Power Scheduling for a Network of Distributed Generators

by

Etienne Jacques Dupuis

Bachelor of Science in Engineering, UNB, 2004

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

Masters of Science in Engineering

In the Graduate Academic Unit of Department of Electrical and Computer Engineering

Supervisor(s):	Dr. James H. Taylor, PhD, Elec. Eng.
	Dr. Balasubramanian Venkatesh, PhD, Elec. Eng.
Examining Board:	Dr. Chris Diduch, PhD, Elec. Eng.
	Dr. Howard Li, PhD, Elec. Eng.
External Examiner:	Dr. Donglei Du, PhD, Faculty of Business Admin.

This thesis is accepted

Dean of Graduate Studies

THE UNIVERSITY OF NEW BRUNSWICK

August, 2008

Abstract

Growing concerns over the economic and environmental costs of fossil fuels have spurred renewed interest in the area of renewable energy. The Atlantic Sustainable Power Research Initiative (ASPRI) aims to integrate low- and no-emission technology into Atlantic Canada's energy generation mix. This research focuses on the scheduling of distributed generators to improve their aggregate performance while maintaining a small ecological footprint. Combining renewable units with low-emission units using conventional fuels promises improved profitability and reliability over any single technology. Four Unit Commitment methods have been considered for the Energy Control Center (ECC), one of them novel. The novel solution appears to be the first to successfully solve the static Unit Commitment problem in less than exponential time with respect to the number of units - this is especially important when incorporating numerous distributed generators. The stochastic Unit Commitment problem is also considered in an effort to maximize the advantages of intermittent, renewable generators.

Dedication

À mon amour et à nos famille.

Table of Contents

List of Tables

List of Figures

List of Symbols, Nomenclature or Abbreviations

Typically subscripts denote the unit index and time index, superscripts denote the iteration. Deviations, when they occur, are highlighted in the text.

SYMBOLS

i	The unit index, the set of all unit indices is I.
J	Generic optimization objective function.
t	The time index, the set of all time indices is T.
a_i	Constant part of the i^{th} generator's cost function.
b_i	Linear part of the i^{th} generator's cost function.
c_i	Quadratic part of the i^{th} generator's cost function.
k	Iteration index for the Lagrangian methods.
$p_{i,t}$	Power supplied by the i^{th} generator at the t^{th} instant.
$C_i(\cdot)$	Cost of power function, typically $C_i(p) = a_i + b_i p_i + c_i p_i^2$, depending on context can also include startup/shutdown cost.
$\underline{p_i}$	Low limit on the i^{th} unit's power.
$\overline{p_i}$	High limit on the i^{th} unit's power.
λ	The Lagrangian variable, used in constrained optimization - represents the marginal cost of power in Lagrangian Relaxation.
D_t	The power demand at the t^{th} instant.
$\theta(\cdot)$	Shorthand for the constraint matrix, for example, the power balance constraint can be expressed $\theta(p) = D_t - \sum_{i \in I} p_{i,t}$.
$\langle \cdot, \cdot \rangle$	The scalar product of two vectors/matrices.
$\mathcal{L}(\cdot)$	Lagrangian or Dual function obtained by dualizing the constraint matrix. The shorthand $\mathcal{L}(p, \lambda) = J(p) + \langle \lambda, \Theta(p) \rangle$ shows $J(p)$ the 'primal cost' and $\mathcal{L}(p, \lambda)$ the 'dual cost'.
$\mathcal{L}_c(\cdot)$	The Augmented Lagrangian function, a penalty of weight c is added to the Lagrangian, $\mathcal{L}(p, \lambda) = J(p) + \langle \lambda, \Theta(p) \rangle + \frac{c}{2} (\theta(p))^2$.

ABBREVIATIONS

ASPRI	Atlantic Sustainable Power Research Initiative.
ALR	Augmented Lagrangian Relaxation, an extension of LR it is intended to improve the range of convergence.
DG	Distributed Generation, refers to generators with relatively small ratings, typically from a few tens of kilowatts to a few megawatts. The definition spans from renewable alternatives such as wind, solar or hydro to units using conventional fuels, which include but is not limited to scaled down versions of utility scale generators.
DP	Dynamic Programming, a scheduling method applied to integer programming problems. Finds an optimal path from optimal sub-paths.
ED	Economic Dispatch, the problem of finding the cost-optimal loading for a given set of generators while meeting the power balance constraint.
LR	Lagrangian Relaxation, a popular approach to the UC problem which uses a Lagrangian or Dual variable to permit unit by unit decomposition.
UC	Unit Commitment, the problem of finding the cost-optimal set of generators to meet a load.

Chapter 1

Introduction

Distributed Generators (DG) are electrical generators with relatively low power ratings, these typically range from a few tens of kW to a few MW. These small units can typically be placed closer to the point of consumption than would be admissible for their larger counterparts due to environmental, safety and other societal considerations. This factor has increased the interest for DG in some regions of the world, particularly where liberalized energy markets have replaced state monopolies. This is in part because adding generating capacity close to the point of consumption helps to defer costly investments in the electrical transmission system.

The investment required to establish DG units is in proportion to their physical size, therefore much less significant than for utility scale generators. This second factor has raised similar interest in privatized power pools where investment in large units with pay-back periods measured in decades seem even less attractive under volatile market conditions. These advantages are balanced by relatively low thermal efficiencies and limits on the types of fuels that can be economically transported, stored and consumed.

An Energy Control Center (ECC) for Distributed Generators is being designed for the Atlantic Sustainable Power Research Initiative (ASPRI) project. The scheduling of power production is an important component of modern energy systems and is an important tool in the marketing of electricity. An algorithm has been designed to provide an aggregate cost function which can be used by the ECC to support bidding to system operators without requiring an undue computational burden. Because wind turbines and photo-voltaic panels are included, special attention has been paid to the effect that intermittent resources can have on operation.

The installed wind and solar power capacity is following an upward trend worldwide. Some contributing factors include growing awareness of the impact on the climate of conventional generation as well as steadily increasing energy costs. Significantly the profitability of these options have improved thanks to technological innovation and to political resolve. The growing role of these intermittent resources present new and interesting challenges to the operation of the power system. One of these challenges is to the scheduling of generators which is dealt with here.

Short term power scheduling is termed Unit Commitment (UC); its aim is to decide which units to utilize over an upcoming period of time so as to minimize the cost of energy production while maintaining a balanced system with enough redundancy factored in to meet contingencies. The need to solve this problem over many time periods, and for systems with many generators, leads to an exponential growth in the number of possible schedules to consider. For systems of realistic size, the number of potential combinations is so large that it is infeasible to evaluate the full set in order to identify the optimal policy.

1.1 Literature Review

The scheduling of generators is usually performed at least a day if not a week in advance. This requires some anticipation of the power which will be required. This is addressed by the load forecast. Several approaches have been used including time series analysis [1], regression analysis [2] and neural networks [3, 4]. The final forecast can be based on a complex blend of meteorology, statistics and sociology but is still never perfect. For this reason, and to deal with unforeseen contingencies, some amount of spinning reserve must be available. Spinning reserve is a margin of generating capacity that is readily available in a timely manner to accommodate unexpected load increases or network contingencies.

Intermittent generators, such as wind turbines can complicate the forecasting task. Various attempts at forecasting wind power production are reviewed and compared in [5]. These authors indicate that numerical weather prediction models outperform time series approaches when the forecast horizon exceeds 3 to 6 hours. The lead time of at least 24 hours required for UC therefore necessitates such an approach when installed wind capacity is a significant component of the energy mix. Meanwhile, an interesting method for estimating confidence in the forecast is provided in [6].

Turning to the Unit Commitment problem itself, we find that many studies have been brought to bear on this problem; a survey paper [8] lists more than 150 articles. We will limit ourselves to presenting some of the basic methods and those that are important to our approach. The reason for such interest in this problem is two-fold, first the fuel cost borne by utilities is quite important and even modest improvements in the scheduling of these resources can have significant financial impact. The second reason is that optimal Unit Commitment remains an open problem for medium to large systems, a truly optimal scheme that operates in polynomial time has not yet been devised.

The priority list methods are one of the early algorithms for solving the UC problem [7, 9]. The priority list reduces the number of unit combinations which must be considered. Under a strict priority list, only N generator combinations need be considered for a system with N machines. The combinations are obtained by progressively adding units in order of economic merit to the previous combination. The schedule is then drawn by assigning the generators identified by the priority list to cover the load forecast. The method is certainly straightforward and its output is predictable, but the optimality of such an algorithm is far from assured because the optimality of the priority list itself may be in question, particularly as load fluctuates throughout the scheduling period.

Dual methods were first proposed to solve the UC problem in [10]. Dual or Lagrangian approaches are a hallmark of constrained optimization; they transform constrained problems into equivalent unconstrained problems. Lagrangian Relaxation (LR) is an extension of the Lagrangian methodology into the realm of integer programming.

In the LR solution to UC, the power balance constraint is dualized or 'relaxed', i.e, the coupling between generator decisions is removed. This permits each generator decision to be made independently of other generator decisions. In order to balance supply and demand, a dual variable which is analogous to the marginal price of power is introduced. Based on this price, the algorithm determines which units are profitable, and which set-points maximize their utility. The price is updated until the supply of power equals the demand. The duality gap, which is evaluated by comparing the 'primal cost' to the 'dual cost' is typically used to judge how far from the optimum the resulting solution is.

Restrictions on the rate at which units can be turned on and off couple the decisions in subsequent hours. Use of Dynamic Programming (DP) provides an efficient solution to the generator subproblems. Most LR approaches use the number of hours a unit has been continuously running or off as the state in DP and the cost of bringing a unit on line, or shutting it off, is then fully represented by the transition matrix. In [11] more states are added to the DP method to discretize generation levels; this allows solving the problem for units with ramp constraints.

The success of Lagrangian methods has motivated some investigators to add more features to the UC solution. Baldick proposes the generalized UC problem in [12] which considers all the usual constraints on generation plus line flow limits, voltage limits, ramp limits and total fuel/energy limits. Environmental constraints are introduced into the formulation in [13]. Of great interest, particularly for systems with high penetration of intermittent generation, is the development of stochastic UC algorithms in [14, 15, 16, 17]. Factoring in deviations from the forecasted power requirements promises to maximize the economic benefit of such intermittent generators as wind turbines and solar panels.

Typically, LR schedules require some adjustments to make them feasible. These adjustments can be related to line flow limits or other constraints which might not have been included in the UC model. More disconcerting is the tendency for LR algorithms to commit more capacity than necessary, leading to sub-optimal costs. The sequential de-commitment method introduced in [18] tries to address this issue.

There have been documented problems with setting up a centralized UC algorithm to assign units for power production. In the England and Wales system, the pool holds a power auction based on a UC model. Observers have noted apparently erratic system marginal prices for power [19]. It is noted that the deficiencies occur not only because suppliers can manipulate their bids, but also because of the inherent deficiencies of the LR solution. The LR algorithm can only converge if a competitive equilibrium exists; as shown in [20], this assumption does not hold in general.

The principal reason why Lagrangian Relaxation does not always yield truly optimal schedules is that only schedules which are optimal to each generating unit can be obtained; however optimizing the benefit of each producer is not always consistent with obtaining the lowest possible cost. The integer decisions inherent in the UC problem are linked to fixed costs, or indivisibilities which are not related to the specific level of production but rather to production itself. Indivisibilities tend to favor operation near the full rated output of the machine to offset the fixed cost. This behavior is critical to the optimality of the method but is also its main shortcoming because portions of the generator's cost curve are 'thrown out'. Under LR a generator will not commit to a policy that is sub-optimal to its own objective despite that it might be an optimal choice from a system view.

To improve the convergence of LR and to limit the need for sub-optimal heuristics which are used to obtain a feasible schedule Renaud proposes the use of an Augmented Lagrangian, or Penalty Method in [21]. The Augmented Lagrangian approach is based on general nonlinear programming principles first explored in [23]. A quadratic penalty term is added to the Lagrangian objective, this helps to offset the effect of fixed costs in the problem domain and aims to improve the range and speed of convergence.

The algorithms for solving the UC problem discussed thus far, priority lists, Lagrangian Relaxation and Augmented Lagrangian Relaxation each have specific advantages and disadvantages. The simplicity of priority lists is likely its' greatest asset, however the schedules it returns will generally be more expensive because many unit combinations that may be more attractive under fluctuating power requirements are excluded. The Lagrangian Relaxation algorithm can draw from a larger pool of unit combinations, one that rests on maximizing the utility of each generator through price based decomposition. When maximizing the benefit of each generator is in line with minimizing the overall cost, the LR algorithm will return the optimal solution. However if even one unit must take a loss for the 'greater good' the algorithm is not likely to return this option. Augmented Lagrangian Relaxation has many of the same advantages as LR with the added benefit of prompter and broader convergence. However this benefit comes at the cost of a much complicated iterative search. ALR's penalty term obstructs, although it those not preclude, the recourse to price based decomposition.

1.2 Scope of Work

The objective of this thesis is to review existing Unit Commitment algorithms with special attention to their applicability to the scheduling of Distributed Generators.

The widely used method of Lagrangian Relaxation is considered first, it uses price-based decomposition and permits complex inter-hour constraints to be observed.

The LR algorithm has limited convergence properties, for this reason, two Augmented Lagrangian Relaxation algorithms are considered next. The first ALR method adds a penalty term to the 'ordinary' Lagrangian objective and promises to improve convergence.

The second ALR algorithm is referred to as 'Variable Split Augmented Lagrangian'; it separates the UC problem into two separate problems one dealing with the network constraints and one dealing with the unit-wise constraints. A penalty term is then used to coax both problems to converge to the same solution.

As part of this work, a novel solution has been elaborated to deal with the requirements of DG scheduling. The method, 'Cost Curve Aggregation', is based on solving the static (single-hour) Unit Commitment problem. The method does not easily admit complex dynamic constraints. It does, however, provide solid single-hour commitment decisions which have been shown experimentally to be optimal. The small size of distributed generators translates to small inertias and very little coupling between scheduling instants. For this reason, the Cost Curve Aggregation solution appears to be very promising for DG scheduling.

Finally, the applicability of the Cost Curve Aggregation method to stochastic scheduling is investigated. An example featuring a bid-based scheduling problem with stochastic energy contributions from intermittent generators is included. The goal is to include as effectively as possible the contributions from units such as wind turbines or photo-voltaic cells into the electrical dispatch.

Chapter 2

Lagrangian Relaxation

The Lagrangian Relaxation method is widely applied to solve the Unit Commitment problem. The problem is formulated to make use of the fact that Unit Commitment, integer decisions aside, has much in common with a generic constrained optimization problem. Starting from a simplified definition of Unit Commitment

$$J = \min \sum_{i \in I} \sum_{t \in T} C_i(p_{i,t}) \tag{2.1}$$

$$\text{subject to} \quad \sum_{i \in I} p_{i,t} = D_t \ , \forall t \in T$$

$$p_{i,t} \in \{0\} \cup [\underline{p_i}, \overline{p_i}]$$

where $p_{i,t}$ is the i^{th} unit's power production at the t^{th} instant and $C_i(p_{i,t})$ is the cost incurred for this generator loading. Typically, the cost functions for the UC problem are quadratic functions of power [8, 30]. Linear and piece-wise linear functions are also sometimes used; whatever the choice of curve an empirical fit between generator power and fuel requirement and therefore cost is represented. The power consumed by the loads in the system at the t^{th} moment is represented by D_t. The set of scheduling instants is represented by T while the set of generators is held in I. The lower-case instances of i and t represent the members in the sets I and T. Finally, the generators are constrained to deliver power within a lower limit, $\underline{p_i}$ and an upper limit, $\overline{p_i}$ or else contribute no power to the dispatch while off line.

