# Using High Throughput Computing for solving a Stochastic Unit Commitment Problem

Julio C. Góez
Department of Industrial Engineering
Lehigh University

October 17, 2011

**Abstract**

In this paper we explore the use of High Thorughput Computing for solving a Stochastic Unit Commitment Problem. This problem is to find the production scheduling of a set of electric power generating units, and the generation levels for each unit, over a short term, typically from 24 hours to one week. The objective is to find the optimal schedule which meets the energy demand forecast at a minimum cost, and satisfy technological constraints such as the minimum up and down times for the units. The presence of integer variables and the dimension of the problem makes it difficult to solve. Here we review a technique that allows to break the big problem into small problems that can be solve independently. Then we present a plan to perform some experiments to analyze how using high throughput computing can help to improve the solution times.

## 1 Introduction

The Unit commitment problem was first proposed in 1966 by Kerr et al. in [7], and since then the basic definition of the problem has remain almost the same. Reviewing the definitions given in [11, 10, 6, 12, 13, 8, 9], we can see that the general definition of the unit commitment problem can be stated as finding the optimal scheduling of production of electric power generating units over a short term, typically from 24 hours to one week, in order to minimize the operations costs. This optimal solution must obey the operating constraints and must satisfy the demand forecast.

One thing that makes this problem interesting is that usually a typical electrical system has a variety of units available for generating electricity, and each has its own characteristics [1]. For example, a nuclear power unit can provide electricity at a very low incremental cost for each additional megawatt/hour of energy, but it has both a high cost for starting up again once it has been shut down and it takes a while to bring it back up to full power. At the other extreme, a gas turbine generator can be started up in a few minutes. However, its incremental cost per megawatt/hour is much more expensive.

Stochastic Programming have been used for modeling the Unit Commitment problem to take into account the random nature of the demands. In this technique several scenarios are considered in the formulation and some know probability distribution is used for optimizing the expected value of the cost over all the scenarios. The first model that tries to take account of the randomness in the demands was the models presented by Muckstadt and Wilson on 1968 in [9]. The Stochastic Programming for the unit commitment problem was first presented by Takriti et al. on 1996 in [14]. Another example in this direction is the paper by Carøe, and Schultz [2]. Finally, with the recent developments of the energy markets some models have started to consider the randomness in the energy prices. In this models the new objective is maximize the profit. An example of this new trend is given in the work of Valenzuela and Mazumdar in [16].

In this paper we focus on a Stochastic Unit Commitment model. In particular, we study some techniques used for decomposing the problem. This result in series of problems that can be solved independently in each iteration of the method. This motivated the study of the potential of using high throughput computing techniques for solving these independent problems distributing them among available resources.

## 2 Stochastic Unit Commitment Mathematical Model

Before presenting the stochastic model of the Unit Commitment Problem we need to introduce the deterministic model first. A basic deterministic mathematical model can be the following,

$$\text{Minimize: } \sum_{i=1}^{I}\sum_{t=1}^{T} z_{i,t}F_{i,t} + \sum_{i=1}^{I}\sum_{t=1}^{T} g_{i,t}C_{i,t} + \sum_{i=1}^{I}\sum_{t=1}^{T} y_{i,t}U_{i,t} \tag{1}$$

$$\text{Subject to: } \sum_{i}^{I} g_{i,t} \geq D_t,\ t = 1,\ldots,T, \tag{2}$$

$$g_{i,t} \leq Q_i z_{i,t},\ t = 1,\ldots,T, i = 1,\ldots,I \tag{3}$$

$$g_{i,t} \geq q_i z_{i,t},\ t = 1,\ldots,T, i = 1,\ldots,I \tag{4}$$

$$y_{i,t} \geq z_{i,t} - z_{i,t-1},\ t = 2,\ldots,T, i = 1,\ldots,I \tag{5}$$

$$z_{i,t} - z_{i,t-1} \leq z_{i,\tau}, \tau = t,\ldots,\min\{t + L_i - 1, T\}, t = 2,\ldots,T, \tag{6}$$

$$z_{i,t-1} - z_{i,t} \leq 1 - z_{i,\tau}, \tau = t,\ldots,\min\{t + l_i - 1, T\}, t = 2,\ldots,T, \tag{7}$$

$$\sum_{i=1}^{I} Q_i z_{i,t} \geq D_t + r_t,\ t = 1,\ldots,T \tag{8}$$

$$z_{i,t}, y_{i,t} \in \{0,1\} \tag{9}$$

$$g_{i,t} \geq 0, \tag{10}$$

where we have the following definitions:

