DYNAMIC PROGRAMMING FOR POWER GENERATION MANAGEMENT

JONATHAN CELAYA, MARGARET GACHERU AND JULIO LEDESMA

Contents

1. Introduction

When operating a power system, power generation management is an important challenge faced by the energy sector. Since the energy demand varies from hour to hour, it is desirable to determine when to turn the generators on and off in order to prevent financial losses or unmet demand. The main goal of power generation management is to operate the system in an economically beneficial manner, while remaining within the systems limitations. In a power system, each unit might have its own unique set of physical limitations. Each generator has a minimum number of hours it must be turned on or off, maximum capacity, maximum number of start-ups on a monthly and yearly basis, startup fuel consumption costs, and maximum change in production from hour to hour. In addition, each unit could be considered as a power generator, but one may have dependency on others. The unit commitment problem is used to generate an optimal schedule of turning generators on and off within a system that is subject to the aforementioned constraints. The optimal schedule is the solution of a complex nonlinear and non-convex constrained optimization problem, which is difficult to solve due to the complexity of the limits. In this paper, we explore various methods, such as dynamic programming, that have the capability to produce algorithms that efficiently solve the yearlong unit commitment problem as accurately as possible.

The paper proceeds as follows: we first produce a summary of power generation management and its motivations, then conduct a literature review on methods that have been previously used to approximate optimal unit commitment but which we do not consider. After reviewing these other methods, we introduce methods similar to those which we do consider. We then provide a mathematical introduction to the unit commitment problem as a binary integer program, and follow with a section on the methods we have so far used to attempt to solve the problem. We discuss the main method which we have implemented and illustrate that it is unlikely to provide a full solution to the problem. Finally, we discuss future directions which we will consider in order to overcome the shortcomings of our current results, and present a potential plan of action for solving the generalized version of the unit commitment problem by April 2018.

2. Summary

Management of power generation systems is of importance in multiple fields. With the integration of renewable energies and the volatility of these energy sources, power grid optimization is a hot topic in optimization. Often, the optimization of an electrical power grid is computed in order to minimize risk of blackout in urban areas. A similar motivation for studying power generation planning problems is ensuring that various communities receive sufficient energy to power their homes and businesses throughout the course of a day, month or even year. This motivation requires that the researcher minimizes the total cost for which all demands in a power grid are met. Under this paradigm of minimizing cost, we also often have the noteworthy side effect of minimizing environmental impact, since the minimum cost generally correlates well with the minimum resources expended. Additionally, businesses in the energy industry have a strong interest in optimizing power generation scheduling, as the minimum cost problem can be reformulated into a maximum profit objective with some reasonable modification. It is this last concept which primarily provides the motivation for this project: we consider the problem of producing the greatest profit from a set of generators over

some given period of time, in our case a year. Given that at every hour we have different power prices and costs, and many constraints, the problem of optimally scheduling generators for any reason is difficult therefore, most methods for solving the problem today are heuristic. Furthermore, these problems are closely related, so the development of an algorithm which can provide faster and/or more optimal solutions to the maximum profit problem is useful to not only the business sector, but for safety and meeting power demand as well.

3. **Literature Review**

3.1. **Problem Background.**

The unit commitment problem is used to generate an optimal schedule of generators within a power system that is subject to various constraints. The process identifies when the units should be on or off, remaining within the operational limitations (minimum consecutive run/down hours, maximum start-ups, etc.) and at the same time, minimizing the operational costs. With an optimal solution, it is possible for power companies to supply power with reduced losses and minimum fuel consumption in order to maximize the profit, an important economic consequence [9]. Furthermore, minimal fuel consumption and the efficient use of energy is beneficial to the environment as well as the power companies. As Padhy notes in [7], fuel cost is a major cost component therefore reducing the fuel cost by little as 0.5% can result in savings of millions of dollars per year for large utilities. In addition, it is important to develop a reliable and efficient algorithm due to uncertainty in the power market, such as the integration of wind power and price responsive demand, and emergency situations, such as generation outages and load forecast error [1].