The LR method starts by dualizing the equality constraints to define the Lagrangian objective function $\mathcal{L}(p_{I,T}, \lambda)$

$$\mathcal{L}(p_{I,T}, \lambda) = \sum_{i \in I} \sum_{t \in T} C_i(p_{i,t}) + \sum_{t \in T} \lambda_t \left(D_t - \sum_{i \in I} p_{i,t} \right) \tag{2.2}$$
$$p_{i,t} \in \{0\} \cup [\underline{p_i}, \overline{p_i}]$$

The dual approach is quite common in constrained optimization and is also often referred to as the 'Lagrangian Method' after the 18^{th} century originator of the approach, Joseph Louis Lagrange. As will be shown, the set of dual variables introduced, λ_t, is used to coordinate the primal variables. The primal variables are those from the original objective, commitment decisions and power setpoints, they must be coordinated so that the power balance constraint is met at minimal cost. Note that the value λ_t is analogous to the marginal price of power at instant t.

2.1 Price-Based Decomposition

The principal advantage of the Lagrangian form is that the equality constraint on power generated which appears in (2.1) is moved into the objective. The equality constraints can be said to couple the variables at each hour; if one value is changed the other values must be adjusted to balance out the change so that the constraint holds.

The coupling of the variables precludes the possibility of optimizing one variable at a time. The Lagrangian form of the problem allows for decomposition precisely because this equality constraint is moved into the objective function. Note that we can rewrite the objective as

$$\begin{aligned}\mathcal{L}(p_{I,T},\lambda_T) = &\sum_{t\in T}\left[C_1(p_{1,t}) - \lambda_t p_{1,t}\right] \\ &+ \sum_{t\in T}\left[C_2(p_{2,t}) - \lambda_t p_{2,t}\right] \\ &+ \dots \\ &+ \sum_{t\in T}\left[C_I(p_{I,t}) - \lambda_t p_{I,t}\right] \\ &+ \sum_{t\in T}\lambda_t D_t\end{aligned} \tag{2.3}$$

We see that each line has variables associated with a single generator. The interdependence between generators has been relaxed by the introduction of the dual variables. We can proceed to look for the minimum of (2.3) by solving the generator subproblems with a given set of lambdas. Typically, this will return a schedule with either too much or too little power allocated. By adjusting the dual variables up or down, the power obtained at the next iteration can be adjusted. This yields the following iterative algorithm:

$$\begin{aligned}p_i^k &= \arg\min \sum_{t\in T}\left[C_i(p_{i,t}) - \lambda_t^k p_{i,t}\right] \\ \lambda_t^{k+1} &= \lambda_t^k + \alpha\left(D_t - \sum_{i\in I} p_{i,t}\right)\end{aligned} \tag{2.4}$$

where α is a step-size constant and the superscript k denotes the iteration. The above algorithm bears an economic interpretation; it is common to refer to the values of

λ_T as 'shadow prices'. The assumption is that there exists a price for which supply of power exactly equals the demand. If supply exceeds demand, the shadow price is reduced to drive down the offer. Alternatively, the supply can be increased by raising the price. The search proceeds until supply exactly meets demand. The existence of such a competitive equilibrium is uncertain because of the discontinuities that occur in integer programming problems.

2.2 Dynamic Programming

Following decomposition, a dual variable replaces the constraints on power demand. These new variables, one per scheduling instant, serve to separate the objective but also to coordinate individual generator decisions into a coherent system-wide schedule. The goal is to minimize the separated problems in (2.4), one per generator and each comprised of generating costs minus the symbolic compensation provided by the values of lambda.

Although drawing a schedule for a single machine is easier than for an entire group, the integer decisions associated with the on/off decisions are tied by inter-period constraints and still suffer the complication of exponential complexity. Note that using complete enumeration 2^{24} prospective schedules have to be considered for a 24 period problem. Even if the comparison of all these schedules can proceed fairly quickly it ultimately poses a tremendous computational burden. This burden is all the more important because it must be assumed for each unit and repeated as the values of lambda are adjusted to meet energy requirements. Dynamic programming efficiently solves the decomposed generator sub-problems to optimality.

Dynamic programming represents multi-period decision problems as a set of states, the costs to assume these states and to transition between them. In the generator scheduling problem, at least two states must be present; the on-state and the off-state. The 'costs' of these two states for the generator sub-problems are the net loss for the on-state and nothing for the off-state. The transition cost between these two states are then the start-up and shutdown costs.

Thermal units will often have start costs that vary based on how long the unit has been off. This situation results from the inertia of the thermal system. A boiler will require less heat input and therefore less fuel to return to service if it is still warm. Also, generally a unit must run for a certain number of hours before it can be turned off. Similarly, when off-line a unit cannot be returned to service immediately. These factors can be effectively represented by considering the state of a unit to be composed of its on/off status combined with the number of periods from the last state change. For the LR sub-problems the states are typically designated by the number of contiguous hours a unit has been on (positive values) or off (negative values).

To illustrate the application of dynamic programming to the generator subproblems, the cost matrix A, and the state vector S for a 3-hour problem are provided in (2.5). Each column in A represents an instant, each row a state corresponding to the state vector S. Each of the rows in S represents the number of contiguous hours a unit has been on (positive values), or off (negative values).

$$A = \begin{bmatrix} \rho_1 & \rho_2 & \rho_3 \\ \rho_1 & \rho_2 & \rho_3 \\ \rho_1 & \rho_2 & \rho_3 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \qquad S = \begin{bmatrix} 3 \\ 2 \\ 1 \\ -1 \\ -2 \\ -3 \end{bmatrix} \tag{2.5}$$

The values ρ_t are selected to minimize the net loss of the unit at hour t. The values of ρ_t are computed from the minimization

$$\rho_t = \min_{p_t} a + bp_t + cp_t^2 - \lambda_t^k p_t \tag{2.6}$$
$$p_t \in [\underline{p}, \overline{p}]$$

The preceding is the objective (2.4) re-written as a single period minimization. The first three terms in the objective (2.6) represent the cost of generation. The last term is the income of the unit for generating p_t under the analogy of λ as the marginal price of power. The negative values in the state vector S are for periods when a unit is not running; a unit reports zero loss in the A matrix for the hours for which it does not generate.

The cost of moving from one state to the next is represented in the transition matrix $\mathcal{T}$;

$$\mathcal{T} = \begin{bmatrix} \infty & \infty & \infty & \text{stop}(3) & \infty & \infty \\ 0 & \infty & \infty & \text{stop}(2) & \infty & \infty \\ \infty & 0 & \infty & \text{stop}(1) & \infty & \infty \\ \infty & \infty & \text{start}(1) & \infty & 0 & \infty \\ \infty & \infty & \text{start}(2) & \infty & \infty & 0 \\ \infty & \infty & \text{start}(3) & \infty & \infty & \infty \end{bmatrix} \tag{2.7}$$

Rows represent the 'from' state and columns represent the 'to' state. For example, $\mathcal{T}(3, 2)$ represents the cost of moving from state S(3)=1 to state S(2)=2. The zeros appearing in the off diagonal of the $\mathcal{T}$ matrix are present because there is no transition cost for a unit to stay on or to stay off. Minimum up and down time, as well as the rate of cooling of a unit can be included in the state dependent values of 'stop' and 'start'. The dynamic constraints of each unit are modeled in fairly high detail by the transition matrix; this is a distinct advantage of the LR methods.

Forward dynamic programming can be used to minimize the problem represented by the A, S and $\mathcal{T}$ matrices. Remember that these matrices represent the decomposed generator-wise problems from the Lagrangian objective (2.2). Algorithmic details of forward dynamic programming are covered in more detail in [24] and Section 4.5. The principal concept revolves around using the least cost paths from previous hours to obtain those of successive hours.

The dynamic programming formulation yields optimal schedules for each unit with respect to the Lagrange values. The method is able to cope with very complex constraints on the dynamic behavior of generators; however, because of the discontinuous nature of the UC problem, the existence of a 'perfect price' is uncertain. An algorithm of the form (2.4) is not guaranteed to converge.

2.3 Problems with Lagrangian Relaxation

We have seen that the Lagrangian Relaxation algorithm is essentially a search for the 'perfect price' with which to reward the generators for the power they generate in the hopes of eliciting 'perfect offers'. Perfect offers, of course, would be those that would minimize the overall cost while abiding by the relevant constraints. Because each unit optimizes its schedule based on the latest prices it is fair to say that when a Lagrangian Relaxation algorithm does converge the resulting solution is optimal for the entire system but also for each of the generators.

To illustrate some difficulties with this approach, we begin by plotting the 'free market supply' of a 4-unit system as a function of the shadow price. The 'free market supply' is obtained by summing the power generators make available for a given value of λ.

Each generator contribution is found by performing the p-optimization in (2.4) [1]. The power freely supplied as a function of λ is shown in Figure 2.1 for the generating group in Table 2.1. The overall power offer grows with the price per unit of power, but it does so in leaps and bounds. Unit 3 is the most economical; it starts to break even for values of λ near 7.5 but it quickly saturates to its maximum output. Near $\lambda = 10$, unit 2 starts to offer capacity to the market but also quickly becomes insensitive to increased prices because maximum output is reached. Unit 4 briefly joins in before Unit 1 and the entire capacity of the four units is reached for values near $\lambda = 13$ and above.

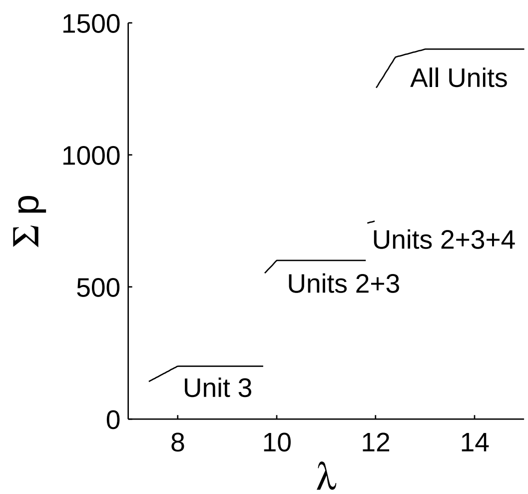

Figure 2.1: Aggregate power as a function of λ (Parameters in Table 2.1)

What we note from the mapping in Figure 2.1 is that it is *very* discontinuous. The search for a price which corresponds to a generation level of 400, for example, is doomed to fail because the mapping shows a large jump in that region. No matter how small the update step of the dual variable, the decomposed solution is confined to the four segments mapped in Figure 2.1. Because none of the segments intersect

[1]For the sake of simplicity the developments in this section are for a single hour schedule but the insights apply equally to the general T-hour problem.

the $\Sigma p = 400$ line, we will move from a solution with too little accumulated power to one with too much.

index	a	b	c	$\underline{p}$	$\overline{p}$
1	500	10	0.002	100	600
2	300	8	0.0025	100	400
3	100	6	0.005	50	200
4	200	9	0.01	50	200

Table 2.1: 4-unit parameters

Discontinuities are inherent in the unit commitment problem because of the fixed cost which must be assumed when a unit is committed. The contour map of profit and λ offers some insights into the discontinuities with which price-decomposition is faced. Remember that profit is a function of the generator cost function, its loading and the value of lambda, i.e., profit$= \lambda p - [a + bp + cp^2]$. The profit map is drawn for Unit 1 in Figure 2.2; positive contours denote profit while negative ones represent net losses. It is of significance that the 'break-even' contour in Figure 2.2 is near the upper power limit of the unit. The lowest value of λ for which the unit will at least break even is denoted by a star on the contour.

First refer to Figure 2.1 and note the second discontinuity near $\lambda = 12$. This discontinuity occurs as a result of unit 1 becoming profitable and has a magnitude of just over 500 power units; this is at first somewhat surprising as we might have anticipated a discontinuity not much greater than the lower limit of the unit which is roughly one fifth of this. By studying the profit map of unit 1 (Figure 2.2), we can explain why the discontinuity in the p-domain is far greater than the generator's lower limit of $p_1 = 100$. The smallest value of λ that the zero-profit contour intersects is marked with a star. The zero-profit contour is important because a generator in Lagrangian Relaxation will not be committed if it does not at least break even. This behavior

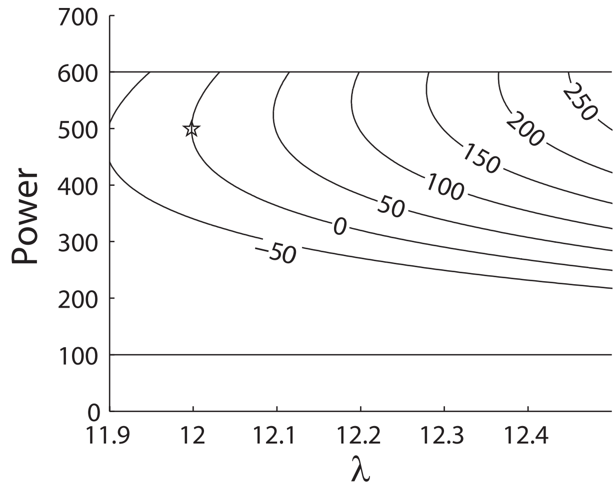

Figure 2.2: Profit contours of unit 1 as a function of λ and p_1

is critical as it drives the cost minimization of the algorithm, but in this case it amplifies the discontinuity associated with adding unit 1 to the dispatch by a factor of five. If we try to back-track by reducing the shadow price to lessen the transition, the algorithm will simply drop the unit from the commitment instead of curtailing its offer.

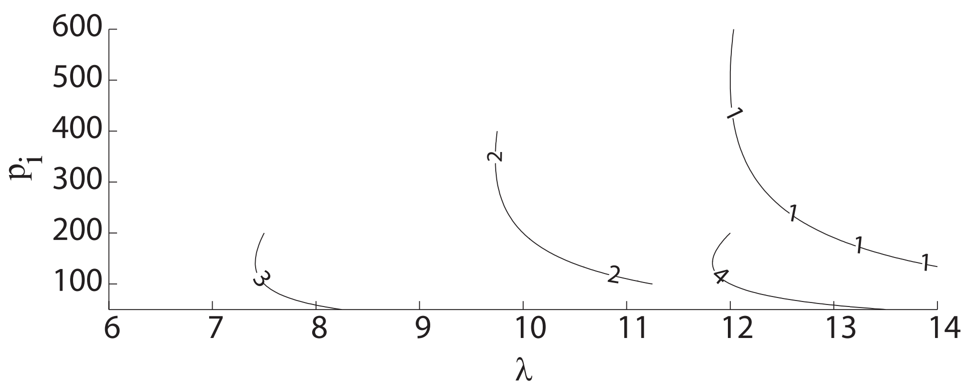

Figure 2.3: Zero-profit contours for the units in Table 2.1

The 'break-even' curves of the four generators are provided in Figure 2.3. We can see that the other generators in the 4-unit system will also not offer their full operating range in response to changes in λ. The amount of power can easily be adjusted in the Economic Dispatch calculations, but in Unit Commitment lowering the shadow price will drop the unit instead of de-rating it.

