Term Project Final Report

: Charge of electric vehicles in a company

Vincent Meng

Hyun Gyu Shin

Contents

1. Description

2. Mathematical Model

3. NP Completeness

4. Approximation Algorithm

1) Simulated Annealing

2) Genetic Algorithm

3) Heuristic

5. Result

6. Conclusion and Discussion 1. Description

Nowadays, people try to protect the environment. There are many efforts to do, but the critical way to solve this problem is to increase the utilization of electric vehicles. In reality, many companies are making an effort to make this kind of vehicles, but it is not commercialized now.

To commercialize it, we need not only electric vehicles, but also so many electric charge buildings. However, the cost for charge all of the vehicles are pretty much. Therefore, we have decided to study how organize the electricity charge of car in a company to minimize the cost for charging electric vehicles.

Here is example. A Koran company has 11 different vehicles that are working with electricity. That is why during the night the company needs to charge all the vehicles for the next day. The figure 1 shows the battery capacity for each vehicle and the charge time.

The company use rapid terminal charge, thus a charge for a car is approximately 45 min (Depends type of vehicle see figure 1).

The total electricity capacity of the building where are charge the vehicles is 100KW/h.

Figure 1



Unfortunately is not easy to realize because a building can’t charge all of the vehicles completely. However, we can add other constraints to make this problem more realistic after solving this simplified problem first.

Indeed, the company wishes to minimize the cost to charge the vehicles. During the night some hours are more expensive than the other (see figure 2).

Figure 2



Figure 3 shows the t intervals:



Thus the objective is to verify if it is possible to charge all the vehicle during the night and find the less expensive. For this exercise we have create the data because we do not find any literature about the subject. However, the charge time and battery capacity for each vehicle are true.

Example for understanding of our problem

If the company wish recharge 4 vehicles Vi during the night (8pm -8am) and minimize the cost. The electricity cost is the cheapest between midnight until 6 am (see previous tab). The limit of the electricity that the building can deliver is 100 Kw/h

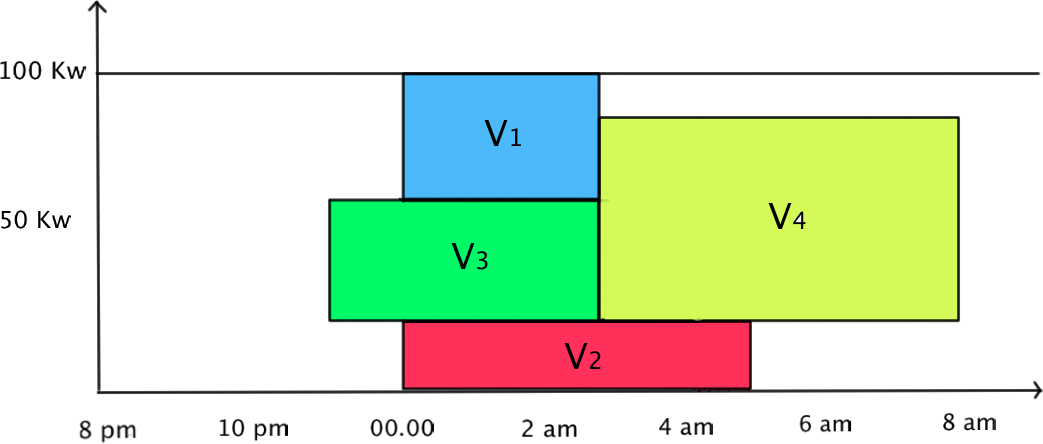
- V1 needs 40 Kw during 3 hours ( We’ll charge V1 during 3 hours with 40 KW. )

- V2 needs 20 Kw during 5 hours

- V3 needs 40 Kw during 4 hours

- V4 needs 70 Kw during 5 hours

This figure can explain our problem



We can charge all the vehicle completely, but it is **not optimal** solution. We’ll find it later.

2. Mathematical Model

In this part, we will explain the mathematical model for find the optimal solution of our problem. Firstly we will explain the basically mathematical model for our scheduling problem, then we will improve our model for adapts it in different situation.

