start 와 end 는 예약의 시작/종료 시간 블록을 의미한다. real은 전체 예약 시간 중 실제로 충전하는데 필요한 시간 블록의 양을 의미한다. 예를 들어, a번 세션의 예약의 경우 07:00 ~ 08:30 동안 충전을 예약했으며, 이 1시간 30분의 충전 예약 시간 중 총 1시간 동안 충전해야 함을 의미한다. 이를 이분 그래프로 표현하면 그림 4 와 같다.

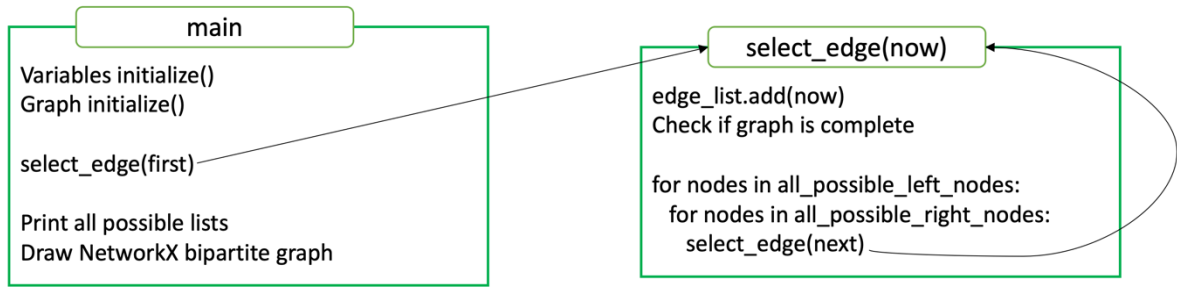


그림 5 재귀함수를 이용한 Max-N-Condition-Matching 알고리즘

Max-N-Condition-Matching 알고리즘의 간단한 수도코드는 그림 5와 같다. 처음에 각 변수와 그래프를 초기화 해준 후, 재귀함수를 이용해 `select_edge()` 재귀함수를 DFS로 순회하며 간선을 가능한 모든 노드들에 대해서 선택하고, 이 때마다 각 왼쪽 노드와 오른쪽 노드에 필요한 조건이 맞는 지 확인해 그래프가 완성되었으면 최종 리스트에 해당 그래프를 추가하고, 이어서 순회를 계속 진행하는 방식이다. 결과적으로, 위 알고리즘을 통해 조건에 맞게 생성된 가능한 모든 이분 그래프를 찾을 수 있었다.