A further point can be made from Figure 2.3. The states that can be considered by the LR algorithm amount to a fairly restrictive priority listing. As the marginal price is increased new units are gradually added to the commitment group but never removed; the list of binary states that LR can search for this example is $\{0010, 0110, 01111, 1111\}$. The issue that arises is that even for our simple system, LR failed to include some of the unit combinations which lead to an optimal commitment such as $\{1110\}$ which is optimal for $D = 850$ as can be checked using complete enumeration. The Lagrangian Relaxation algorithm will not visit this state because there is no value of the shadow price for which unit 4 will suffer a loss while unit 1 breaks even.

The discontinuities in the p-domain, inherent in the LR algorithm, can be interpreted as blind spots in what this algorithm can 'view' of the primal cost curve. To illustrate this consider Figure 2.4, where the optimal cost curve is displayed along with the LR algorithm's cost mapping. The optimal cost curve can be obtained using either complete enumeration, which is reasonable for this size of a system, or cost curve aggregation presented in Chapter 4 for larger systems. The LR mapping is obtained by summing the costs associated with the levels of generation displayed in Figure 2.1. It is evident that the profit paradigm offers a limited mapping of the primal cost function; when the demanded power is 'off the map', it can be expected that the LR algorithm will not converge.

Of course the above arguments do not play to the strengths of Lagrangian Relaxation. Attention is drawn implicitly to the single hour UC problem to facilitate illustration. As mentioned earlier, a principal advantage of unit-by-unit decomposition is that it

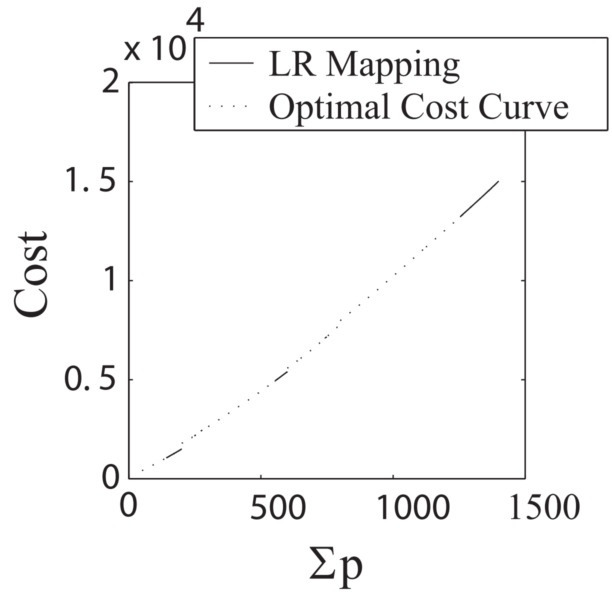

Figure 2.4: Optimal cost curve and LR mapping for units in Table 2.1

permits complex dynamic constraints to be considered. However the arguments apply equally to single- and multi-hour problems.

In the case of distributed generation, dynamic constraints can be expected to be less dominant than fuel costs. A method such as cost curve aggregation which yields solid single hour commitment solutions is appropriate for this niche in the power system scheduling field.

Despite the limitations on dual methods identified in this section, these approaches justifiably remain popular for their ability to incorporate complex dynamic behavior at the generator level. Two approaches which improve the convergence based on the Augmented Lagrangian method are presented in upcoming chapters.

Chapter 3

Augmented Lagrangian Methods

The Lagrangian Relaxation algorithm decomposes the Unit Commitment problem into a series of subproblems each associated with a generator. The smaller problems are eminently easier to solve than the master problem. Under certain conditions the algorithm converges to the optimal solution, however if the cost function is not strongly convex or if it is not smooth enough close to the solution the algorithm may alternate between an expensive solution and an infeasible one.

The convexity condition is a more general, if less specific, way of stating one of the shortcomings highlighted in subsection 2.3. The problem of convergence occurs because the problem domain is not convex enough, or bowl shaped enough to force convergence of LR. For our four-unit system, a new problem which is more 'bowl shaped' could be obtained by making the quadratic parameters in the cost function larger relative to the no-load cost. For example, if we multiply the quadratic parameters by a factor of 8 and divide the constants by a factor of 4 the problem domain will become more convex in relation to lambda. New break-even curves can be drawn for each unit, this is done in Figure 3.1 the new curves can be compared to those in

Figure 2.3.

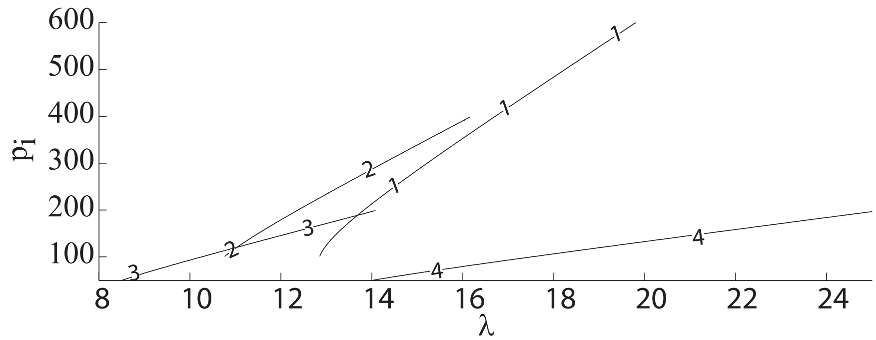

Figure 3.1: Break-even curves for a convexified problem

Note that the generators now offer their full range of operation as λ is tuned and the 'blind-spots' occurring in the LR mapping will be very much reduced. However, we are no longer solving the same problem and the solution of the convexified problem will have very little if anything to do with the solution of the original problem. Luckily, there is an approach which manages to 'convexify' the problem without distorting it, the Augmented Lagrangian method.

3.1 Geometric Interpretation

The most direct way of applying Augmented Lagrangian to the unit commitment problem is to add a quadratic penalty term to the ordinary Lagrangian. We start by defining the Lagrangian using the shorthand notation of [21].

$$\mathcal{L}(p, \lambda) = J(p) + \langle \lambda, \Theta(p) \rangle \tag{3.1}$$

where,

$$\Theta(p) = D - \sum_{i \in I} p_i$$

where $\langle \cdot, \cdot \rangle$ denotes the scalar product and $J(p)$ is the primal cost to be minimized. Note that the problem defined by (3.1) is the same as (2.3) but with a less cluttered, if less specific, notation.

The more general form of (3.1) allows us to draw certain conclusions as to the type of equilibrium to which Lagrangian Relaxation algorithm will try to converge. Using the Karush-Kuhn-Tucker rule for constrained optimization, we can determine that the equilibrium is a minimum in p for the primal cost function $J(p) = a + bp + cp^2$ if $c \geq 0$ because the second derivative is positive under this condition which indicates an upward-opening parabola. It is common to limit cost curves in the UC problem to be monotonically increasing with $c \geq 0$.

To determine the behavior with regard to λ at the equilibrium we start by taking the first derivative of $\mathcal{L}(p, \lambda)$, equating to zero and solving for p.

$$\frac{d\mathcal{L}(p, \lambda)}{dp} = b + 2cp - \lambda = 0 \tag{3.2}$$

$$p = \frac{\lambda - b}{2c} \tag{3.3}$$

By then substituting our value of p into $\mathcal{L}(p, \lambda)$ we obtain a function in λ only,

$$\mathcal{L}(\lambda) = a + b\left(\frac{\lambda - b}{2c}\right) + c\left(\frac{\lambda - b}{2c}\right)^2 + \lambda\left(D - \frac{\lambda - b}{2c}\right) \tag{3.4}$$

of which the second derivative is

$$\frac{d^2\mathcal{L}(\lambda)}{d\lambda^2} = \frac{1}{2c} - \frac{1}{c} = -\frac{1}{2c} \tag{3.5}$$

We note that the second derivative is negative under the assumption that $c \geq 0$, therefore the optimum is a maximum in λ.

We can therefore conclude that the equilibrium of the Lagrangian Relaxation algorithm is a saddle point because the second derivative along one axis is positive and along the other it is negative. Following this, one will not be surprised by the shape of the contour plot of $\mathcal{L}(p, \lambda)$ for our four unit system in Figure 3.2. The saddle point is marked with an 'x'. Note that in order to evaluate the contours in Figure 3.2 one must have access to the optimal cost curve as a function of power, namely $J(p)$ in (3.1). This curve can be obtained using either complete enumeration or the cost curve aggregation method.

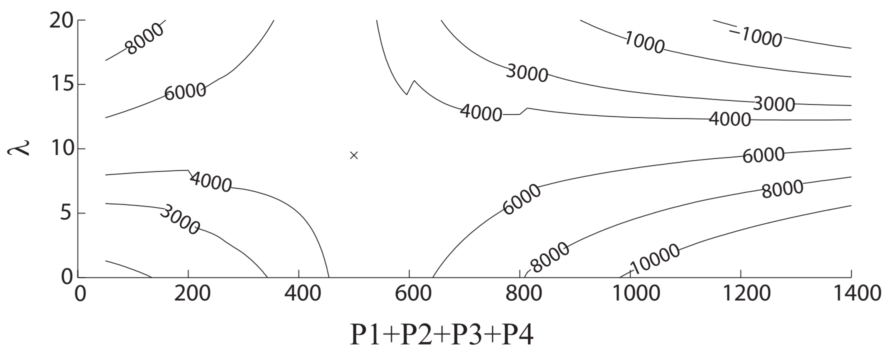

Figure 3.2: Contours of $\mathcal{L}(p, \lambda)$ for a demand of $D = 500$

The saddle-point to which the LR algorithm should converge will generally exist, but the discontinuities in the problem domain will lead to difficulties in estimating $J(p)$ using this paradigm. Evidence of the discontinuities is clear on some of the contour lines, where sharp jumps break up the smooth curves.

To make the saddle point more attractive we add a penalty term to obtain the Augmented Lagrangian function,

$$\mathcal{L}_c(p, \lambda) = J(p) + \langle \lambda, \Theta(p) \rangle + \frac{c}{2} \left(\Theta(p) \right)^2 \tag{3.6}$$

Its contours are depicted in Figure 3.3.

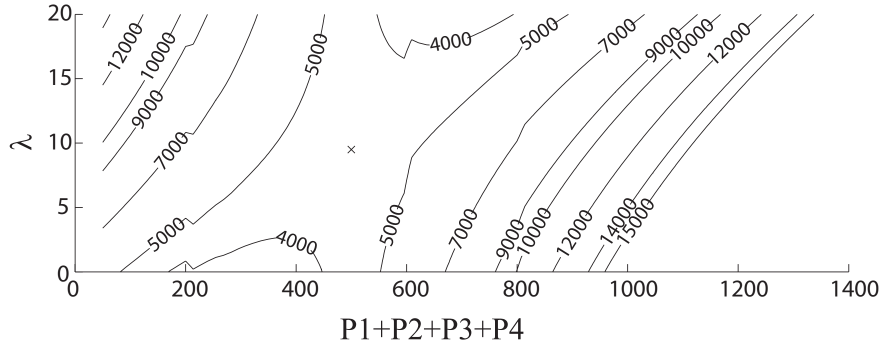

Figure 3.3: Contours of $\mathcal{L}_c(p, \lambda)$ with $c = .05$ and $D = 500$.

Note that the Augmented Lagrangian contour is now much steeper when traversed parallel to the p-axis, this precipitates convergence of the primal variables. The contours have been folded into a fairly narrow valley and the discontinuities have been pushed further from the saddle-point. The drawback is that steepness along the λ-axis is reduced, we can therefore expect progress of the dual variables toward the optimum to be impeded.

By comparing the plots in Figure 3.2 and Figure 3.3 we note that the shape has changed, yet the location of the saddle point[1] remains the same, the observation is important because it indicates that we have not altered the problem. In [23] it is shown that the function (3.6) has the same optimum as (3.1) and its differential is Lipschitz for a large enough value of c.

3.2 An Example

The usefulness of a differential that is Lipshitz is evident in Renaud's 2-unit example [21], the parameters are provided in Table 3.1, the UC problem for this simple example is written in (3.7).

[1] At the saddle point in both cases $\Sigma p = 500, \lambda = 9.5, \mathcal{L} = \mathcal{L}_c = 4425$

index	a	b	c	$\underline{p}$	$\overline{p}$
1	0	1	0	0	1
2	0	2	0	0	1

Table 3.1: 2-unit example from [21]

$$\min_{p1,p2} (p_1 + 2p_2) \tag{3.7}$$

$$\text{subject to}$$

$$p_1 \in [0, 1], \qquad p_2 \in [0, 1],$$

$$p_1 + p_2 = 1.5$$

This 2-unit problem can be solved using many approaches, perhaps the most direct is to map the feasible region, the cost contours and the power constraint, see Figure 3.4. The optimal solution is $p_1 = 1, p_2 = .5, J = 2$.

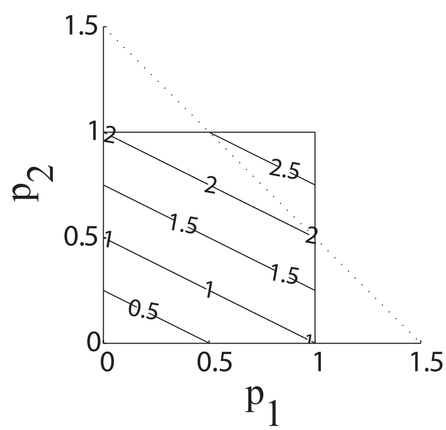

Figure 3.4: Graphical solution of (3.7)

Alternatively, if we are to solve the UC problem using a dual approach, we start by writing the Lagrangian

$$\mathcal{L}(p_1, p_2, \lambda) = p_1 + 2p_2 + \lambda(1.5 - p_1 - p_2) \tag{3.8}$$

Decomposing the problem leads us to the Lagrangian Relaxation algorithm,

$$p_1^{k+1} = \arg\max_{p_1} p_1 - \lambda^k p_1 \tag{3.9}$$

$$p_2^{k+1} = \arg\max_{p_2} 2p_2 - \lambda^k p_2 \tag{3.10}$$

$$\lambda^{k+1} = \lambda^k + \mu(1.5 - p_1 - p_2) \tag{3.11}$$

When we solve the above program we note that it does not converge, see Figure 3.5. The dual variable oscillates near its optimal value of $\lambda^\star = 2$ and p_2 assumes a value of zero or one. The weakly convex nature of the cost function in this example can be used to explain the lack of convergence. If the cost function of unit 2 was more strongly convex (had a bigger quadratic component) the slope at $p_2^\star = .5$ would be distinct from the slope at other points. Choosing the value of λ which corresponds to the slope of the cost function at $p_2^\star$ would then solve the problem to optimality. Because the cost function is linear, the slope is the same for the entire range and λ can not be used to pinpoint a value of the primal variable.