- $g_{i,t}$ is a continues variable that represents the MV of energy produced by generator $i$ in period $t$,
- $y_{i,t}$ is a binary variable that is 1 if generator $i$ is started at the beginning of period $t$, 0 otherwise,
- $z_{i,t}$ is a binary variable that is 1 if generator $i$ is on during period $t$, 0 otherwise,
- $F_{i,t}$ is the fixed cost in \$/period of operating generator $i$ in period $t$,
- $C_{i,t}$ is the cost of generation for generator $i$ in period $t$ in \$/MV/period,
- $U_{i,t}$ is the cost of start up for generator $i$ in period $t$ in \$,
- $Q_i$ is the upper bound in MV for energy generated with generator $i$ in every period,
- $q_i$ is the lower bound in MV for energy generated with generator $i$ in every period,
- $L_i$ is the minimum up time for generator $i$ when it is started up,
- $l_i$ is the minimum down time for generator $i$ when it is shutted down,
- $r_t$ the MV in spinning reserve that must be available in period $t$,
- $1,\ldots,I$ is the set of generators,
- and $1,\ldots,T$ is the set of time periods.

In this model we have the basic Unit Commitment problem. Then, given a set $\{i = 1,\ldots,I\}$ of generators which can be of different types, and a set of time periods $\{t = 1,\ldots,T\}$, the objective function (1) is minimizing the total operative costs. That is, the sum of the fixed operating cost $F_{i,t}$, the generating costs $C_{i,t}$, and the start up costs $S_{i,t}$, for all the generators in all time slots. In constraint (2) we have that the energy produced by all generators must meet or exceed the demand $D_t$ in each period. Now, in constraints (3) and (4) we have that the generators must operate within their minimum an maximum generations bounds. Then, equation (5) is controlling when the generators are started up. In equations (6) and (7) we have that the generators must accomplish the minimum up and down times. Finally, constraint (8) refers to the additional generating capacity that must be available in short time in the system to meet demand in case a generator fails, this is call the spinning reserve. This basic model has been extended during the past five decades to consider different issues related with the operation of energy generation utilities. Then, we can find different types of new constraints and different types of objective functions depending on what kind of new consideration is being accounted with the model, see e.g [3, 4, 17].

Takriti et al. formally presented the first stochastic model in [14]. However, Muckstadt and Wilson presented the first model that incorporate randomness in the energy demands [9]. Here, we focus on the stochastic model technique applied to the Unit Commitment problem by Takritri et al. [14], and Takitri and Birge [15]. In that model the authors use the multiscenarion approach for modeling the uncertainty about the future demand. Then, they create a set of deterministic models $S$, each one of them using a particular demand vector $D^s$ that gives limited information about the demand uncertainty. Additionally, they assign a weight, $P_s$, that reflects the probability of this occurrence in the future. Hence, the objective functions will minimize the expected value of the operations cost over all possible scenarios. As a result, using the basic model in (1)-(10), a stochastic model for the Unit Commitment problem would be,

$$\text{Minimize: } \sum_{s \in S} P_s \left( \sum_{i=1}^{I} \sum_{t=1}^{T} z^s_{i,t} F_{i,t} + \sum_{i=1}^{I} \sum_{t=1}^{T} g^s_{i,t} C_{i,t} + \sum_{i=1}^{I} \sum_{t=1}^{T} y^s_{i,t} U_{i,t} \right) \tag{11}$$

$$\text{Subject to: } \sum_{i}^{I} g^s_{i,t} \geq D^s_t,\ t = 1, \ldots, T, \tag{12}$$

$$g^s_{i,t} \leq Q_i z^s_{i,t},\ t = 1, \ldots, T, i = 1, \ldots, I \tag{13}$$

$$g^s_{i,t} \geq q_i z^s_{i,t},\ t = 1, \ldots, T, i = 1, \ldots, I \tag{14}$$

$$y^s_{i,t} \geq z^s_{i,t} - z^s_{i,t-1},\ t = 2, \ldots, T, i = 1, \ldots, I \tag{15}$$