5. 연구 결과 분석

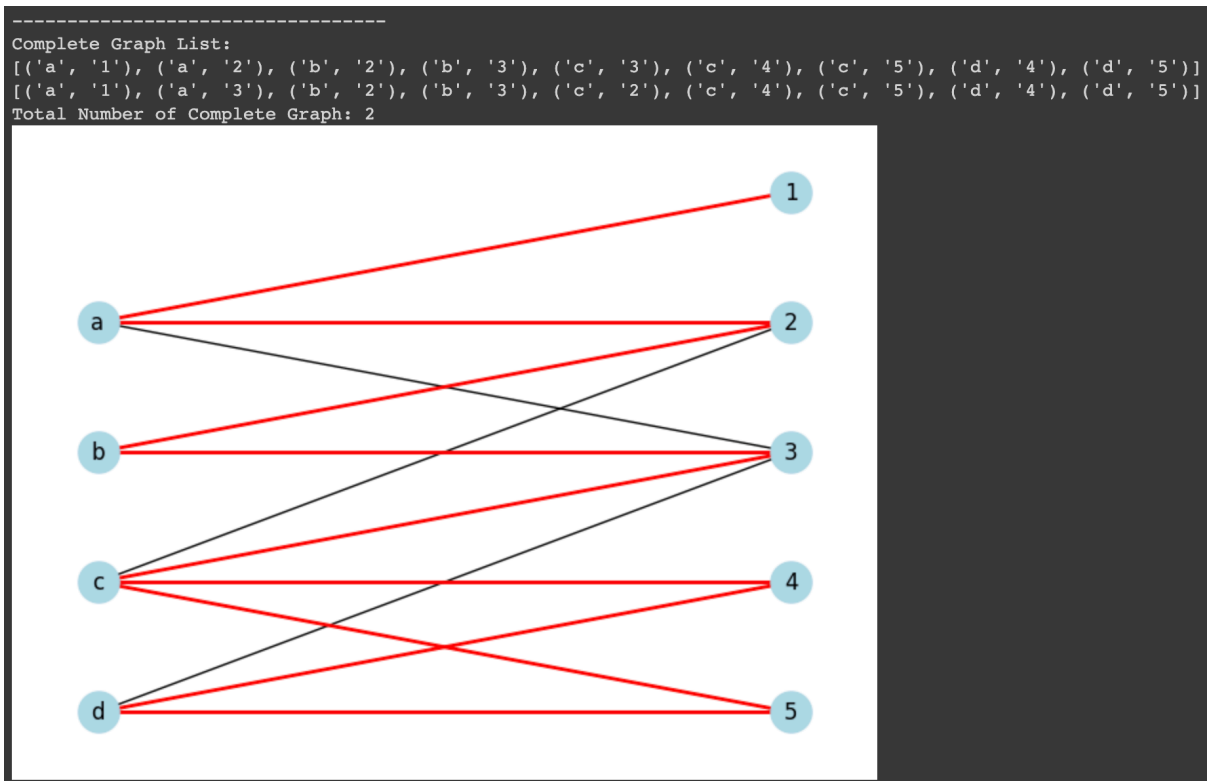


그림 6 예시 데이터의 알고리즘 결과

첫 번째 예시로, 그림 2를 입력 데이터로 알고리즘을 실행하면, 그림 6과 같은 2가지의 가능한 결과와 상위 1개의 이분 그래프가 출력된다.

id	start	end	real
a	1	5	2
b	1	5	2
c	1	5	2
d	1	5	2

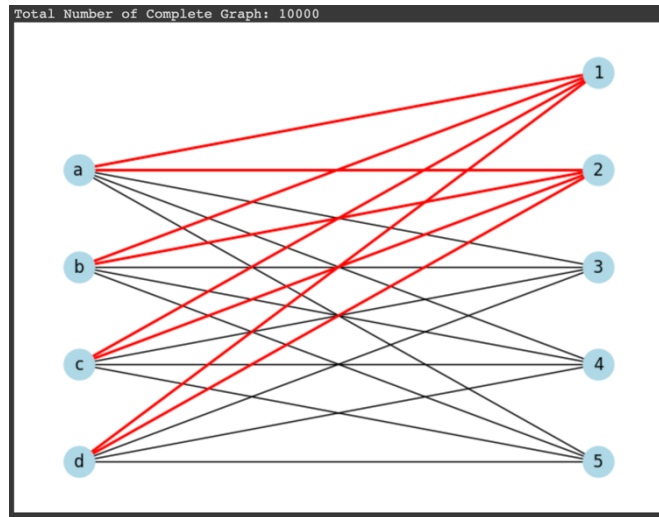


그림 7 예시 데이터 2 와 알고리즘 결과 (N=5)

```
Invalid Input: Total needed time exceeds the maximum available charging time
Total needed time : 20
maximum available charging time : 5
```

그림 8 예시 데이터 2 의 알고리즘 결과 (N=1)

그림 7 은 알고리즘을 테스트하기 위해 제작한 데이터 셋으로, N=5이며, 총 10000개의 가능한 모든 이분 그래프를 찾아서 반환해 준다. 그림 8에서는 N=1로 설정해 일부러 불가능한 충전 경우를 만들어 테스트를 실행한 모습으로, 충전 스케줄링의 성공/실패를 잘 구분할 수 있음을 보여준다.

6. 향후 계획

이번 연구에서는 오직 시간만을 이용한 스케줄링 알고리즘을 구현한 것으로 시간과 비슷한 중요도를 가진 가격이나 거리 등은 고려하지 않은 한계점이 있다. 차후에는, 충전기의 좌표나 충전 속도에 따른 시간 등을 알고리즘 구현 요소에 추가해 더 정교한 스케줄링 기법을 제시하는 연구를 진행해 보려 한다.