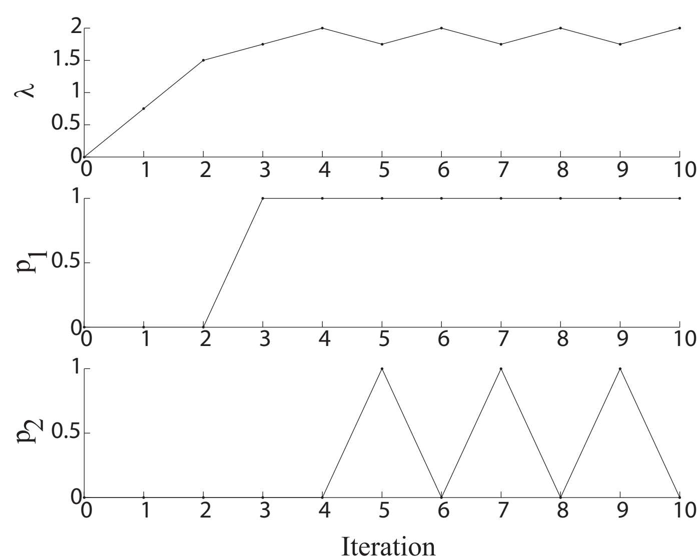

Figure 3.5: LR solution to (3.7), $\mu = .5$

To further understand why the LR algorithm is unstable in this case, it is interesting to consider the shape of the dual function $\mathcal{L}(p_1, p_2, \lambda)$ as a function of λ, Figure 3.6. The curve is obtained by finding the optimal values of p_1, p_2 for a given λ and then computing $\mathcal{L}(p_1, p_2, \lambda)$. First note that the Dual function is strictly, if not strongly,

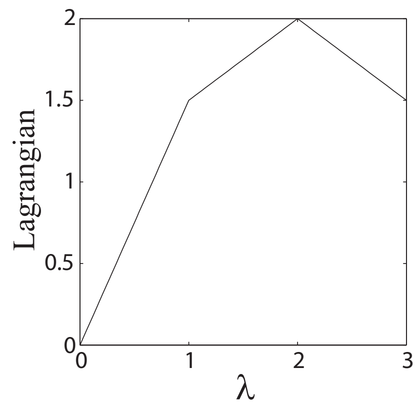

Figure 3.6: Dual as a function of λ

convex and has the inflection predicted by our derivation of the saddle point for the Lagrangian function. The important feature to note is that the function is not smooth at its peak. The sharp crest, at $\lambda = 2$, means that the function does not have a derivative at this point. The iterative algorithm is confounded by this feature of the dual curve as it will only settle if the optimum has a derivative equal to zero.

If we now consider the addition of a quadratic penalty term to the Lagrangian, to form the Augmented Lagrangian

$$\mathcal{L}_c(p_1, p_2, \lambda) = \mathcal{L}(p_1, p_2, \lambda) + \frac{c}{2}\left(1.5 - p_1 - p_2\right)^2 \tag{3.12}$$

which can also be solved for the optimal values of p_1, p_2 given λ. The optimization is somewhat more difficult because the variables are now linked through the penalty term. Luckily, we know that there is only one combination of units which is feasible, namely $[u_1, u_2] = [1, 1]$, as neither of the units has enough range to cover the demand

$D = 1.5$ on its own. The distinction makes decomposition unnecessary because its principal use is to determine the value of these binary variables. Solving the Augmented Lagrangian (3.12) results in the quadratic program

$$\mathcal{L}_c(\lambda) = \min_x x^T Q x + f^T x \tag{3.13}$$

$$\text{subject to, } Ax = b$$

$$\text{where,}$$

$$x = \begin{bmatrix} p_1 \\ p_2 \end{bmatrix}, f = \begin{bmatrix} 1 - \lambda - 1.5c \\ 2 - \lambda - 1.5c \end{bmatrix}, Q = \begin{bmatrix} c & c \\ c & c \end{bmatrix},$$

$$A^T = \begin{bmatrix} 0 & 1 & 0 & -1 \\ 1 & 0 & -1 & 0 \end{bmatrix}, b^T = \begin{bmatrix} 1 & 1 & 0 & 0 \end{bmatrix}$$

The above can be solved using any generic quadratic programming solver. The curves obtained for $c = 1$ and $c = 10$ are shown alongside the ordinary Lagrangian in Figure 3.7. Note that the optimum is the same for the three dual functions. Also note that the augmented curves are smooth and have well defined gradients at the optimum. This result means that an iterative solver applied to this objective function will be able to converge to the optimum and to pinpoint the appropriate generator loadings.

Note that the Augmented Lagrangian dual function becomes flatter as c is increased such that the gradient gradually approaches zero everywhere. The lack of a prominent gradient affects the convergence of iterative solvers. Convergence to $\lambda^\star$ is precluded for c large enough.

A look at the primal subspace in Figure 3.8 confirms that the quadratic penalty term improves the convergence of the primal problem. As the value of c is increased, we note that the cost surfaces become steeper. The contours of the ordinary Lagrangian

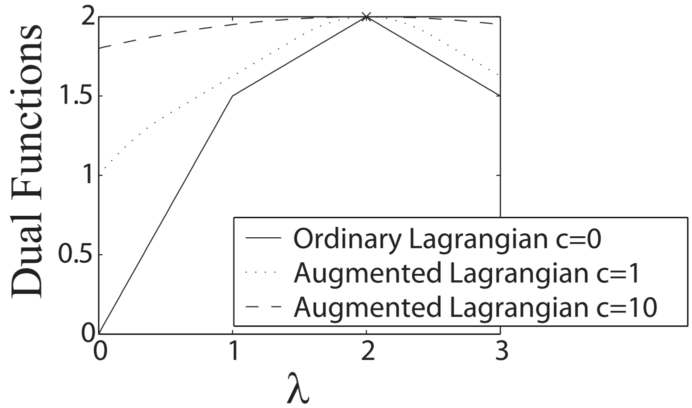

Figure 3.7: Augmented Lagrangian curves in λ subspace, D=1.5

(c=0) illustrate that the optima $p_1^\star = 1, p_2^\star = .5$ is not attractive because the objective is independent of the value of p_2 when $\lambda = 2$ (the p_2 terms cancel out in (3.8) when $\lambda = 2$). The addition of a quadratic penalty term permits the primal problem to go from an ill-conditioned one to one which is readily solved. The trade-off associated with choosing a value of c is not trivial and should be considered carefully when implementing an Augmented Lagrangian solution, especially when decomposition of the problem domain is to be applied. The previous applies because the convergence properties of the primal and dual problems must be counter-balanced; an improvement in one is a worsening of the other.

3.3 The Perturbed View

A perspective found in [25] allows one to appreciate the improved range of convergence which the Augmented Lagrangian permits. First consider a 'perturbed' constrained optimization problem,

$$\varphi(\nu) = \min_p J(p) \qquad \text{subject to} \qquad \Theta(p) = \nu \tag{3.14}$$

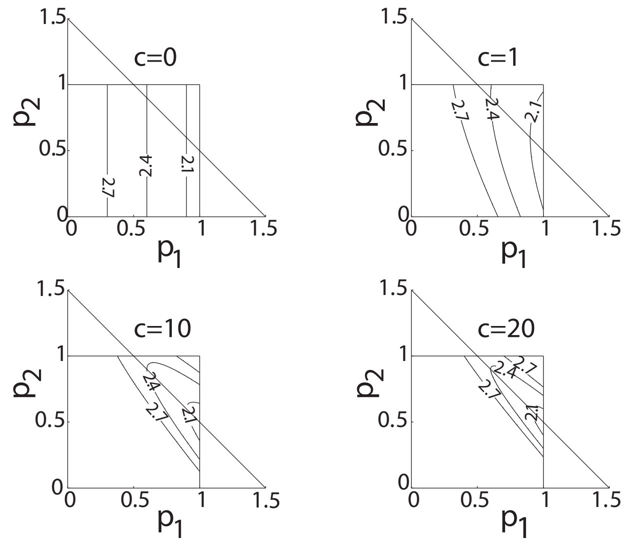

Figure 3.8: Augmented Lagrangian contours in primal subspace, $\lambda = 2$, D=1.5

The problem is said to be disturbed because the constraints equate to ν instead of zero as in 3.1. We plot $\varphi(\nu)$ as a function of the disturbance ν in Figure 3.9 for Renaud's 2-unit system. Note that $\varphi(\nu)$ corresponds to $J^*(p)$ with ν substituted for p. The ordinary Lagrangian of (3.14) for values near $\nu = \nu^\star$ can then be expressed,

$$\mathcal{L}(\nu, \lambda) = \varphi(\nu^\star) - \langle \lambda, \nu^\star - \nu \rangle \,. \tag{3.15}$$

From the above notation notice that the Lagrangian represents a hyperplane which touches the primal curve at $\nu = \nu^\star$ or $\Theta(p) = 0$. The parameter λ allows us to choose the slope of the plane, while the value of $\varphi(\nu^\star) - \lambda\nu^\star$ determines the height at the origin $\nu = 0$. Similarly the Augmented Lagrangian,

$$\mathcal{L}_c(\nu, \lambda) = \varphi(\nu^\star) - \langle \lambda, \nu^\star - \nu \rangle - \frac{c}{2}\,(\nu^\star - \nu)^2 \tag{3.16}$$

describes a parabola which just touches the primal curve at $\nu = \nu^\star$.

Figure 3.9 shows the primal curve along with the dual plane and parabolas for Re-

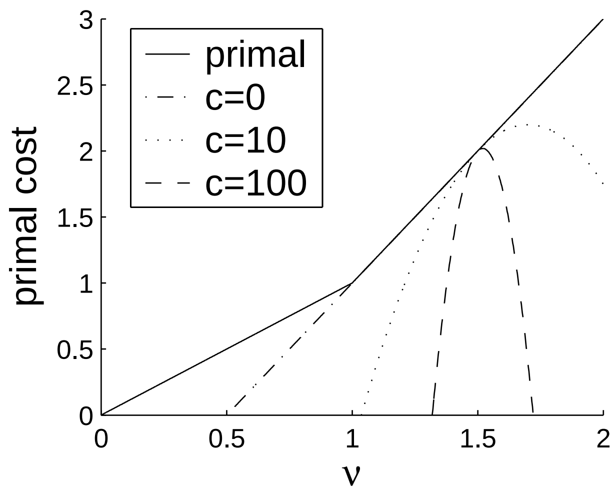

Figure 3.9: Perturbed view of dual functions $\nu^\star = 1.5$

naud's 2-unit problem. One notes that the ordinary Lagrangian ($c = 0$) is indefinite because the plane touches a large section of the primal curve, which is coherent with our earlier conclusions. As for the Augmented Lagrangians, we see that they both intersect the primal curve at $\nu = \nu^\star = 1.5$ which is the solution of the primal problem.

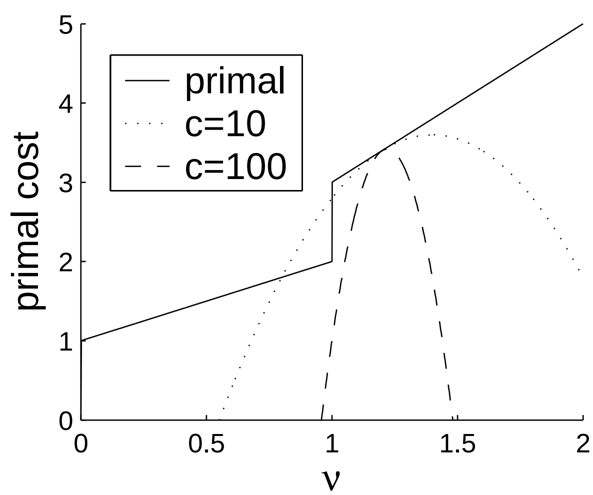

Figure 3.10: Dual hyperplanes for a discontinuous problem $\nu^\star = 1.2$

Note that increasing the value of c results in a sharper parabola with which to probe the primal cost curve. This feature of the Augmented Lagrangian is useful when the primal domain contains discontinuities. A conspicuous example is of course the UC problem which has a discontinuous objective function because of the fixed costs of generation. This is illustrated in Figure 3.10 where a fixed cost of 1 has been added to the cost function of each unit. We note that a large value of c is required such that the Augmented Lagrangian intersects the primal curve at a single point. If the solution sought is too close to a discontinuity the value of c will need to be large and the convergence properties of the dual problem will be poor.

3.4 Decomposed Algorithm

To improve the applicability of dual methods a quadratic penalty term is added to the Lagrangian objective function; as discussed previously, this approach is usually referred to as the Augmented Lagrangian method or sometimes as multiplier method. Consider (3.17), the Augmented Lagrangian where the power constraint is bolstered with a quadratic penalty term,

$$\mathcal{L}_c(p_{i,t}, \lambda_t) = \mathcal{L}(p_{i,t}, \lambda_t) + \frac{c}{2} \sum_{t=1}^{T} \left[\left(D_t - \sum_{i=1}^{N} p_{i,t} \right)^2 \right] \tag{3.17}$$

where $\mathcal{L}(p_{i,t}, \lambda_t)$ is the ordinary Lagrangian function for unit commitment, repeated below for convenience.

$$\mathcal{L}(p_{i,t}, \lambda_t) = \sum_{t=1}^{T} \left[\sum_{i=1}^{N} C_i(p_{i,t}) + \lambda_t \left(D_t - \sum_{i=1}^{N} p_{i,t} \right) \right] \tag{3.18}$$

The function $\mathcal{L}_c(p_{i,t}, \lambda_t)$ is shown to have the same optimum as $\mathcal{L}(p_{i,t}, \lambda_t)$ as well as being strictly convex for c large enough in [23]. A drawback is that the penalty term introduces non-separable terms; this result is unfortunate as our interest in dual methods is precisely that they permit decomposition. The non-separable terms are shown clearly by considering the penalty term added to a 2-unit problem.

$$\frac{c}{2}(D - p_1 - p_2)^2 = \frac{c}{2}\left(D^2 - 2Dp_1 - 2Dp_2 + p_1^2 - 2p_1p_2 + p_2^2\right) \tag{3.19}$$

The presence of the term p_1p_2 makes decomposition much less straightforward, as we cannot separate the objective function into a sum of terms associated with generator one and two alone. Nonetheless, iterative solutions can still be devised. In [21] the Auxiliary Problem Principal is proposed. First, 'proximal terms' are added to the objective function. These proximal terms keep subsequent iterations relatively close together which aids convergence. The 'prox' terms for a 2 unit problem are

$$\frac{b}{2}\left((p_1 - p_1^k)^2 + (p_2 - p_2^k)^2\right) \tag{3.20}$$

where the superscripts 'k' indicate that these values were obtained at the k-th iteration. The constant b is chosen as $2c$ in [22], and a convergence proof is provided for this choice in [26]. The decomposed iterative algorithm is obtained by assuming that the variables at $(p_1^k, p_2^k, \ldots, p_I^k, \lambda^k)$ are all fixed except for p_i

$$p_i^{k+1} = \arg\min_{p_i} \mathcal{L}_{cd}^{k+1}(p_i) \tag{3.21}$$

$$\mathcal{L}_{cd}^{k+1}(p_i) = C_i(p_i) + \frac{b}{2}(p_i - p_i^k)^2 - \lambda^k p_i - cp_i\left(D - \sum_{j=1}^{N} p_j^k\right) \tag{3.22}$$

$$\lambda^{k+1} = \lambda^k + \mu\left(D - \sum_{i=1}^{N} p_i^{k+1}\right) \tag{3.23}$$

where $\mathcal{L}^k_{cd}(p_i)$ is the portion of the objective which the i^{th} unit affects at the k^{th} iteration. The lowest point within the operating range is obtained by taking the first derivative of the objective in (3.22), setting it equal to zero and solving for p_i. By using the usual $C_i(p_i) = a_i + b_i p_i + c_i p_i^2$ we obtain

$$p_i^{k+1} = \frac{\lambda^k + c\left(D - \sum_{j=1}^{N} p_j^k\right) - b_i + bp_i^k}{2c_i + b} \tag{3.24}$$

$$p_i^{k+1} = \min(p_i^{k+1}, \overline{p_i}) \tag{3.25}$$

$$p_i^{k+1} = \max(p_i^{k+1}, \underline{p}_i) \tag{3.26}$$

The equations (3.24), (3.25) and (3.26) are used to find the optimal p_i at each hour. The values of $\mathcal{L}^k_{cd}(p_i^k)$ then take the place of the ρ values in the Dynamic Programming cost matrix. The transition matrix is the same as per Lagrangian Relaxation.