$$z^s_{i,t} - z^s_{i,t-1} \leq z^s_{i,\tau}, \tau = t, \ldots, \min\{t + L_i - 1, T\}, t = 2, \ldots, T, \tag{16}$$

$$z^s_{i,t-1} - z^s_{i,t} \leq 1 - z^s_{i,\tau}, \tau = t, \ldots, \min\{t + l_i - 1, T\}, t = 2, \ldots, T, \tag{17}$$

$$\sum_{i=1}^{I} Q_i z^s_{i,t} \geq D^s_t + r_t,\ t = 1, \ldots, T \tag{18}$$

$$z^s_{i,t}, y^s_{i,t} \in \{0, 1\} \tag{19}$$

$$g^s_{i,t} \geq 0, \tag{20}$$

where constraints (11) - (20) must be satisfied for all possible scenarios $s \in S$, and the superscript $s$ identifies to which scenario belongs each variable. Additionally, for consistency in the solution they add a set of constraints called Bundle constraints. That is, if two scenarios $s$ and $s'$ are indistinguishable up to time $t$, then the decisions for both scenarios by time $t$ must be the same. Then, a Bundle $\Omega_k$ is a set of scenarios that at time $\tau_k$ have the same demands for every period in $1, \ldots, \tau_k - 1$. Now, they denote the Bundle of $s$ at time $t$ by $B(s, t)$. Hence, if scenario $s$ belongs to bundle $\Omega_k$ at time $\tau_k$ we say that $B(s, \tau_k) = \Omega_k$. As a result, if two scenarios $s_1, s_2$ are members of the same bundle at time $\tau_k$, then we have

$$B(s_1, \tau_k) = B(s_2, \tau_k) = \Omega_k \Rightarrow B(s_1, t) = B(s_2, t), t = 1, \ldots, \tau_k - 1.$$

Hence, they add the following constraint to the problem,

$$B(s_1, \tau_k) = B(s_2, \tau_k) = \Omega_k \Rightarrow z^{s_1}_{i,t} = z^{s_2}_{i,t} = c^k_{i,t}, i = 1, \ldots, I, t = 1, \ldots, \tau_k - 1,$$

where $c^k_{i,t}$ is the common decision for all units $i$ up to time $\tau_k - 1$ for all the scenarios that are in the bundle $\Omega_k$.

# 3 Solution strategies

As it can be seen from the model in (11) – (20) the Stochastic Unit Commitment Problem could be solved in principle as a one big optimization problem. However, the size of the problem combined with the presence of integer variables make this problem particularly difficult to solve. Indeed, depending on the data a cutting edge optimization solver like CPLEX 12.0 can take more that 24 hours for solving an instance of this problem. This time is clearly not acceptable when a company need to be reviewing its plan every day. For that reason, big efforts have been done in designing heuristics and decomposition methods to tackle this difficulty. In 2008 Goez et al. analyzed some of this techniques in [5]. The main characteristic of these methods is the possibility of breaking the problem in (11) – (20) in a series of independent problems. Then, one can distribute these problems in different resources such that can be solved simultaneously. Hence, the objective of this project is to analyze the impact in the solution times for the methods described in [5]. In this section we will briefly described the methods and then we will talk about the strategy to apply HTC in this methods.

## 3.1 Decomposition Methods

For solving the stochastic unit commitment problem Goez et al. [5] tested three different methods. First, they use a Lagrangian relaxation where each bundle constraint has a multiplier $\lambda^s_{i,t}$ associated with it. Then, a penalty term $\lambda^s_{i,t}(z^s_{i,t} - z^{s'}_{i,t})$ is added to the objective function, where $s$ and $s'$ are in the same bundle $\Omega_k$. When the bundles constraints are relaxed, the problem is separable in $S$ different problems, which can be solved as independent deterministic unit commitment problems for each scenario. Therefore, one can use any strategy for solving the deterministic unit commitment problem for each scenario with the objective function

$$\text{Minimize: } \sum_{i=1}^{I}\sum_{t=1}^{T} z^s_{i,t}F_{i,t} + \sum_{i=1}^{I}\sum_{t=1}^{T} g^s_{i,t}C_{i,t} + \sum_{i=1}^{I}\sum_{t=1}^{T} y^s_{i,t}U_{i,t} + \sum_{i=1}^{I}\sum_{t=1}^{\tau_k} \lambda^s_{i,t}(z^s_{i,t} - z^{s'}_{i,t}). \tag{21}$$

Once these independent problems are solved the Lagrange multipliers $\lambda^s_{i,t}$ are updated and the problems are solved again. This is repeated until the duality gap for the optimization problem (11) – (20) goes below some arbitrary threshold. A schematic description of this process is presented in Algorithm 1.