Here is our model.

= Quantity charge vehicles type i number j during the intervals t

= Cost during the intervals t

= Demand in electricity for the vehicle type i number j

with , the capacity of the battery for each vehicle in Kw/h and , the total charge time for each vehicle.

Objective function

Constraints:

Constraint for the quantity of electricity available in the building

Constraint for the respect of need in electricity for each vehicle  
( We have to charge exactly Kw for unit hour. Maybe it has to be improved. )

Constraint for the respect of the capacity of the battery of each vehicle

With this mathematical model, we can find solution easily by using solver such as CPlex, excel solver.

Here is the solution we get with Excel solver.

A company has 5 cars and with recharge them with electricity during the night.

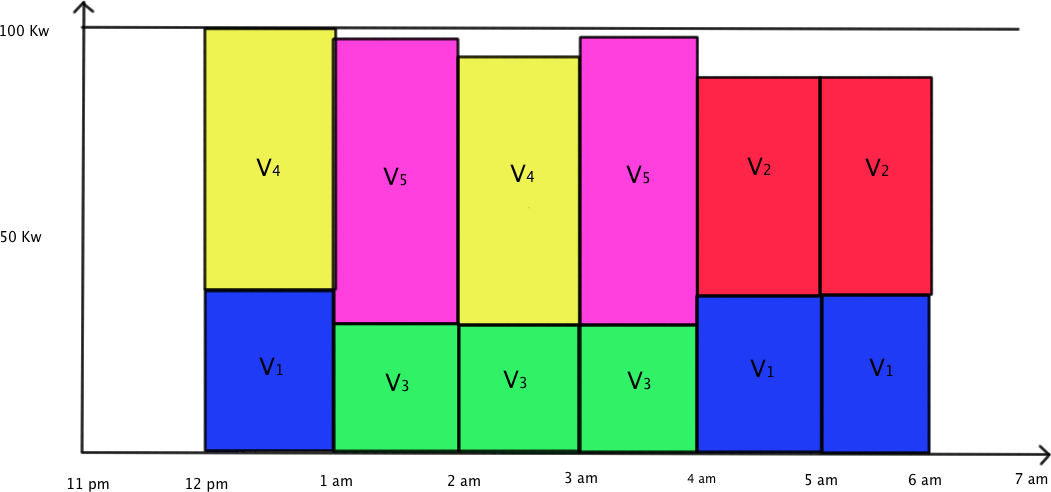




Solution:







Obj function: 44.96$

3. NP Completeness

We can prove that this problem is NP Complete.

To prove the NP Completeness, we need to show two things, NP and NP Hardness. Here is the proof of NP.

Theorem 3-1. This problem is NP

proof) Consider the decision problem. The decision problem is “Is there any plan with cost budget?“. Trivially, we can easily calculate the cost of any plan. Therefore, we can give a answer for the decision problem in polynomial time. Therefore, this problem is NP.

Then, the remaining job is to show that this problem is NP Hard.

Theorem 3-2. This problem is NP hard ( weak )

proof) To prove the NP hardness, we need to reduce a NP complete problem into our problem. Consider 1D Bin packing problem. The brief explanation of this problem is “ What is the minimum number of bin to insert all of the objects ? “  
 Let be the capacity of each bin, be the object . First, build time interval. Let be the interval i. Then we have to give the cost for each . Let be the cost of interval . The charging schedule should contain the charging capacity for each unit time. Simply take as the height of .  
 For example, suppose that we have 3 objects, each object have height (1, 4, 3), and the capacity of each bin is 4. Then make 3 time intervals, and the maximum charging capacity of each interval is 4 which is exactly same with the capacity of each bin. In addition, make charging schedule as (1, 4, 3). Then our job is almost done.

4

Because the object of 1D bin packing problem is to minimize the number of bin, we have to think about the proper cost for each interval to restrict the overuse of interval in our problem. To minimize the number of used interval, we use interval i if there is no capacity to insert among . Therefore, give cost more than fully charging cost of interval if when insert a object into interval . We can represent this mathematically.