3.2. **Past Approaches.**

Researchers and employers in the power industry have employed various methods to solve the unit commitment problems. One of the well-known approaches to solving the unit commitment problems is the Lagrangian method. In this case, the problem is formulated in terms of a cost function (sum of terms involving each unit), a set of constraints for one unit, and a set of coupling constraints for all units (the generation and reserve requirements). Then Lagrangian relaxation methods are used where the relaxation temporarily ignores the coupling constraints and solves the formulation. Based on dual theory, a separable problem is generated by including the coupling constraints into the objective function. The dual objective function will contain the penalty factors, known as Lagrangian multipliers, which are functions of the constraint violations. For fixed values of the multipliers, the Langragian function is separated by unit so the problem converts to minimization of N (number of units) sub-problems. Now, the procedure solves the dual by maximizing the Lagrangian function and at the same time, minimizing with respect to the unit commitment control variable [3].

Kazarlis, Barkirtzis, and Petridis discuss the utilization of a genetic algorithm to solve the unit commitment problem in [5]. The basic principle of this algorithm is maintenance of a set of solutions to the original problem that evolves over time. The first step is to randomly create a solution space with M initial genotypes with H binary entries then the fitness of each of genotype is evaluated, considering the cost of a particular solution and if a solution violates

the specified constraints. Consequently, a new offspring genotype is created by selecting two parent genotypes by applying a probability proportional to each genotypes fitness. The procedure is repeated until M new genotypes are generated and the new solution space replaces the parent. In some cases, using a simple genetic algorithm results in a failure to converge to the optimal solution. To address this issue, the authors suggest using the varying quality function method and adding problem specific operators to obtain a sufficient solution [5].

Cheng, Liu, and Liu present an application of a combined the Genetic Algorithms and Lagrangian Relaxation method for this problem in [2]. The proposed Lagrangian Relaxation and Genetic Algorithms (LRGA) incorporates Genetic Algorithms into Lagrangian Relaxation method to update the Lagrangian multipliers and improve the performance of Lagrangian Relaxation method. The LRGA method consists of a two-stage cycle. The first stage is to search for the constrained minimum of Lagrangian function under constant Lagrangian multipliers by two-state dynamic programming. The second stage is to maximize the Lagrangian function with respect to the Lagrangian multipliers adjusted by Genetic Algorithms. An advantage of using genetic algorithm is that it searches for the optimal genotypes in parallel through reproduction, crossover, and mutation. Additionally, the advantage of using normalized Lagrangian multipliers is that the number of genotypes is only dependent on number of hours, not number of units [2].

3.3. **Approaches Considered.**

While the above heuristics are very useful in providing a good solution with reasonable computation time, we develop a dynamic programming algorithm which finds a solution with decreased computation time. In order to do this we must make the trade-off of computation time for optimality, as dynamic programming (DP) is a method which inherently suffers from the curse of dimensionality. We attempt to alleviate this computational burden by utilizing a method similar to the approach proposed by Pang and Chen in [8]. Their method reduces the size of the feasible set of solutions by constructing a priority list of all units in a system based on average incremental production cost. On the other hand, we consider the problem of reducing the size of the feasibility region by implementing a variation of the branch and bound technique which Egan, Dylan and Morsztyn apply to the generator maintenance scheduling problem in [4]. Specifically, while Egan et al. use branch and bound in the traditional sense by computing the value of solutions to a binary scheduling problem of increasing length and eliminating branches where the value is less optimal than at another branch, we branch on the binary schedules, check for feasibility and eliminate branches once an infeasible set is found. In both of these papers, once a smaller feasible set is constructed, the computation time greatly decreases.

Given this decrease in computation time, dynamic programming becomes more tractable as a solution approach. Pang and Chen employ an algorithm known as truncated dynamic programming in order to take advantage of this smaller solution set. This algorithm computes least cumulative cost solutions for the remaining feasible solutions at each hour, and at the last hour computes the minimal cost before backtracking to find which solution produced that cost. Through the incorporation of various DP heuristics, Ouyang and Shahidehpour modify the algorithm by Pang and Chen to reduce the computation time while preserving the quality of the solution [6]. They save time by modifying various parameters in these heuristics, and

provide case studies illustrating the reasonability of the varying parameters.

Concerning our work, we must note that in [4, 6, 8] the underlying goal is to minimize cost and satisfy load demand, whereas we wish to maximize profit. As we do not have the requirement of satisfying any particular demand, we have that our feasible solution set is larger than the class of problems previously considered, and thus our results will be slightly more general than those found in the literature we have reviewed.