**Algorithm 1** Lagrangian Relaxation

```
Require: Initial values for $\lambda^s_{i,t}$, $\forall s, i, t$
  repeat
    Solve problem defined by 21, 2 – 10 for each scenario $s \in S$
    Compute duality gap for (11) – (20)
    Update Lagrangian multipliers $\lambda^s_{i,t}$
  until The duality gap for (11) – (20) is small enough
```

The second technique explored was the heuristic method defined by Takriti and Birge in [15]. This is a similar technique to the Lagrangian, but the bundle constraints are written in the alternative form $z^{s'}_{i,t} = c^k_{i,t}$, where $c^k_{i,t}$ is the value that need to be assigned to the $i$th decision variable $z^{s'}_{i,t}$ of all scenarios $s'$ in the bundle $\Omega_k$ at time $t$. The value of $c^k_{i,t}$ is unknown, then in each iteration its value is updated as

$$c^k_{i,t} = \frac{\sum_{s'} P_{s'} z^{s'}_{i,t}}{\sum_{s'} P_{s'}},$$

and the convergence is achieved when the value of $c^k_{i,t}$ approach to 1 or 0. Once the algorithm converge one needs to approximate the current solution to a feasible solution. Again, in this method the problem is separable in $S$ different problems, which can be solved independently. In this case the objective function takes the form

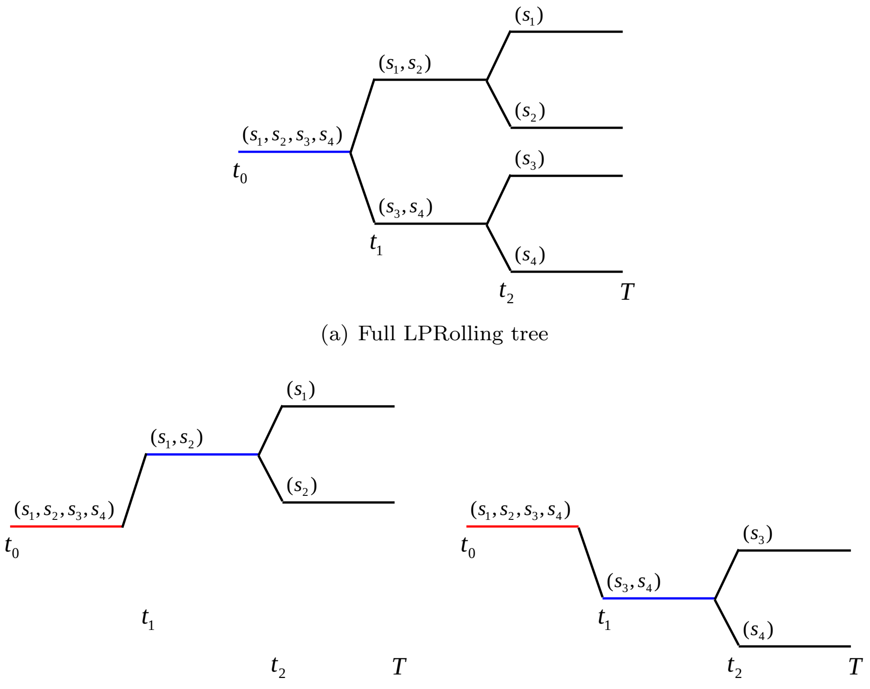
$$\text{Minimize: } \sum_{i=1}^{I}\sum_{t=1}^{T} z^s_{i,t}F_{it} + \sum_{i=1}^{I}\sum_{t=1}^{T} g^s_{i,t}C_{it} + \sum_{i=1}^{I}\sum_{t=1}^{T} y^s_{i,t}U_{it} + \sum_{i=1}^{I}\sum_{t=1}^{\tau_k} \lambda^s_{it}(z^s_{i,t} - c^k_{i,t}).$$

A brief description of this process is described in Algorithm 2.