3.5 Convergence of the Decomposed Algorithm

Use of the Augmented Lagrangian improves convergence compared to Lagrangian Relaxation. The penalty term leads to an extended range of convergence for an appropriate value of c. Choosing the magnitude of the penalty factor is not at all trivial. For small values of c, the Augmented Lagrangian is too similar to the ordinary Lagrangian and convergence remains elusive. Conversely, if c is selected too large the dual problem becomes ill-conditioned.

We return to the 4-unit system from Table 2.1, as it is complex enough to illustrate the difficulties of the approach while being simple enough to allow resorting to complete enumeration as needed. Plotting of the Augmented Lagrangian objective as a function of the dual variable requires this enumeration. We do not have the leisure of knowing the optimal commitment a priori, so we solve a quadratic program of the form (3.13) for each of the possible commitments and choose the lowest objective values for each λ to find $\mathcal{L}_c(\lambda)$.

Let us first consider a case when the value of c is too small to assure convergence. Consider the four graphs in Figure 3.11, the first graph (upper left) is the perturbed view introduced in subsection 3.3. Note that the parabola intersects the primal curve twice which indicates that ALR will provide an ambiguous result. The second graph moving clockwise shows us the dual functions along with the location of the optimum marked 'x'. The peak of the Augmented Lagrangian marked 'o' occurs at the intersection of two commitments $\{0110\}$ and $\{0111\}$, the difference in slopes makes it such that the ALR has an indefinite differential at its peak. The last two graphs show the evolution of the primal and dual variables in the decomposed algorithm. The same convergence problem observed in Renaud's 2-unit problem is evident in the last two graphs; the shadow price oscillates about the ALR peak $\lambda \approx 12.5$ and the algorithm never converges to a solution which satisfies the power balance constraint.

In Figure 3.12 the value of c is increased until the dual parabola touches the primal curve at single point. The peak of the dual curve in the second graph coincides with the optimum, which is what we expect for a large enough value of c. The last two graphs show that the decomposed ALR algorithm can converge to the optimal solution but lacks globally convergent properties. With good initial conditions, the algorithm converges to the optimum. However some initial conditions can cause the primal variables to oscillate, it is not simple to guarantee good initial starting conditions for the algorithm.

The need for a large penalty factor is directly related to the proximity of the solution

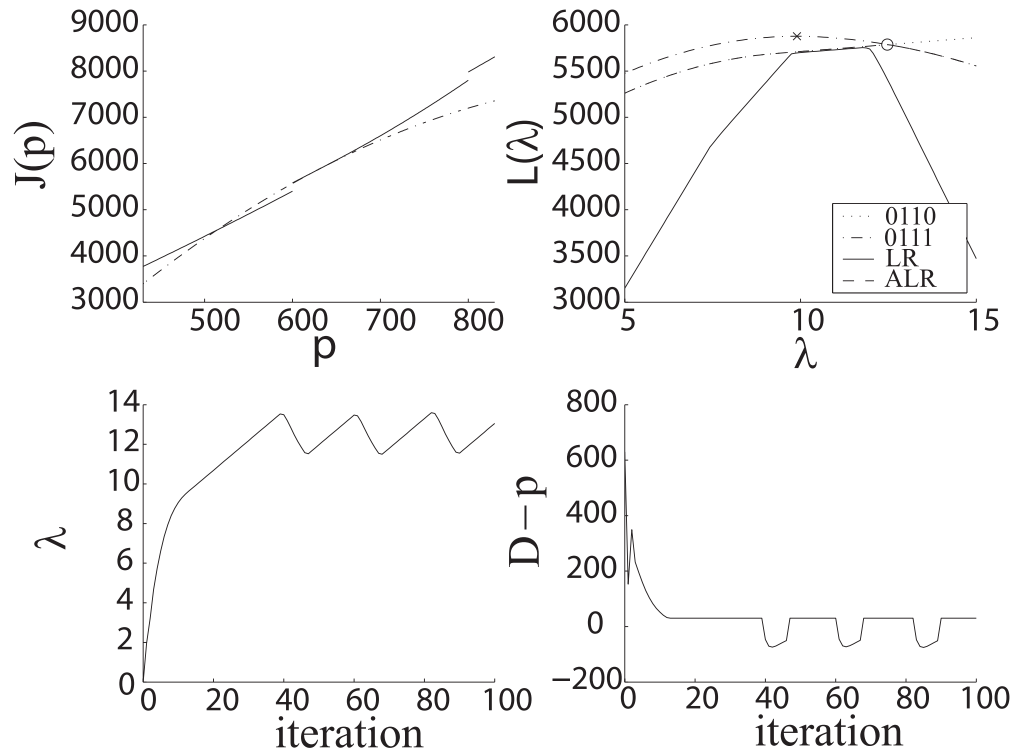

Figure 3.11: Failure to solve for $D = 630$ with $c = .025$

to a discontinuity in the primal cost curve. The discontinuities in UC are very 'hard', they result from the addition of fixed costs as units get committed. For example, the slope at $p = 600$ is essentially infinity thanks to the inclusion of unit 1 at this point. As we approach the discontinuity from the right hand side we require a value of $c = \infty$ at the limit so that the dual parabola does not intersect the primal curve for $\{0110\}$. It is not generally possible to know where the discontinuities in the primal domain occur, so an investigator may still obtain an ambiguous result for an arbitrarily large value of c.

The oscillation of the primal variables noted in the last plot of Figure 3.12 is linked to c exceeding an unspecified threshold. The phenomenon can be explained as resulting from the penalty term dominating over the cost functions of the machines being modeled. The difference between power generated and the demand swings violently

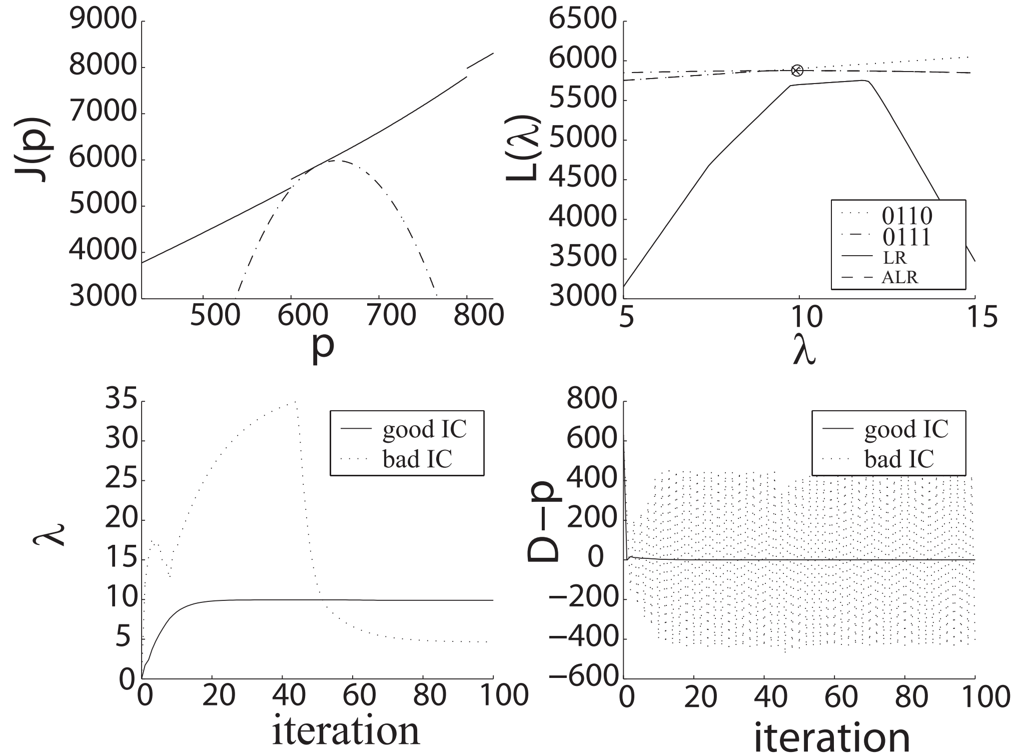

Figure 3.12: Solving for $D = 630$ with $c = .45$

at each iteration while the shadow price settles out near $\lambda = 5$. Because the surplus in generation at one iteration is very nearly the deficit of the following, the dual variable ceases to change appreciably and its ability to direct the solution is canceled.

The unstable equilibrium to which the primal variables may converge when c is large enough is analogous to what happens when two people meet head-on in a corridor. As both individuals try to avoid collision they find that they have both side-stepped into their neighbors new path. The awkward dance which ensues lasts until a feasible solution is reached at random, through convention or mutually agreed coordination.

Returning to the UC problem, we note that, as in the corridor example, it is the disaggregation of individual decisions and the impulse to reach a feasible solution which leads to seemingly irrational behavior. The deterministic structure in which the generator sub-problems are solved precludes the possibility that a feasible solution will be reached at random. Imposing a convention as to which generators should yield might work, but the optimality of the final solution would be undermined. This is a concern we did not have in the corridor example, as any solution that does not lead to a collision is close enough to optimality. Finally, coordination implies that we devise a way for the generators to work together instead of determining their state in isolation. The cost curve aggregation method, a novel approach described in Chapter 4, offers a measure of coordination by directly comparing and ordering the generator cost characteristics.

3.6 Variable Split Augmented Lagrangian

In [21], Arnaud Renaud rearranges the unit commitment problem into two less-constrained problems, one for the dynamic constraints and one for the system-wide constraints. The set of variable $p \in P$ as seen previously is split into the variables $d \in D$ and $s \in S$ which respectively meet some of the constraints: the variables d satisfy the dynamic constraints but pay no heed to system constraints, while s is the opposite. Using script capital letters to represent sets and lower case to denote the members, the dynamic constraints for the UC problem can be expressed as,

$$d_{i,T} \in \mathcal{D}_i, \quad \forall i \in I \tag{3.27}$$

where I is the set of units and T is the set of hours for which the problem is solved. Then $d_{i,T} \in \mathcal{D}_i$ is understood to constrain the i^{th} unit to operate within its operating domain for all hours in T. The operating domain of each unit is determined by its initial state along with minimum up/down time, power limits and ramp rates if required. The value $d_{i,t}$ is analogous to $p_{i,t}$ as it represents the loading of the generator i at hour t. The constraints in (3.27) ensure that the returned schedules are feasible on a unit-by-unit basis.

The second set of variables address system wide constraints. Formally these constraints are expressed as,

$$s_{I,t} \in \mathcal{S}_t, \quad \forall t \in T \tag{3.28}$$

which translates to the condition that the variables $s_{I,t}$ must be within the operating domain defined by $\mathcal{S}_t$ for each hour t. In our development, we limit ourselves to

$$\sum_{i \in I} s_{i,t} = D_t, \quad \forall t \in T \tag{3.29}$$

which is the by now familiar power balance constraint. In [22] the variable split approach is used to address the power balance constraint and maximum power flow constraints at buses.

To ensure that the two new problems lead to the same answer, and ultimately to the solution of the original problem, the solutions for the 'd-problems' and the 's-problems' must come together; this is addressed by adding a third equality constraint,

$$s_{I,T} = d_{I,T} \tag{3.30}$$

It is this third set of constraints that gets a quadratic penalty term and leads to an alternative way of implementing an Augmented Lagrangian Relaxation scheme. Instead of one value of lambda for each hour in the problem there is now a vector of lambdas at each period with a distinct dual variable for each of the generators.

3.7 Decomposed Algorithm

The new problem obtained by splitting the constraints between variables is solved using Augmented Lagrangian Relaxation. The Augmented Lagrangian of the variable split objective for two units and a single hour is expressed;

$$\begin{aligned}\mathcal{L}_{VS} =& C_1(d_1) + \lambda_1(s_1 - d_1) + \frac{c}{2}(s_1 - d_1)^2 \\ &+ C_2(d_2) + \lambda_2(s_2 - d_2) + \frac{c}{2}(s_2 - d_2)^2\end{aligned} \tag{3.31}$$

$$\begin{aligned} s.t.\ \ & s_1 + s_2 = D \\ & d_1 \in \{0\} \cup [\underline{p_1}, \overline{p_1}] \\ & d_2 \in \{0\} \cup [\underline{p_2}, \overline{p_2}] \\ & d_1 = s_1,\ \ d_2 = s_2 \end{aligned}$$

Note that there are four primal variables and two dual variables in the problem (3.31) despite the fact that only two variables must be determined in the original problem (the loading of generators 1 and 2).