4. **Problem Statement**

For this project, the unit commitment problem will be broken down into two steps. The first step involves creating the profit maximization function and processing the historical price data. In order to determine the profit of each generator, two types of data are needed: hourly power prices and daily gas prices. Electric transmission companies publish hourly and daily locational marginal prices for each month. Therefore, the goal of the first step is to parse these datasets and extract the relevant information. The second step involves determining a set that satisfies the generator constraints. As the number of hours increase, the number of possible combinations exponentially increases. Therefore, it would be beneficial to dwindle down the possible solutions through the generator constraints.

Definition:
Let a set of m vectors

$$(x_{11}, x_{21}, \ldots, x_{n1}), (x_{12}, x_{22}, \ldots, x_{n2}), \ldots, (x_{1m}, x_{2m}, \ldots, x_{nm})$$

be given. For each vector, x_{ij}, is a binary entry, representing the condition of the generator j at time i.

$$x_{ij} = \begin{cases} 1, & \text{generator j is turned on at hour i} \\ 0, & \text{generator j is turned off at hour i} \end{cases}$$

As stated above, the objective of the unit commitment problem is to maximize the amount of profit while determining the optimal scheduling on generators. For simplicity, to start off, the unit commitment problem will be solved for one generator. Otherwise, working with multiple generators would create the problem of interdependence since a system of generators would have certain coupling constraints.

Assuming the system has only one generator, the objective function of the unit commitment problem has the following components:

$$max\{(p_i - g_i * h - v_i) * c * x_{i1}, 0)\}$$

(1) p_i - Power prices for hour i=1, . . . , n
(2) h - Heat rate
(3) g_i - Gas prices for hour i=1, . . . , n
(4) v_i - Variable operation costs (VOM) for i=1, . . . , n
(5) c - Capacity for the specific generator
(6) x_{i1} - Condition of a generator at time i=1, . . . , n

Furthermore, the constraints which must be satisfied in the optimization formulation (simplified case) include:

(1) Unit minimum down times: once a generator is turned off, it has to cool off for x hours before it is turned back on
(2) Unit minimum up times: once a generator is turned on, it has to run for x hours
(3) Maximum number of starts

The solution should be an n-dimensional vector with 0-1 entries, representing when the generator is turned on or off within the specified number of hours.

$$X^* = (0, 0, 1, 1, \ldots, 0, 0, 1, \ldots)^T$$

5. Methods

5.1. **Branch and Bound.** The first instance of the generator scheduling problem was to create a schedule for a 24-hour period and satisfy the constraints for maximum generator starts ups and minimum on and off hours. The first instinct was to use brute force to formulate a solution by generating every possible vector with n binary entries and determine which maximized the profit. However, due to the exponential number of possible schedules, a brute force solution would not be feasible in solving the year long instance of this problem. Therefore, a more nuanced method that could reduce the size of the feasible set was necessary. To solve the day long instance, a branch and bound algorithm was used to generate all schedules that satisfy the constraints. By pruning branches that do not satisfy the constraints, the number subtrees that will ever be generated was reduced.

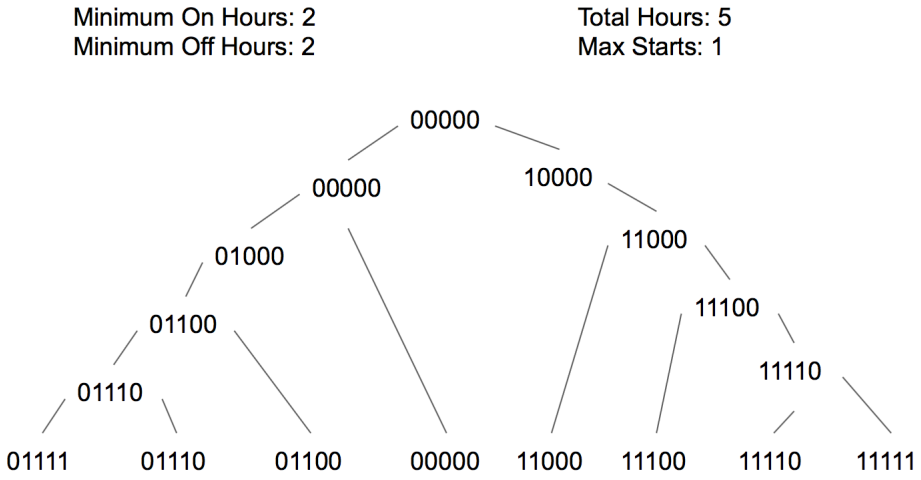
The algorithm for generating potential schedules is presented as follows:

Algorithm 1: GenerateBranches(schedule)

```
if index = maxIndex or maxStarts then
    add schedule to solutionSet ;
else if minOnHours = FALSE then
    GenerateBranches(schedule add one)
else if minOffHours = FALSE then
    GenerateBranches(schedule add zero)
else
    GenerateBranches(schedule add one) ;
    GenerateBranches(schedule add zero) ;
```

Generate Branches is recursive in structure. The input schedule is an object with various methods to update and verify information about the schedule's current state. With each recursive call, a copy of the schedule object with the appropriate modifications is passed as the input. As indicated by the base case, we add schedules only when we have reached the last index or if we have met the maximum start ups. If the base case conditions are not met, we check to see if the other constraints have been violated. In particular we check to see if a schedule had been on, that it will remain on until the minimum on hours constraint has been satisfied. Likewise, if the generator had been on and the minimum off hours constraint had not been met, then we recurse down this branch. If both constraints have been met, we have no choice but to continue recursing down both branches of the tree. Figure 1 represents the process of pruning a tree.

FIGURE 1. Generate Branches Visualization



A key aspect in our methods design is separating the concerns for the problem. This algorithm simply creates a solution set with candidate schedule. More work needs to go into building the program to parse prices from CSV files. Generate Branches populates the global set of candidate solutions, and does nothing else. Once this solution set is built, the set is passed as an input to an argument maximizing function that will compute the profit produced by each schedule, and will return the value of the maximum profit as well as its corresponding schedule. While our branch and bound method successfully generated feasible schedules, in our results section we will illustrate why this method is not sufficient in solving the yearlong instance of the Generator Scheduling Problem.

5.2. **Dynamic Programming.** After determining that the branch and bound result is too computationally expensive to reasonably extend it to the year-long problem, we developed a dynamic programming algorithm to solve the power generator scheduling problem.

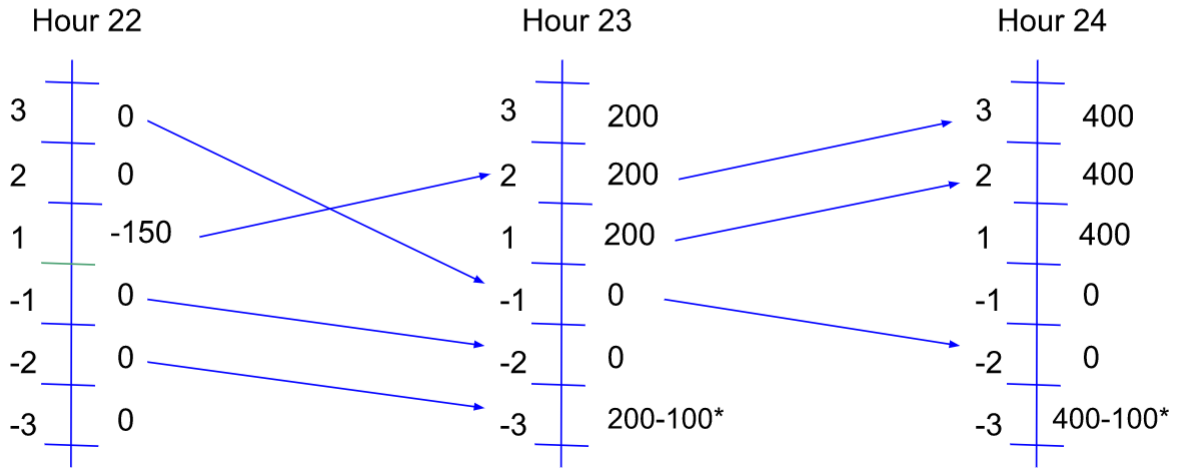
In dynamic programming approaches, one minimizes runtime by performing computations which have results that can be used in future computations. This is due to the nature of dynamic programming, as it is capable of converting problems which are generally exponential into polynomial time. Dynamic programming approaches have been studied before, as in [8] and [6]. However, we believe it will be beneficial to utilize dynamic programming in a way that we solve our problem in reverse to standard practice. Specifically, our new approach will not initially consider the first hour in a given time frame; rather, it will first consider the last hour which we are scheduling generators for, and determine which possible set of prior hours led to the most profit in that hour. In the same way, our algorithm will consider the entire set of hours decrementing towards the first chronological hour and will ultimately produce a near optimal solution in good time. Figure 2 demonstrates the first step of this new method for the 24 hour unit commitment problem given start-up costs and minimum on and off hours. We omit the max starts constraints in our example for visual simplicity.