**Algorithm 2** Takriti–Birge algorithm

```
Require: Initial values for $\lambda^s_{i,t}, c^k_{i,t}$, $\forall s, i, t$
  repeat
    Solve problem defined by 3.1, 2 − 10 for each scenario $s \in S$
    Updated the values for $c^k_{i,t}$
    Update Lagrangian multipliers $\lambda^s_{i,t}$
  until All the values $c^k_{i,t}$ approach 1 or 0
  Approximate the current solution to a feasible solution.
```

Finally they proposed a simple LPRolling heuristic for finding a feasible solution for the stochastic problem. In this strategy on solves the problem progressively, using the bundles structure of the scenarios tree. Then, given the list of bundles in the scenarios tree, the first bundle $\Omega_1$ is that one which its starting time is the time 1. For the first bundle the binary constraints for the up/down variables $z^s_{i,t}$ are kept and the binary constraints for $t \geq \tau_1$ are relaxed. Once that problem is solved, the value of the up/down variables in the first bundle $\Omega_1$ are fixed for all the scenarios, and the procedure is repeated for the next bundles. That is, the binary constraints are kept for the up/down variables in the current bundle $\Omega_k$. Additionally, the values of the up/down variables are fixed in all the scenarios in $\Omega_k$ for the time periods before the starting time of $\Omega_k$. Finally, the binary constraints are relax for the up/down variables $z^s_{i,t}$ for all scenarios in $Omega_k$ and $t \geq \tau_k$ and the new problem is solved. This process is repeated until all the bundles have been visited. In this method the problem is decomposed by bundles and all of them need to be solved in a specific sequence. At the end the solution for the stochastic problem is given by the solutions at the final branches of the scenarios bundles tree. In brief, for finding the solution to (11)–(20) all the problems in the tree have to be exhausted.



(a) Full LPRolling tree

(b) Top branch of the LPRolling tree

(c) Bottom branch of the LPRolling tree

Figure 1: LPRolling tree

### 3.2 Implementation Plan

For this project we plan to test the usage of distributed computed for the three decomposition methods mentioned in Section 3.1.

First, observe that the Lagrangian Decomposition and the Heuristic proposed by Takriti and Birge are two iterative methods. In each iteration one has to solve as many problems as scenarios are available for the problem. These problems can be solve independently, so the main idea is to take advantage of this independence. One can distribute these problems among the different processors available in each iteration. Then, the main goal here is to reduce the time needed to get the solution to these problems if the were ran sequentially. One important observation is that the information coming from the solution of the problems needs to be consolidated to compute the duality gap in the Lagrangian relaxation and to update the parameters $c^k_{i,t}$ in the Takriti and Birge algorithm. Given this observation, we plan to perform a benchmark between a master-worker implementation in HTC and the use of MPI for this two decomposition methods. Basically, as is shown in Algorithms 2 and 1 we need to run a loop until it satisfy a predefined criteria. In each iteration each processor receives the problem data, solves a problem associated with one scenario an return the solution. In the case of the Lagrangian relaxation this information is moved back to a master node to compute the duality gap. On the other hand, in the case of the Takriti and Birge algorithm this imformation is consolidated to update the values of the $c^k_{i,t}$ parameters. Then, the $c^k_{i,t}$ values are sent back to each of the processors to solve the problems again. The current implementation runs the algorithms solving the problems sequentially in a single processor. It needs approximately 50 to 70 iterations for the Lagrangian relaxation and more than 200 iterations for the Takriti and Birge Heuristic. Each iteration takes about 15 minutes to solve all the problems sequentially. Then, in average the solution time for each problem is less than a minute. Hence, we expect a significant improvement in the solution time distributing the problems in different nodes.