As before, our interest in the dual methods is that they provide a path to decomposition of the problem. The iterative solution of Batut and Renaud [22] includes proximal terms, as seen earlier in the direct application of ALR to the UC problem (3.20). In the case of the variable split approach for two generators, the proximal terms are

$$\frac{b}{2}\left((d_1 - d_1^k)^2 + (d_2 - d_2^k)^2 + (s_1 - s_1^k)^2 + (s_2 - s_2^k)^2\right) \tag{3.32}$$

Applying decomposition by using the Auxiliary Problem Principle yields the d-values at each iteration;

$$d_1^{k+1} = \arg\min_{d_1} C_1(d_1) - \lambda_1^k d_1 - c d_1 \left(s_1^k - d_1^k\right) + \frac{b}{2}\left(d_1 - d_1^k\right)^2 \tag{3.33}$$

$$s.t. \quad d_1 \in \{0\} \cup [\underline{p_1}, \overline{p_1}]$$

$$d_2^{k+1} = \arg\min_{d_2} C_2(d_2) - \lambda_2^k d_2 - c d_2 \left(s_2^k - d_2^k\right) + \frac{b}{2}\left(d_2 - d_2^k\right)^2 \tag{3.34}$$

$$s.t. \quad d_2 \in \{0\} \cup [\underline{p_2}, \overline{p_2}]$$

and the s-values;

$$s_1^{k+1} = \arg\min_{s_1} \lambda_1^k s_1 + c s_1 \left(s_1^k - d_1^k\right) + \frac{b}{2}\left(s_1 - s_1^k\right)^2 \tag{3.35}$$

$$s_2^{k+1} = \arg\min_{s_2} \lambda_2^k s_2 + c s_2 \left(s_2^k - d_2^k\right) + \frac{b}{2}\left(s_2 - s_2^k\right)^2 \tag{3.36}$$

$$s.t. \quad s_1 + s_2 = D$$

Finally, the decomposed optimization problems are coordinated through the update of the λ-values.

$$\lambda_1^{k+1} = \lambda_1^k + c\left(s_1^{k+1} - d_1^{k+1}\right) \tag{3.37}$$

$$\lambda_2^{k+1} = \lambda_2^k + c\left(s_2^{k+1} - d_2^{k+1}\right) \tag{3.38}$$

The formulation that results for an N-generator T-hour problem is the following;

$$
\begin{aligned}
d_{i,T}^{k+1} =& \arg\min_{d_{i,T}} \sum_{t\in T} \left[C_i(d_{i,t}) - \lambda_{i,t}^k d_{i,t} - c d_{i,t} \left(s_{i,t}^k - d_{i,t}^k\right) + \frac{b}{2}\left(d_{i,t} - d_{i,t}^k\right)^2 \right] \\
& s.t. \;\; d_{i,T} \in \mathcal{D}^i \qquad (3.39) \\
s_{I,t}^{k+1} =& \arg\min_{s_{I,t}} \sum_{i\in I} \lambda_{i,t}^k s_{i,t} + c s_{i,t}\left(s_{i,t}^k - d_{i,t}^k\right) + \frac{b}{2}\left(s_{i,t} - s_{i,t}^k\right)^2 \\
& s.t. \;\; \sum_{i\in I} s_{i,t} = D^t, \quad \forall t \in T \qquad (3.40)
\end{aligned}
$$

The generator or d-problems, (3.39), contain the temporal and unit-wise constraints of the UC problem. Because the feasibility of solutions is influenced by inter-period constraints Dynamic Programming is a sound solution method for this sub-problem. The values entered in the Dynamic Programming cost matrix for $C_i(d_i) = a_i + b_i d_i + c_i d_i^2$ are

$$
\begin{aligned}
d_{i,t}^{k+1} =& \frac{\lambda_{i,t}^k - b_i + c\left(s_{i,t}^k - d_{i,t}^k\right) + b d_{i,t}^k}{2c_i + b} \qquad (3.41) \\
& s.t. \;\; d_{i,t}^{k+1} \in [\underline{p_i}, \overline{p_i}] \;\; \forall t \in T
\end{aligned}
$$

The schedules returned by DP will meet all of the individual unit constraints. Although each schedule is implementable on a generator by generator basis, the final result is not likely to be feasible globally in terms of the power balance constraint, at least at first. Matching supply to demand is a static problem, static in the sense that there are no inter-period constraints. This part of the algorithm is addressed by the s-optimization, ultimately through the dual coordinating variables these constraints get reflected into the d-problems.

The s-problems (3.40) represent an independent set of optimizations with one equality constraint, one for each period of the schedule. None of the generator constraints are applied to the s-problem as these are. To obtain a feasible set of s-values at each iteration of the algorithm a sub-iteration is required. Λ_t is introduced to dualize the power balance constraint at the t^{th} hour, the following algorithm is obtained;

$$s_{i,t}^{k+1} = \frac{\Lambda_t^h - \lambda_{i,t} - c\left(s_{i,t}^k - d_{i,t}^k\right) + bs_{i,t}^k}{b} \tag{3.42}$$

$$\Lambda_t^{h+1} = \Lambda_t^h + \alpha\left(D_t - \sum_{i \in I} s_{i,t}^{k+1}\right) \tag{3.43}$$

where h is the index of sub-iterations for the static variables sub-problem. By iterating (3.42) and (3.43) to convergence, the power balance constraint at each hour for each iteration of the main problem is met by the s variables.

Finally a coordinating step is performed at the end of each major iteration to bring the s-values and d-values together. Remember there is a dual variable λ for each generator at each hour.

$$\lambda_{I,T}^{k+1} = \lambda_{I,T}^k + c\left(s_{I,T}^{k+1} - d_{I,T}^{k+1}\right) \tag{3.44}$$

As with the direct application of Augmented Lagrangian it is wise to select a value of the update step which is smaller than c, although experimentation tends to indicate this to be less critical to stability when using the variable split approach.

3.8 Convergence and Optimality

As reported in [21, 22] convergence is steadier using the variable split formulation. Sustained oscillation of the primal variables, as seen for the direct application of the Augmented Lagrangian method to the UC problem, does not seem to occur in the

Variable Split method.

Reflections on the role of the penalty term in each method explain the relative stability of the variable split approach. Penalties on the surplus or deficit in generation are applied across the board and induce all generators to react in the same direction in the direct ALR method. Decomposition can cause the generators to commit and de-commit in blocks at subsequent iterations and push the algorithm into an unstable equilibrium. In the Variable Split approach, the penalty terms are applied to individual variables and no global effect is induced.

The Variable Split Augmented Lagrangian tends to converge even for fairly large values of c regardless of the quality of the starting conditions. The trade off is that convergence to a false optimum can occur. This is illustrated in Figure 3.13 where the Variable Split algorithm converges at the 25th iteration, while the direct ALR converges at the 40th and LR is unstable. Unfortunately the variable split approach converged to the commitment vector $\{1111\}$ while the optimum is $\{1110\}$ which direct ALR managed to find. Figure 3.13 is very similar to one found in [21]. Under some conditions faster convergence may justify the adoption of the Variable Split method, particularly if adequate starting values are available. The method of Optimal Cost Curve Aggregation presented in the next chapter is a sound source of these starting values.

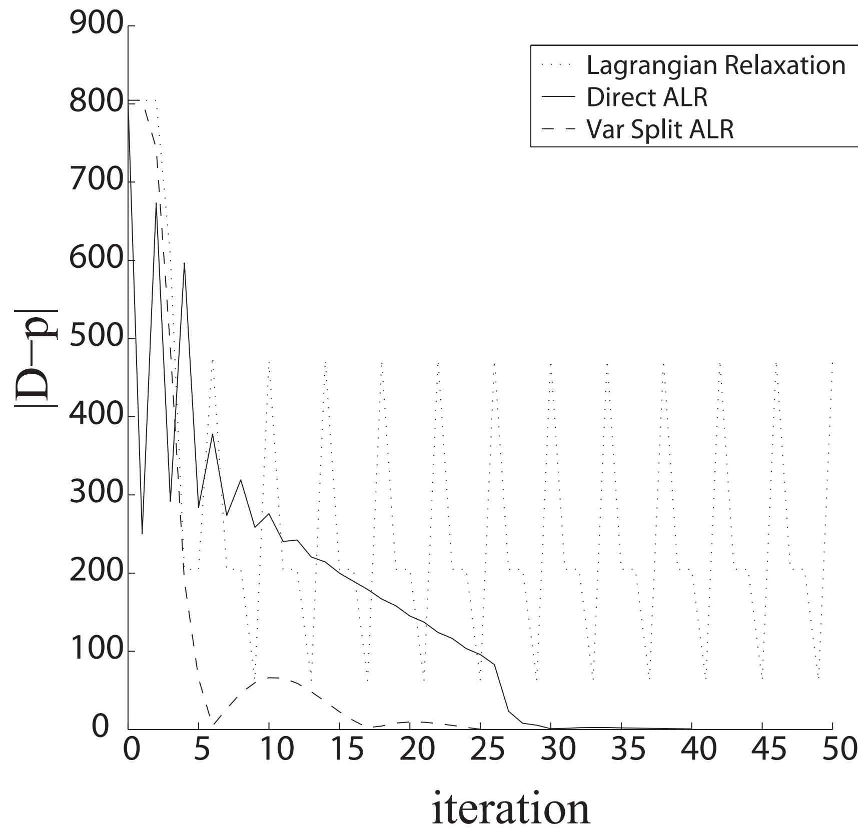

Figure 3.13: Evolution of the absolute difference between offer and demand for three algorithms, 4-unit system, $D = 804.74$, $c = .0455$

Chapter 4

Cost Curve Aggregation

Dual methods have been shown to provide a limited mapping of the least-cost curve for the Unit Commitment problem. It is clear that Lagrangian methods effectively minimize cost for parts of the feasible range, but discontinuities in the primal domain, arising from integer decisions, produce large blind-spots in this mapping. The Augmented Lagrangian methods improve the mapping but sub-optimal convergence can be a problem and the presence of discontinuities can require penalty terms that make the algorithm unstable. The Augmented Lagrangian methods also hinder the disaggregation of generator decisions.

The cost curve aggregation approach presented in this thesis returns the entire cost curve as a function of aggregate power despite the presence of discontinuities and does not require complete enumeration. The cost curve aggregation method is comparable to a sorting algorithm and is specifically adapted to the combinatorial nature of the UC problem. Furthermore, it solves the static UC problem in polynomial time.

For a single period UC problem with N generators the cost of generation is reduced to two dimensions, aggregate power and cost. This is remarkable as the original problem has N continuous dimensions and N integer decisions plus cost. Fundamentally, for two generators, an optimal dispatch occupies a limited subspace of the 3-D space available. This limited subspace is shown on the contour plot of Figure 4.1.

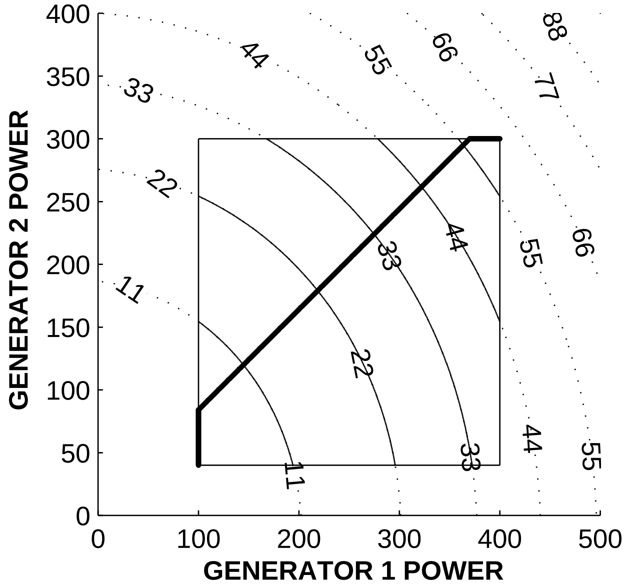

Figure 4.1: Cost contours/1000 and optimal policy (heavy) (Parameters in Table 4.1)

index	a	b	c	$\underline{p}$	$\overline{p}$
1	500	10	0.2	100	400
2	300	8	0.25	40	300

Table 4.1: Cost parameters for 2 units

Note that the optimal policy is composed of three sections each with a fixed ratio of participation between the units. The resulting cost/power curve has three sections which are quadratic functions in $p = p_1 + p_2$; the parameters for each section can be found analytically, as will be demonstrated.

4.1 Curve Aggregation Algebra

First consider the generation cost for two units,

$$F(p_1, p_2) = \sum_{i=1}^{2} a_i + b_i p_i + c_i p_i^2 \tag{4.1}$$

We define $F^*(p)$ to be the minimization of (4.1) with $p = p_1 + p_2$ and seek an analytical expression for $F^*(p)$. If power constraints do not interfere, the optimal loading of each generator is obtained when the incremental costs of the units are equal;

$$p_i = \frac{\lambda - b_i}{2c_i}, \quad i = \{1, 2\} \tag{4.2}$$

where λ is used to dualize the equality constraint $p = p_1 + p_2$ into the objective (4.1). From the same equality constraint and (4.2) one can express p as a function of λ,

$$p = \frac{\lambda - b_1}{2c_1} + \frac{\lambda - b_2}{2c_2}. \tag{4.3}$$

It follows that the cost of aggregate output as a function of λ can also be expressed. This is accomplished by substituting (4.2) into (4.1). After multiplying through and collecting terms one obtains:

$$F^*(\lambda) = a_1 + a_2 - \frac{b_1^2}{4c_1} - \frac{b_2^2}{4c_2} + \lambda^2 \left[\frac{1}{4c_1} + \frac{1}{4c_2}\right] \tag{4.4}$$

Rewrite equation (4.3) as

$$\lambda = \frac{2pc_1c_2 + b_1c_2 + b_2c_1}{c_2 + c_1} \tag{4.5}$$

and substitute (4.5) into (4.4) to obtain

$$F^*(p) = a_1 + a_2 - \frac{b_1^2}{4c_1} - \frac{b_2^2}{4c_2} + \left[\frac{2pc_1c_2 + b_1c_2 + b_2c_1}{c_2 + c_1}\right]^2 \left[\frac{1}{4c_1} + \frac{1}{4c_2}\right] \tag{4.6}$$

After collecting terms, we have the quadratic terms for the cost curve when both units participate without constraint in the dispatch, that is,

$$F^*(p) = a + bp + cp^2 \tag{4.7}$$

where,

$$a = a_1 + a_2 - \frac{b_1^2}{4c_1} - \frac{b_2^2}{4c_2} + \frac{(b_1c_2 + b_2c_1)^2}{c_1 + c_2}\left[\frac{1}{4c_1} + \frac{1}{4c_2}\right] \tag{4.8}$$

$$b = \frac{4c_1c_2(b_1c_2 + b_2c_1)}{(c_1 + c_2)^2}\left[\frac{1}{4c_1} + \frac{1}{4c_2}\right] \tag{4.9}$$

$$c = \frac{4c_1^2c_2^2}{(c_1 + c_2)^2}\left[\frac{1}{4c_1} + \frac{1}{4c_2}\right] \tag{4.10}$$

Equations (4.7) through (4.10) hold when both generators are operating between their power constraints. This range corresponds to the diagonal segment for the example in Figure 4.1. Generally, the min/max power constraints of one of the units forces saturation to one of the power constraints somewhere in the combined range of the machines. For example, the segment of the optimal policy parallel to the ordinate in Figure 4.1 shows the first unit saturating to its low limit in the lower portion of the combined range. Similarly, the segment parallel to the abscissa shows the second generator saturating to its upper limit in the upper portion of the combined range. To find the lower bound where (4.7) can be applied, rewrite (4.2) to solve for λ and keep the highest value;

$$\lambda_{low} = \max_i \left(b_i + 2c_i\underline{p_i}\right), \quad i = \{1, 2\} \tag{4.11}$$

Using λ_{low} one can proceed to evaluate the value of $p = p_1 + p_2$;

$$p_{low} = \mathcal{F}_1\left(\frac{\lambda_{low} - b_1}{2c_1}\right) + \mathcal{F}_2\left(\frac{\lambda_{low} - b_2}{2c_2}\right) \tag{4.12}$$

where $\mathcal{F}_1$ and $\mathcal{F}_2$ are shorthand for the saturation functions that correspond to the min/max generating bounds. The unit that returned λ_{low} will keep to its low constraint for $\underline{p_1} + \underline{p_2} < p < p_{low}$. To find the quadratic parameters in this constrained segment, set the appropriate unit to its limit and use $p_{nc} = p - \underline{p_c}$ where p_{nc} is the power of the unit not constrained and $\underline{p_c}$ is the binding constraint. The following cost equation applies,

$$F^*(p) = a_c + b_c(\underline{p_c}) + c_c(\underline{p_c})^2 + a_{nc} + b_{nc}(p - \underline{p_c}) + c_{nc}(p - \underline{p_c})^2 \tag{4.13}$$

for the range $\underline{p_c} + \underline{p_{nc}} < p < p_{low}$. The quadratic parameters of (4.13) can readily be identified. The same reasoning applies for generator upper limits. One first evaluates,

$$\lambda_{high} = \min_i \left(b_i + 2c_i\overline{p_i}\right), \quad i = \{1, 2\} \tag{4.14}$$

to identify what unit reaches its upper power limit first. Then the lower limit of the upper constrained segment can be found;

$$p_{high} = \mathcal{F}_1\left(\frac{\lambda_{high} - b_1}{2c_1}\right) + \mathcal{F}_2\left(\frac{\lambda_{high} - b_2}{2c_2}\right) \tag{4.15}$$

The quadratic parameters for the upper segment, $p_{high} < p < \overline{p_c} + \overline{p_{nc}}$, are then;

$$F^*(p) = a_c + b_c(\overline{p_c}) + c_c(\overline{p_c})^2 + a_{nc} + b_{nc}(p - \overline{p_c}) + c_{nc}(p - \overline{p_c})^2 \tag{4.16}$$

In conclusion, the optimal cost for two generators participating in a dispatch is summarized by the following equation:

$$F^*(p) = \begin{cases} (4.13), & \underline{p_1} + \underline{p_2} < p < p_{low} \\ (4.7), & p_{low} < p < p_{high} \\ (4.16), & p_{high} < p < \overline{p_1} + \overline{p_2} \end{cases} \quad (4.17)$$

4.2 Cost Curve Aggregation Algorithm

The cost curve aggregation algebra can be generalized into an iterative algorithm that returns the optimal cost curve and the associated commitment decisions for a set of generators. The algorithm has as many iterations as there are units to aggregate. It is therefore useful to use the unit index, i, to also denote the current iteration. Defining $F^*_{i-1}(p)$ as the optimal cost curve returned after $i-1$ iterations the algorithm is described by the following steps:

1. Evaluate the improvement, if any, of adding the i^{th} unit cost, $f_i(p)$, to the optimal cost curve $F^*_{i-1}(p)$.

   (a) Apply (4.17) to $f_i(p)$ and the first segment in $F^*_{i-1}(p)$. Compare the new curve to $F^*_{i-1}(p)$, keep only the lowest segments and save as $F_i(p)$.