If we insert a object into interval i, it takes the cost very much more than fully charging all the interval 1~i-1  
Give be . Then the above inequality is valid for all i.  
 With this , our problem have to use the minimum number of interval to charge. Therefore, return the number of used interval after getting result. Then it is the result of 1D bin packing problem. Therefore, our reduction is valid.

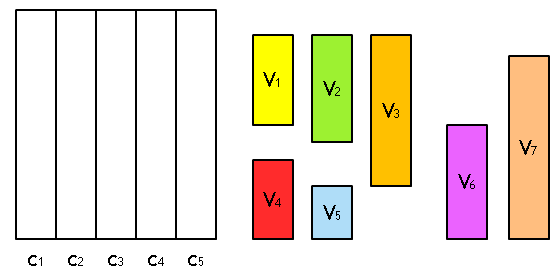
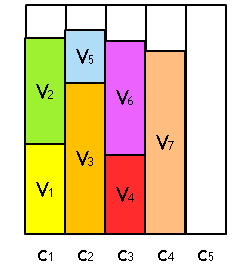
Therefore, our problem is NP-complete.

4. Approximation Algorithm

Because our problem is NP complete, there is no polynomial time algorithm to solve it. Therefore, we have to use approximation algorithm. To solve this, we use 3 approximation Algorithm which is Simulated Annealing, Genetic Algorithm, and Heuristic.

1. Simulated Annealing

To perform Simulated Annealing, we need to define sequence to represent current state. Consider the sequence = {1, 3, 2, 5, 4, 6, 7}. Then it represents the following state. At First, push into left most time interval as much as possible. Secondly, insert , in this order.

To make the left-most assign valid, we sort our cost into increasing order. Then push the jobs into left interval is always better than push it into right interval. Therefore, we can cover all of the possible candidate of the optimal solution with this sequence.

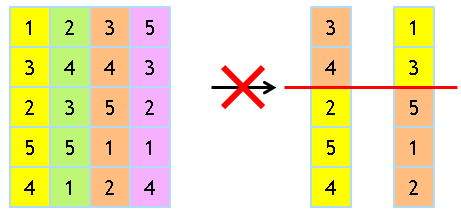
Then we can define the sequence as our state. Let be the cost of sequence . Then we can perform Simulated Annealing !

Here is our procedure

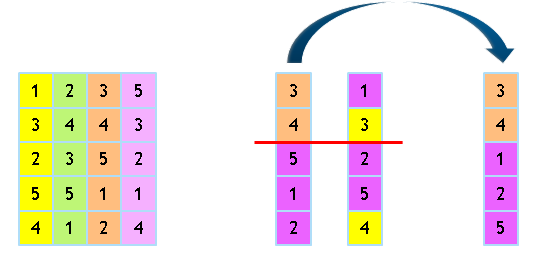
1. Set parameter L = 500, = L / , = 0.95. Set like that will take the new solution with probability 50% in the early stage.
2. Set initial solution with random sequence
3. Generate new solution by swapping two numbers from
4. Compare Cost() and Cost(). If new solution is better, then take it. Otherwise, take a new solution with probability .
5. Find the best solution with in the procedure.
6. Genetic Algorithm

To use genetic algorithm, we should define the genetic information. Each chromosome have a sequence, and each sequence represent a state which is exactly same with Simulated Annealing.

Next, we need to define crossover. Randomly choose the crossover point. In this case, just change each information is not valid because there can be duplication.



Therefore, we have to consider the other way. One way to perform crossover for chromosome x1 is to reorder the crossover part as the order of x2.



With this way, we can do crossover for two chromosomes.

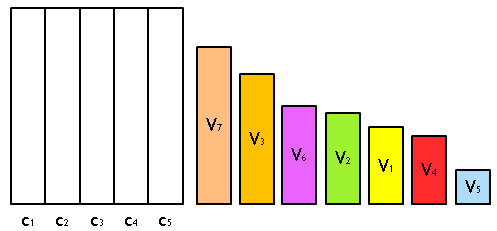
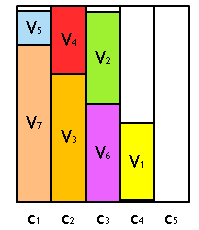
After doing crossover, consider the mutation procedure. It is very simple. For each chromosome, they have mutation with 10% probability. If they have, then just swap two numbers. Then it is mutation.