The LPRolling heuristic has a different structure. In this algorithm one builds a tree following the scenarios structure as the one shown in Figure 1. Each branch of the tree describes a problem. The main idea is to (11)–(20) start on the root with a big relaxed problem and keep building a solution using the solution to the problem found when a new branch is open. In this process the final step will solve as many problems as scenarios are available. The way it is designed allow us to build the next nodes in the tree once we have the solution for a particular problem. For example, for the first bundle we may start with a single problem. Once this problem is finished the description of the problem for that bundle is updated and we build the list of problems next in the three. This problems can be solved independently and once they are finished their childes can be created independently for each of the problems. In order to recover the solution for the stochastic problem (11) – (20) all the problems in the tree must be solve. This is illustrated in Figure 1. Given the structure of this algorithm we think it fits the requirements to use HTC. In particular, we plan to use DAGMan to automate the creation of the problems in each node of the tree. We expect again to have a significant improvement compare to the current sequential implementation.

### 3.3 Implementation Requirements

Given the decomposition methods presented in Section 3.1 we now describe some consideration for the implementation of the plan to study the usage of HTC or HPC in the solution of the Stochastic Unit Commitment model presented in (11) – (20).

The first consideration about the implementation is the size of the problem. This is strictly related to the number of scenarios to be considered in the model. From the decomposition methods is clear that at any time one will need at most as many processors as scenarios in the problem. Currently the cases we have for the analysis have 32, 72 and 150 scenarios, which make them suitable for the local resources we have at Lehigh University.

A second important consideration is the memory needed for solving the independent problems in each node. In each of these nodes we are solving a problem with integer variables. The process of solving these integer problems can demand a lot of memory. Our experience has shown that for solving each of these problems the availability of at least 2 GB of memory in each node is a safe bet. The solvers available to solve this kind of problems use a branch and bound approach. One option in this solvers is, once it runs out of ram memory, to store the branch and bound nodes on temporary files on disk. This approach has the advantage of avoiding the solution process to fail. However, this option will affect critically the solution time slowing down the process significantly. Hence, the availability of enough memory will help to avoid the use of this last option, which will affect significantly the performance of the implementation.

In the planned implementation we do not need to move big files through the network. So we don't need to worry about the network latency to critically affect the performance of our implementation. In general all the files carrying the problem data and the current solution for each problem are always less than 20 MB.

The final requirement is the availability of a mixed integer linear optimization solver in each of the nodes to be used in the implementation. This are not to big and one could install them where needed without problem. However, the cutting edge solver are all proprietary software. Hence the main constrain in this case is the availability of licenses for running the program. Currently, we use CPLEX 12.0, which is available in Lehigh's local cluster under an academic license.

# 4 Conclusions

The initial analysis shows that the strategies considered for solving the stochastic unit commitment problem described in (11) – (20) may benefit from the use of distributed computing. There are two main approaches that we want to explore. For the Lagrangian Decomposition and the Heuristic proposed by Takriti and Birge we think that an implementation using MPI could be more suitable. For the LPRolling heuristic we think that we can make use of an HTC approach to distribute the work done during the solution process. In particular the possibility to describe the scenarios bundles tree as an acyclic graph give us the motivation to use DAGMan to automate the construction of that tree. We expect significant improvements in the solutions times. Additionally, we expect this improvement to allow us the use of a greater number of scenarios in the analysis of the use of stochastic programming for handling uncertainty in the unit commitment problem.

# 5 The role of the OSG summer school to understand the potential of HTC in my research

Being part of the OSG summer school helped me to have a conceptual understanding of HTC as well as to have some practical experience working with it. These two aspects played a key role to understand the potential of using HTC for the challenge of solving a Stochastic Unit Commitment Problem.

Given my previous experience with the Stochastic Unit Commitment Problem I was aware of the potential of using distributing computing for tackling this problem. Indeed, this was an idea that was discussed previously with some colleagues as something that could help to exploit the decomposition structure discussed in Section 3.1. Up to this point, the option considered in this exploration was constrained to the use of MPI to distribute the problems at hand among the resources available. Given my limited understanding of CONDOR and the lack of knowledge about DAGMAN, the use of HTC was not clear as an option in this exploration. However, this changed with my attendance to the OSG summer school, which gave me an overview of the potential of these tools. For example, during the study of the concepts of DAGMAN it was clear to me how the LPRolling heuristic had a structure that was suitable for the use of HTC. Additionally, further discussions with my mentor Igor Sligoi helped me with the understanding of the potential of HTC in the techniques based in Lagrangian relaxation. In brief, the participation in the OSG summer school opened a broad spectrum of ideas to consider for the solution strategies to solve a Stochastic Unit Commitment Problem.