   (b) Apply (4.17) to $f_i(p)$ and the second segment in $F^*_{i-1}(p)$. Compare the new curve to $F_i(p)$, keep only the lowest segments and save as $F_i(p)$.

   (c) Continue until all the segments in $F^*_{i-1}(p)$ have been tested.

2. Compare $F_i(p)$ to $f_i(p)$, retain the lowest segments to obtain $F^*_i(p)$.

3. If there are further units to add, increment i and go back to step 1.

Functionally, there are two major steps required to add a unit to the aggregate cost curve. In the first step, the cost of the new unit paired with the previous commitments is determined. This first step could be called the 'AND' step because the contribution of the new unit is considered in conjunction with the other units. The second step is to compare the cost of the unit alone to the previous commitment decisions. This second step could be called the 'XOR' step because the choice is between the unit on its own and the group. Using this reasoning, the flow of the algorithm is illustrated in Figure 4.2.

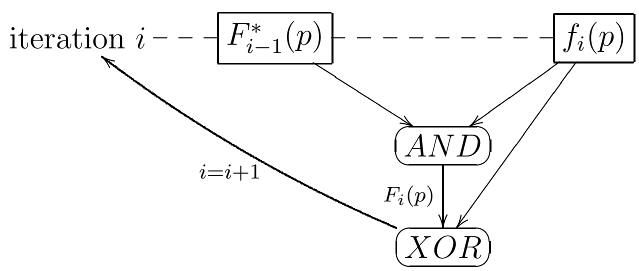

Figure 4.2: Functional representation of the cost curve aggregation algorithm.

4.3 Cost Curve Aggregation - An Example

This section demonstrates how to find the optimal cost curve as a function of aggregate output for three generators. Consider the units in Table 4.2 and their cost curves drawn in Figure 4.3.

Unit	a	b	c	$\underline{p}$	$\overline{p}$
1	40	.05	.008	10	70
2	40	.04	.007	60	110
3	45	.03	.005	70	120

Table 4.2: Three unit cost curve aggregation example

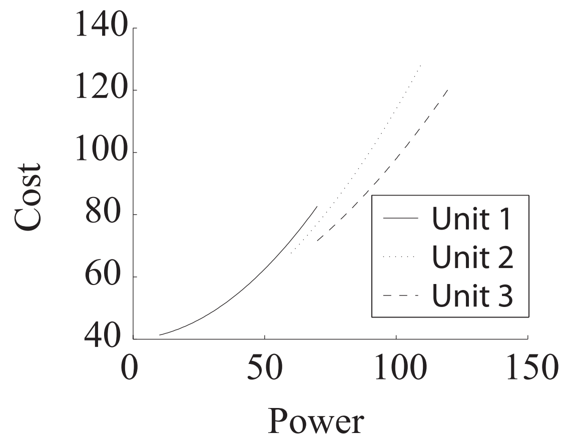

Figure 4.3: Cost curves for units in Table 4.2

The first iteration of the cost curve aggregation algorithm establishes the first unit cost as the optimal curve pending the aggregation of additional units. This aggregate of one requires no computation and is summarized by $F_1^*(p) = f_1(p)$. At the second iteration, $i = 2$, $f_2(p)$ and $F_1^*(p)$ are combined using (4.17) to obtain

$$f_1(p) \cup f_2(p) = \begin{cases} 133 - 0.910p + 0.008p^2, & 70 < p < 111.9 \\ 80.0 + 0.045p + 0.0037p^2, & 111.9 < p < 150.7 \\ 154.2 - 0.94p + 0.007p^2, & 150.7 < p < 180 \end{cases} \tag{4.18}$$

Figure 4.4 shows the cost curve of the first two units along with $f_1(p) \cup f_2(p)$. The optimal cost curve for the first two units is readily identified from Figure 4.4. After picking the lowest segments and discarding the rest, the optimal cost curve is:

$$F_2^*(p) = \begin{cases} 40.00 + 0.050p + 0.0080p^2, & 10 < p < 60 \\ 40.00 + 0.040p + 0.0070p^2, & 60 < p < 110 \\ 133.40 - 0.910p + 0.0080p^2, & 110 < p < 112 \\ 80.00 + 0.045p + 0.0037p^2, & 112 < p < 150.7 \\ 154.20 - 0.940p + 0.0070p^2, & 150.7 < p < 180 \end{cases} \tag{4.19}$$

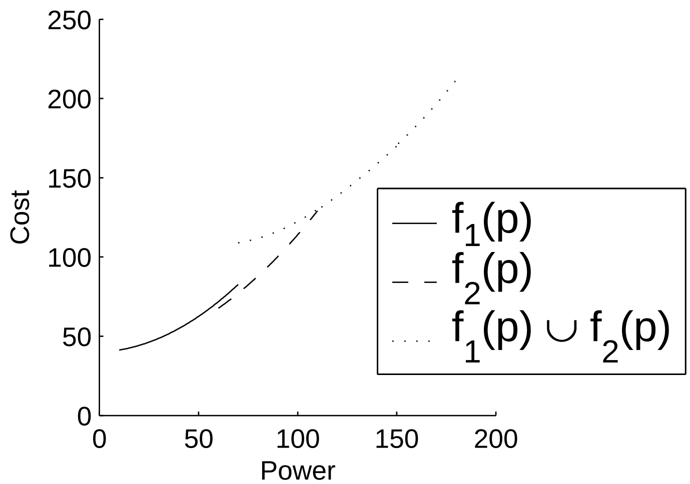

Figure 4.4: Cost curves for the first two units and $f_1(p) \cup f_2(p)$.

with the corresponding commitment decisions;

$$U(p) = \begin{cases} 0 \quad 1, & 10 < p < 60 \\ 1 \quad 0, & 60 < p < 110 \\ 1 \quad 1, & 110 < p < 180 \end{cases} \tag{4.20}$$

Having found the optimal curve for the first two units we proceed to the third iteration and to the third unit. First the 'AND' step, the relation in (4.17) is applied five times with Unit 3 and each of the quadratic segments in (4.19) as input. The lowest segments are retained to obtain $F_3(p)$. Then, in the 'XOR' step, $F_3(p)$ is compared to the cost curve of Unit 3. After having selected the lowest segments, the optimal cost curve as a function of power, $F_3^*(p)$, is obtained. The optimal curve is drawn in Figure 4.5 along with all the other possible commitment curves.

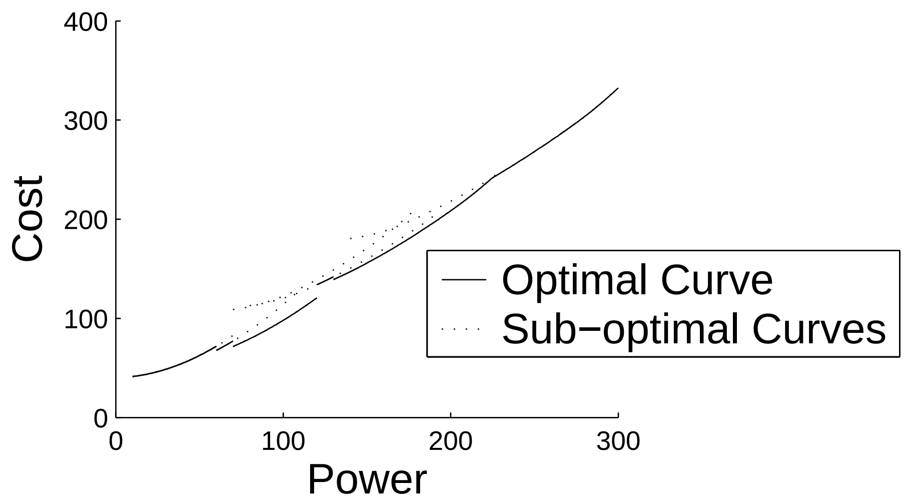

Figure 4.5: Optimal curve for the three unit example.

The optimal commitments for these three units, as can be verified using complete enumeration, are listed in (4.21). Note that the commitment $U(p) = \{011\}$ does not appear anywhere in (4.21); this combination is not optimal for any of the aggregate power levels. The significance of this observation is that this particular combination will not be used when future units are aggregated. This helps to illustrate how the aggregation algorithm can identify the optimal cost curve without the need to enumerate all possible commitment combinations. Only the 'thus far optimal' groupings are retained; this set grows much less quickly than the full set of possibilities which grows at an exponential rate. The number of commitment groups that gradually get filtered out largely accounts for the computational efficiency of the cost curve aggregation method.

$$U(p) = \begin{cases} 0 \quad 0 \quad 1, & 10 < p < 60 \\ 0 \quad 1 \quad 0, & 60 < p < 70 \\ 1 \quad 0 \quad 0, & 70 < p < 120 \\ 1 \quad 0 \quad 1, & 120 < p < 130 \\ 1 \quad 1 \quad 0, & 130 < p < 224.7 \\ 1 \quad 1 \quad 1, & 224.7 < p < 300 \end{cases} \tag{4.21}$$

4.4 Optimality and Computation Time

The optimality of the cost curve aggregation method was investigated using networks of randomly generated units. The output of the cost curve aggregation algorithm is compared to the known optimal result from complete enumeration. The parameters used to generate 'random' units are shown in Table 4.3. The distributions and ranges are selected so that the quadratic parameter is always greater than zero and the cost strictly increases with power. A series of tests, as summarized in Table 4.4 were performed on networks of varying size. Each test was performed by generating a number of random units using the distributions in Table 4.3, then solving the static UC problem using both complete enumeration and cost curve aggregation and finally then comparing the results.

Parameter	Description	Distribution	Range
a_i	Constant Term	Uniform	$0 \le a_i \le 500$
b_i	Linear Term	Uniform	$0 \le b_i \le 500$
c_i	Quadratic Term	Uniform	$10^{-6} \le c_i \le 1 + 10^{-6}$
$\underline{p_i}$	Lower Limit	Uniform	$0 \le \underline{p_i} \le 200$
$\overline{p_i}$	Upper Limit	Uniform	$\underline{p_i} \le \overline{p_i} \le \underline{p_i} + 200$

Table 4.3: Parameters for random network tests

Number of Units	Number of Tests	Number of Test Points	Running time	% Match
5	200	1000	300 seconds	100
10	100	1000	58 minutes	100
15	50	1000	22 hours	100
20	10	100	26 hours	100

Table 4.4: Summary of random tests

From the last column in Table 4.4 note that the cost curve aggregation algorithm is in perfect agreement with complete enumeration. Generator commitments in all cases were the same for either method. From review of the literature, only the cost curve aggregation method and complete enumeration appear to successfully solve the optimal static Unit Commitment problem.

The fourth column in Table 4.4 provides the run-times required to execute the tests. The bulk of the computation time is consumed by the complete enumeration algorithm. For example, the cost curve aggregation algorithm has an execution time of 1.4 seconds for a 20 unit problem on a Pentium 4, 2.66 GHz with 512 MB of RAM. This is in stark contrast to the 220 minutes it takes to run complete enumeration at a resolution of $\Delta p = 5$ and a tolerance of $p = .1$ on the same system. Furthermore, the cost curve aggregation algorithm yields more information: the full curves are returned parametrically, e.g. (4.19), rather than as sets of points along the curves.

The cost curve aggregation algorithm was experimentally observed to have a cubic incremental run-time with respect to the number of generators. The speed of the algorithm was checked by aggregating 200 randomly generated units, this took 69 minutes on the above mentioned system. The incremental computation time for this test is plotted on a semi log scale in Figure 4.6. A cubic fit is plotted alongside the computation time as a comparison.

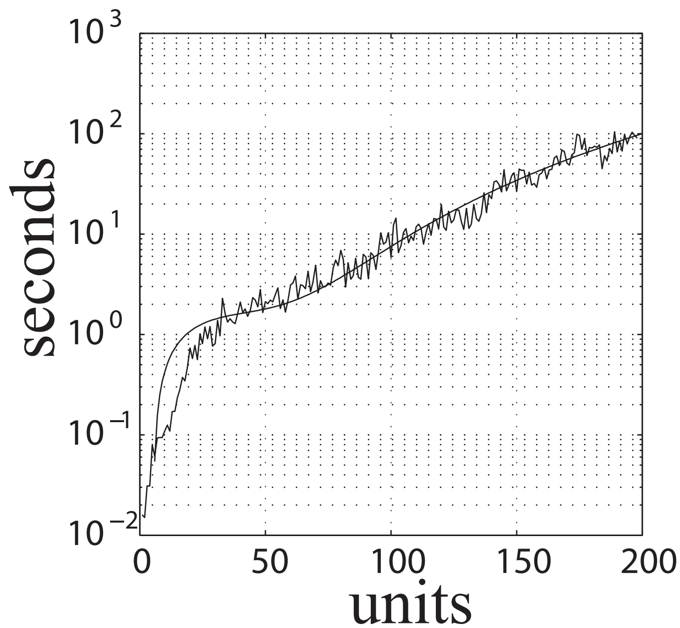

Figure 4.6: Computation time

4.5 Dynamic Programming

The cost curve aggregation algorithm compares and sorts the cost characteristics of generators; it finds the optimal grouping of units to meet a static power demand. Physical factors tie subsequent decision periods together but are not directly represented in the cost curve aggregation method. In Unit Commitment these dynamic properties express themselves as power ramping limits, minimum up- and down-time constraints, start-up costs and equipment cooling constraints. The small size of typical distributed generators and the fact that UC is typically solved for one hour or at the least half-hour scheduling blocks indicates that a suitable schedule can be obtained with cursory consideration of these factors. The same cannot be said for utility scale generators, as the thermal and mechanical inertias of the units couple the scheduling decisions over many hours. These inertias motivate the requirement for a dynamic solution. Applying dynamic programming (DP) extends the applicability of the cost curve aggregation method.