In particular, we have designed and implemented a dynamic programming algorithm which takes as input the minimum on and minimum off hours, a time schedule, and the maximum

FIGURE 2. Dynamic Programming Schedule Visualization

Hour	22	23	24
ON Revenue	-150	200	400

Min off hours: 3
Min on hours: 2
* Start up cost: $100



starts allowed in that time schedule. The algorithm uses the constraints to create a set of states, where each state is a vector of (previous hours on/off, previous total starts, generator is on/off, profit). Here, the profit is aggregated as a sum of the profit at the state in the chronologically previous hour and current hour. Once this algorithm is finished, it computes a set of possible profits based on decisions that can be made in the chronologically first hour. We then use an argmax function to take the maximum profit, and also a basic trace back algorithm which finds the decisions we made in the DP portion in order to achieve that maximum profit. A summary of the algorithm is provided below:

Algorithm 2: DP Algorithm Pseudocode

1 **Input:** Number of hours (m), number of generator on-off states (n), power prices (p_i), constraints
2 **Output:** A locally optimal generator schedule
3 **Initialize:** Calculate profit at hour n for each state and set $j = n - 1$
4 **Generate cumulative profits** for each state at hour j while $j > 0$, using profits at hour $j + 1$ and prices p_j
5 **Determine maximum profit** from values at hour 1
6 **Find optimal schedule** by traceback starting at the state at hour 1 with maximum profit

6. **Results**

Recall that our branch and bound algorithm generates a set of schedules that are potential solutions to the generator scheduling problem. We implemented this algorithm on a MacBook Pro with a 2.8 GHz Intel Core i7 processor. Furthermore, the constraints we used for this

version of Generate Branches were minimum on and off hours, and maximum startups. Table 1 shows the runtime results of generating schedules for a few instances of the 24-hour problem

TABLE 1. Dynamic Programming with Various Durations

Min On Hours	**Min Off Hours**	**Max Starts**	**Total Hours**	**Max Time(seconds)**
6	4	1	24	0.1
7	3	1	24	0.1
5	3	2	24	0.1
15	10	10	8760	0.5
150	50	5	8760	2.1
220	70	100	100,000	52.2

TABLE 2. 24 Hour Branch and Bound Scheduling Running Time

Min On Hours	**Min Off Hours**	**Max Starts**	**Max Time(seconds)**
6	4	1	1.5
7	3	1	0.8
5	3	2	2.3
14	5	1	0.8

Even though there were 16,777,216 possible schedules that could have been generated for the 24-hour period, the branch and bound algorithm completed within a few seconds.In addition, when studying this approach we discovered that adding constraints led to more pruning on the trees. However, while this solution worked for 24-hour period, as we add more hours the running time rapidly increases. In particular, running the algorithm for the yearlong instance would not finish. This indicated that while the solution looked promising for the day long scheduling problem, we needed to identify another method for computing the optimal schedule for a longer scheduling time frame.

In order to significantly decrease runtime, we opted to implement the dynamic programming heuristic and sacrifice a guaranteed optimal result. However, by the nature of dynamic programming, we were able to ensure a local optimum, which is sufficient for most practical purposes at companies such as BP. We found that this algorithm runs on $O(n^2)$ when we use the min on and off hours and max starts constraints. Mathematically, the runtime stems from iterating through two two-dimensional matrices one time each. It is a notable result that we were able to determine an algorithm in this runtime, as we first expected that the max starts constraint would require expanding the two-dimensional matrix containing the information from only the min on and off hours into a three-dimensional matrix, thus running in $O(n^3)$. Fortunately, upon implementing the third constraint in the computer, we found that we could store the information in parallel via a dictionary.