Equally important was the hands on experience acquired during the school. In my case these hands on exercises served as a bridge between the new knowledge and my optimization problem. This process took place first by getting a taste of the usage of this tools using the concrete code and exercise provided during the sessions. Then, once the concepts where materialized with this examples, having extra time to explore some changes helped me to associate this examples with the problem I had on mind. Most important, the opportunity to have direct access to CONDOR experts to ask questions and experiment with the code gave me the confidence to apply this in my optimization problem. In short, I believe that the practical component of the workshop was essential to understand the potential of HTC in my personal research.

## 6 Acknowledgment

I take the opportunity to thank the organizing committee for the OSG summer school 2011, the organizing committee of TeraGrid11 conference for coordinating all the logistic and the National Science Foundation for the financial support. Special thanks to Igor Sligoi for agreeing to be my mentor, as well as to Alain Roy, Brian Bockelman, and Zach Miller for the productive sessions.

## References

[1] Electrical generation unit commitment planning. Technical report, LINDO Systems, 2003.

[2] C. Carøe and R Schultz. A two-stage stochastic program for unit commitment under uncertainty in a hydro-thermal power system. Web, February 1998.

[3] M. Carrión and J. M. Arroyo. A computationally efficient mixed-integer linear formulation for the thermal unit commitment problem. *IEEE Transactions on Power Systems*, 21(3):1371–1378, August 2006.

[4] Antonio Frangioni and Claudio Gentile. Solving Nonlinear Single-Unit Commitment Problems with Ramping Constraints. *Operations Research*, 54(4):767–775, 2006.

[5] Julio Goez, Jayant Kalagnanam, James Luedtke, and Deepak Rajan. Stochastic unit commitment problem. Technical report, IBM Research, 2008.

[6] B.F. Hobbs, M.H. Rothkopf, R.P. O'Neill, and Hung-po (Eds.) Chao. *The Next Generation of Electric Power Unit Commitment Models*, volume 36 of *International Series in Operations Research & Management Science*. Springer, New york, 2001.

[7] R.H. Kerr, J.L. Scheidt, A.J. Fontanna, and J.K. Wiley. Unit commitment. *IEEE Transactions on Power Apparatus and Systems*, PAS-85(5):417–421, May 1966.

[8] G.S. Lauer, N.R. Sandell, D.P. Bertsekast, and T.A. Posberght. Solution of large-scale optimal unit commitment problems. *IEEE Transactions on Power Apparatus and Systems*, PAS-101:79–86, January 1982.

[9] J.A. Muckstadt and R.C. Wilson. An application of mixed-integer programming duality to scheduling thermal generating systems. *IEEE Transactions on Power Apparatus and Systems*, PAS-87(12):1968–1978, December 1968.

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

[11] Sayeed Salam. Unit commitment solution methods. In *Proceedings of World Academy of Science, Engineering and Technology*, volume 26, pages 600–605. World Academy of Science, Engineering and Technology, December 2007.

[12] G.B. Sheble and G.N. Fahd. Unit commitment literature synopsis. *IEEE Transactions on Power Systems*, 9(1):128–135, February 1994.

[13] Walter L. Snyder, H. David Powell, and John C. Rayburn. Dynamic programming approach to unit commitment. *IEEE Transactions on Power Systems*, 2(2):339–348, May 1987.

[14] S. Takriti, J.R. Birge, and E. Long. A stochastic model for the unit commitment problem. *IEEE Transactions on Power Systems*, 11:1497–1508, August 1996.

[15] Samer Takriti and John R. Birge. Lagrangian solution techniques and bounds for loosely coupled mixed-integer stochastic programs. *Operations Research*, 48(1):91–98, 2000.

[16] J. Valenzuela and M. Mazumdar. *Probabilistic Unit Commitment under a Deregulated Market*, chapter 8, pages 139–152. Volume 36 of Hobbs et al. [6], 2001.

[17] C. Wang and S.M. Shahidehpour. Effects of ramp-rate limits on unit commitment and economicdispatch. *IEEE Transactions on Power Systems*, 8(3):1341–1350, August 1993.