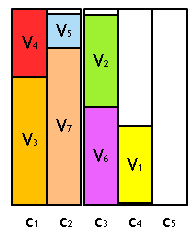
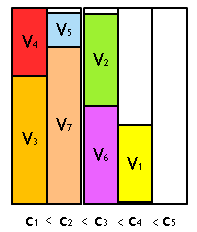
From the crossover and mutation procedure, we can generate next generation. Here is overall procedure of Genetic Algorithm.

1. Generate initial chromosome as a random sequence.
2. Perform crossover and mutation.
3. Repeat procedure 2 until we proceed 10000 generation.
4. Get the best sequence on the overall procedure.
5. Heuristic

Finally, we can introduce new heuristic algorithm. This algorithm is very similar to best-fit algorithm of 1D bin packing problem. Here is the procedure.

1. Sort the job sequence as decrease order.
2. Perform best-fit algorithm. In other words, assign the job into left most interval.
3. Sort the overall assignment as decrease order
4. Assign costs as increase order
5. Compute the cost

Then our remaining job is to analyse this algorithm. First of all, this algorithm takes O( n log n + nm ) which is not that good. However, the number of jobs and schedule is not that much in reality. Therefore, we don’t need the better performance. It is enough.

Secondly, we need to analyse the upper bound of this algorithm. Our claim is that

(Our Cost) ((22/9) \* (max.cost / min.cost)+1) \* (Optimal Cost)

To prove this, we need following lemma.

Lemma. Let be the jobs assigned to interval i. Then the difference between our cost and optimal cost is bounded by 2 \* (our number of bins) \* (Min() \* (maximum cost).

Proof) For interval , let . Be (the difference between our assignment and optimal assignment). Then because the total amount of assignment is same.   
In our algorithm, we need new interval whenever we don’t push it into any other current interval. Consider the minimum assignment. Then (Capa - ) have to be less than minimum assignment. Therefore, consider the interval i such is bigger than . Then for such i, . Therefore, -> |. So we get . Also, for this interval, . Because , we have interval, and = . Therefore, Therefore, = . Therefore, by multiplying maximum cost, we can get upper bound.

With this lemma, we can prove our claim

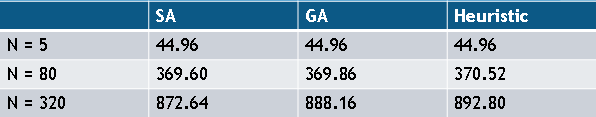
Theorem. (Our Cost) ((22/9) \* (max.cost / min.cost)+1) \* (Optimal Cost)

Proof) because of our lemma,   
(Our Cost) – (Optimal Cost) 2 \* (our bin) \* Min( \* (maximum cost).   
Trivially, (Optimal Cost) (Capacity) \* (Optimal number of bin) \* (minimum cost).   
Therefore, .   
So finally, by multiplying above two inequality, we get  
which gives above theorem.

In reality, the difference between maximum cost and minimum cost are not that much. For korea, max. cost is about 3 times more than min. cost. Therefore, the coefficient is about 75/9.

1. Result

We measure the difference between our methods. Here is our solution.



The result of Simulated Annealing is the best. However, the running time of SA is pretty slow than our heuristic. That is trade off. Also, the result of our heuristic is very good. So, it is very successful.

1. Conclusion and Discussion

We can use the approximation algorithm for solving NP Complete problems. Also, heuristic algorithm is also efficient in terms of performance, even in the result.

Still, we have another job. First of all, the sequence model is not good. It has many duplication, so performance of approximation algorithm is not that good. Also, the bound of our heuristic is NOT that tight. So we need more analysis about this problem. Finally, we need much constraint to realize this problem. However, it will be very hard when we add several constraints. Therefore, we’ll analyse and research about harder problem.