Finally, in the above tables we see just how efficient the Dynamic Programming algorithm is: we can solve our 24-hour problem with the min on and off hours and max starts constraints

in about one tenth of a second on the same MacBook Pro with a 2.8 GHz Intel Core i7 processor that it took approximately one to two full seconds to solve using the branch and bound algorithm. More importantly, the Dynamic Programming approach is tractable to solve the year-long problem, unlike the branch and bound technique. Indeed, with only the constraints we have currently implemented, we can solve the 100,000 hour problem, which is over 11 years, in under one minute. This will give us some flexibility when adding additional constraints.

7. **Future Work**

We met our intended goal of producing an efficient algorithm for scheduling a power generator for a year subject to three real-world constraints. If this project were to continue, adding more real-world constraints to the dynamic programming algorithm would be a starting point. Based on our previous experience with adding constraints, it is possible that each constraint would merely require an additional dimension in our state matrices, or like in in the case of our maximum-starts constraint algorithm, would use additional data structures. Some constraints, such as the capacity of a power generator, would simply require appropriate additional conditionals in our current solution.

Another next step would be to produce schedules of a system of generators, as opposed to just one. For this system, each generator would need to adhere to each constraint discussed in our solution, as well as any additional constraints that arise as a result of this new system. Among these would be meeting demand, and allocating which portion of the demand is to be met by each generator.

8. **Conclusion**

In this report, we have discussed a variation of the unit commitment problem in which the goal is to optimally schedule electric power generators in order to maximize profit for companies such as BP.This problem is fairly well-studied, and we examined some methods which have been used in the literature to help determine what approaches we should take. Ultimately, we first attempted to employ a branch and bound type technique. However, using branch and bound to solve the day long instance of the scheduling problem successfully highlighted the growing exponential complexity that calls for a more efficient method. We then attempted to combat this problem by turning to a heuristic method - Dynamic Programming. We found that implementing a dynamic programming approach produced efficient results for a range of constraint values and schedule durations. While this problem suffers from the curse of dimensionality, adding dynamic programming leaves a relatively simple approach to adding new constraints that produce a more realistic schedule. Thus, the algorithm we have developed appears to be tractable as a fast, heuristic solution to the one generator power generator scheduling problem that is relatively easy to implement and modify based on the exact physical constraints of the generators.

References

[1] D. Bertsimas, E. Litvinov, X. A. Sun, J. Zhao, and T. Zheng. Adaptive robust optimization for the security constrained unit commitment problem. *IEEE Transactions on Power Systems*, 28(1):52–63, Feb 2013.

[2] Chuan-Ping Cheng, Chih-Wen Liu, and Chun-Chang Liu. Unit commitment by lagrangian relaxation and genetic algorithms. *IEEE Transactions on Power Systems*, 15(2):707–714, May 2000.

[3] A. I. Cohen and S. H. Wan. A method for solving the fuel constrained unit commitment problem. *IEEE Transactions on Power Systems*, 2(3):608–614, Aug 1987.

[4] Gerard T Egan, Tharam S Dillon, and Karol Morsztyn. An experimental method of determination of optimal maintenance schedules in power systems using the branch-and-bound technique. *IEEE Transactions on Systems, Man, and Cybernetics*, 6(8):538–547, 1976.

[5] S. A. Kazarlis, A. G. Bakirtzis, and V. Petridis. A genetic algorithm solution to the unit commitment problem. *IEEE Transactions on Power Systems*, 11(1):83–92, Feb 1996.

[6] Z Ouyang and SM Shahidehpour. An intelligent dynamic programming for unit commitment application. *IEEE Transactions on Power Systems*, 6(3):1203–1209, 1991.

[7] N. P. Padhy. Unit commitment-a bibliographical survey. *IEEE Transactions on Power Systems*, 19(2):1196–1205, May 2004.

[8] CK Pang and HC Chen. Optimal short-term thermal unit commitment. *IEEE Transactions on Power Apparatus and Systems*, 95(4):1336–1346, 1976.

[9] B Saravanan, Siddharth Das, Surbhi Sikri, and DP Kothari. A solution to the unit commitment problem–a review. *Frontiers in Energy*, 7(2):223, 2013.