Scheduling problems are efficiently solved with dynamic programming. DP returns an optimal solution in polynomial time by taking advantage of the structure of these problems. The structure of scheduling problems is highlighted by the need to make subsequent decisions which incur costs, these can be thought of as static costs. The stages of a decision scenario are linked by transition costs which denote the cost of moving from one state to the next, which can thought of as dynamic costs. The DP formulation separates and represents these costs using the static cost matrix A and the transition cost matrix $\mathcal{T}$,

$$A = \begin{bmatrix} s_{1,1} & s_{1,2} & \dots & s_{1,T} \\ s_{2,1} & s_{2,2} & \dots & s_{2,T} \\ \vdots & \vdots & \ddots & \vdots \\ s_{N,1} & s_{N,2} & \dots & s_{N,T} \end{bmatrix}, \mathcal{T} = \begin{bmatrix} d_{1,1} & d_{1,2} & \dots & d_{1,N} \\ d_{2,1} & d_{2,2} & \dots & d_{2,N} \\ \vdots & \vdots & \ddots & \vdots \\ d_{N,1} & d_{N,2} & \dots & d_{N,N} \end{bmatrix} \tag{4.22}$$

where $s_{n,t}$ is the static cost of state n at instant t, and $d_{n,n'}$ is the dynamic cost of moving from state n to state n'. The static cost matrix has dimensions $[N, T]$ and the transition matrix has dimension $[N, N]$, where N is the number of states and T is the number of instants or decision stages. It is important when setting up the problem to use a reasonable number of time steps and states such that computational resources are not overrun. To see how these matrices interact, a graphical representation of a 3-state, 4-interval DP problem is drawn in Figure 4.7.

The nodes in Figure 4.7 each have a static, or state cost, while the edges represent transition costs. It is interesting to note that such a small example has $3^4 = 81$ possible solutions; the purpose of DP is to find the minimum cost path through a graph such as this without explicitly finding the cost of each path. The form is quite general

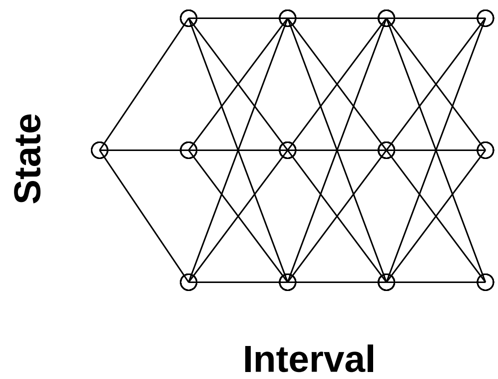

Figure 4.7: Graphical representation of a DP problem

and could potentially represent many different problems, one of them, of course, is the UC problem. Traveling sales-person as well as discrete control problems and portfolio optimization problems have a similar structure because they are each stage-based decision problems where anterior decisions affect present and future outcomes. Taking the traveling sales-person analogy we can easily imagine each node as being a city and the edges as roads between the cities. The goal is then to find the itinerary which maximizes the sales-person's overall profit starting from a given city and traversing the graph from left to right. The profit of the sales-person is calculated as the income from each city visited minus travel costs.

To illustrate the procedure, values for the DP matrices are assigned in (4.23) for the problem depicted in Figure 4.7. As a convenience, the transition matrix $\mathcal{T}$ is chosen to be constant throughout the problem, though this need not be the case, it could vary from decision interval to decision interval.

$$A = \begin{bmatrix} -2 & -4 & -2 & -4 \\ -3 & -1 & -3 & -1 \\ -4 & -1 & -2 & -6 \end{bmatrix}, \mathcal{T} = \begin{bmatrix} 1 & 2 & 4 \\ 2 & 1 & 2 \\ 4 & 2 & 1 \end{bmatrix} \tag{4.23}$$

The first step in forward dynamic programming is to find the cost of getting to each state in the first decision interval. This is obtained by summing the transpose of the second row of $\mathcal{T}$ and the first column of A to obtain costs at the first interval F_1,

$$F_1 = (\mathcal{T}^T)_2 + A_1 = \begin{bmatrix} 2 \\ 1 \\ 2 \end{bmatrix} + \begin{bmatrix} -2 \\ -3 \\ -4 \end{bmatrix} = \begin{bmatrix} 0 \\ -2 \\ -2 \end{bmatrix} \tag{4.24}$$

To find costs at the second interval F_2, we reuse F_1, the entire transition matrix $\mathcal{T}$ and the second column[1] of A ,

$$F_2^T = \min \left\{ \begin{bmatrix} F_1 & F_1 & F_1 \end{bmatrix} + \mathcal{T} + \begin{bmatrix} A_2^T \\ A_2^T \\ A_2^T \end{bmatrix} \right\} \tag{4.25}$$

$$F_2^T = \min \left\{ \begin{bmatrix} 0 & 0 & 0 \\ -2 & -2 & -2 \\ -2 & -2 & -2 \end{bmatrix} + \begin{bmatrix} 1 & 2 & 4 \\ 2 & 1 & 2 \\ 4 & 2 & 1 \end{bmatrix} + \begin{bmatrix} -4 & -1 & -1 \\ -4 & -1 & -1 \\ -4 & -1 & -1 \end{bmatrix} \right\} \tag{4.26}$$

$$F_2 = \begin{bmatrix} -4 \\ -2 \\ -2 \end{bmatrix} \tag{4.27}$$

The relation in (4.25) can be applied to the other decision stages using the recursive

[1]The notation is set up so that each column represents a minimization.

form,

$$F_{i+1}^T = \min \left\{ \begin{bmatrix} F_i & F_i & F_i \end{bmatrix} + \mathcal{T} + \begin{bmatrix} A_{i+1}^T \\ A_{i+1}^T \\ A_{i+1}^T \end{bmatrix} \right\} \tag{4.28}$$

and the least cost path taken to reach each node in Figure 4.7 are conveniently found by evaluating the argument of the minimization in (4.28), namely,

$$P_{i+1}^T = \arg\min \left\{ \begin{bmatrix} F_i & F_i & F_i \end{bmatrix} + \mathcal{T} + \begin{bmatrix} A_{i+1}^T \\ A_{i+1}^T \\ A_{i+1}^T \end{bmatrix} \right\} \tag{4.29}$$

The rest of the example is found in Appendix B, and the matrices F and P are as follows for the problem set in (4.23):

$$F = \begin{bmatrix} 0 & -4 & -5 & -8 \\ -2 & -2 & -5 & -5 \\ -2 & -2 & -3 & -9 \end{bmatrix}, \qquad P = \begin{bmatrix} 2 & 2 & 1 & 1 \\ 2 & 2 & 1 & 2 \\ 2 & 3 & 3 & 2 \end{bmatrix} \tag{4.30}$$

A value in the matrix P points to the state in the previous stage which led to the cost in A; this is shown by redrawing Figure 4.7 with these pointers and the optimal path in Figure 4.8. Note that the two matrices have as many elements as the graph has nodes, excluding the 'starting' node. Figure 4.8 has been drawn such that one could overlay the matrices over the graph to find the values associated with each node. By looking at the right-most values of F we determine that the lowest cost path terminates in the third state with a cost of -9, which represents a revenue of $+9$ in the traveling sales-person problem. The path that led to this state can be determined by looking at the lower right value of P; we find a pointer to state 2. Following this pointer, state 2 at stage 3 points to state 1 at stage 2 which then points back to state 2 at stage 1 which points to the initial state, 2. The optimal path is then $\{2, 2, 1, 2, 3\}$.

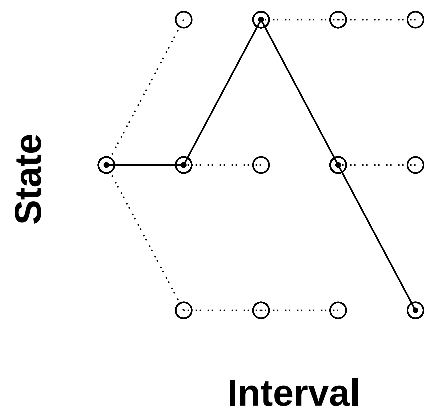

Figure 4.8: DP pointers (dashed) and optimal path (solid)

The relation in (4.28) is easily modified to accommodate more states; the matrices are modified to have dimensions conforming to the problem definition. Meanwhile, solving for additional decisions, or time intervals, simply requires more iterations. The number of iterations required is one-to-one with the number of intervals. The limiting factor in DP tends to be storage capacity and the computational expense of using this stored information as opposed to the sheer number of operations required for complete enumeration.

The trade-off between pure enumeration and DP is the need for storage. The memory requirements for complete enumeration are very modest, in fact one cannot devise a solution that requires less storage capacity. The only memory requirement is to keep track of the best candidate so far and the value it returned, while dynamic programming requires the storage of potentially large matrices. Some amount of data storage is certainly reasonable, and for problems whose structure fits DP, it or a similar formulation should be favored for all but the smallest of applications. If the problem is too large to solve using dynamic programming the number of decision intervals, or of states should be carefully reduced. For such large problems the limited storage requirements for complete enumeration would likely be of no benefit because computation times would be extremely prohibitive.

One method of applying DP to the Unit Commitment problem is to consider groupings of generators as the state variables. The cost of starting or stopping generators naturally arise as the cost of transition between commitment groups with different unit assignments. The method of cost curve aggregation is a good companion for this DP implementation, specifically because it provides a reduced set of economically advantageous generator combinations. The need to limit the states which are considered is great because the number of potential unit combinations grows exponentially with the generator count.

Cost curve aggregation provides some of the states which are likely candidates to be part of an optimal policy. In this regard it can be used to improve priority list algorithms, as it can provide commitment vectors which would not be identified in a strict priority list. However, because the optimal cost curve is drawn without taking into account the peaking ability of generators some effort to supplement the DP-states with peaking generators is required to improve the final schedule.

Although the mechanics and reasons for using DP are the same as in Section 2.2, it should be noted that the problems solved are not exactly the same. In the Lagrangian solution, price-based decomposition permits a solution to proceed from a multitude of scheduling problems involving a single unit and the set of coordinating dual variables. Single-unit scheduling is obviously easier than multi-unit scheduling. Additionally, the ability to choose periods of continuous commitment, or de-commitment, permits a very thorough representation of dynamic constraints. The limited ability of dual variables to coordinate solutions, of course, is the one shortcoming of this approach.

The problem of coordination does not arise if one does not use unit-wise decomposition. However, the need to solve a multi-unit schedule requires the formulation to be less expressive than that of LR. Some level of detail is necessarily lost, making the algorithm responsive to cooling constraints, ramping limits and prohibited starting/stopping regions requires significantly more work and complexity than for LR.

Taking generator combinations as states for DP means that the presence of minimum up- and down-time constraints can cause the returned schedule to deviate from feasible generator operating domains. The difficulty is that the formulation in use does not represent the number of instants a unit has been on or off, it only represents whether or not the unit is on. Expanding the state representation to include hours of commitment and de-commitment is a possibility, but can easily lead to an explosion in the number of states to be considered; this quickly causes the problem to become intractable.

A second solution is to retain more paths at each iteration, this is the approach used in [7]. To understand how this works, imagine sorting the matrices in (4.28) and (4.29) prior to minimization. The algorithm then retains the winners and as many runners-up as deemed reasonable. Upon termination of the DP algorithm, the set of schedules are reviewed until a feasible solution is found. The difficulty is in determining how many runners up must be kept in memory in order to return a feasible schedule.

The solution used with the cost curve aggregation method is to use a branching algorithm to try the different options to satisfy a binding constraint. For example, if a unit violates its down time constraint there are three options available:

- keep the unit off longer by changing its re-commitment time

- keep the unit off longer by changing its de-commitment time
- avoid turning off the unit altogether

These search directions are implemented in DP by making parts of the A matrix infinite and re-solving the problem.

To provide a quantitative measure of the schedules returned by this approach we compare them to published results using Lagrangian Relaxation in Appendix B. For the 20-unit, 24-hour problem in [27] the new method was 0.0378% more costly than the LR solution. The differences in cost result from different scheduling decision, the two schedules are provided in Table B.1 and Table B.2. The cost curve aggregation/DP solution is less explicit in its representation of dynamic costs, a fixed cost was used for starting and stopping but the final comparisson of the results for both methods were based on the true cooling rate functions. Because both methods seek an optimal solution, but each have their limitations it is natural that both returned different schedules. The biggest challenge for the cost curve aggregation/DP algorithm is to find and maintain enough generator combinations without out-stripping computational capacity while for the LR based methods the principal limitation is the difficulty in converging to a feasible and frugal solution.

Chapter 5

Stochastic Scheduling

The Unit Commitment problem is typically solved in deterministic form, i.e., one assumes that perfect forecasts of the upcoming system demand and generating resources are given. The spinning reserve criterion helps to ensure that the system will have adequate extra capacity to meet deviations in anticipated demand without having to radically alter the schedule on the fly. Deviations in demand are common, as the load necessarily varies from its forecast because of the requirement for lead-times between 24 and 36 hours. These long lead-times are at least in part mandated by the need to bring up large machines in a safe and orderly manner.

Low impact technologies such as wind, solar and run-of-river small hydro have very limited ability to ramp up production at the whim of the operator. These units add uncertainty to the dispatching of generators because they typically cannot guarantee their rated capacity at any one instant, so adherence to a production forecast by these intermittent generators is less than certain. In part this is because most DG-scale renewables have limited or nonexistent energy storage. The amount of power they produce is then subject to much greater uncertainty and variability than more conventional sources. This factor requires consideration in order to get the most benefit, both economical and environmental, from these technologies, such that not too much fuel is spent backing up the renewables.

Conventional technologies, whether thermal or utility-scale hydro, have massive energy reserves stored as stockpiles of fuel and in their head reservoirs. These reserves can be released as needed and as befits the cycles of human activity. The uncertainty associated with these conventional sources is therefore more or less limited to equipment contingencies. Nonetheless, the size of these units means that the loss of any single unit or of a major transmission component can have severe implications for the entire network. This then requires a large component of spinning reserve to be included to cover these rare occurrences.

The potential benefits of hedging schedules on the basis of forecast variability can be expected to be of increasing importance with growing wind and solar power usage. Hedging the schedule of the power system such that forecast deviations are considered can further improve the value of these resources. Ideally a schedule should limit the need to bring up expensive peaking generators in the case of wind or insolation deficits while not carrying too much unneeded capacity in the nominal or surplus case. Forecasts of wind speed and cloud cover help to anticipate production but the correlation between these phenomena and power demand cannot be neglected. The nature of the correlation depends on season, climate and load breakdown.

The load and generation forecasts are critical component to the success of integrating significant intermittent resources into any energy system. However, these forecasts are complex and must take into account factors as diverse as human behavior and climatology. It is therefore not a trivial effort to improve one of these forecasts. Applying a stochastic methodology promises to make the most of a given forecast by improving the schedule without necessarily requiring an improvement of the forecast itself. This is possible, because the optimization is based and hedged on many possible outcomes instead of simply on the forecast's nominal outcome.

5.1 Accommodating Uncertainty

The UC model can be modified to accommodate uncertainty in demand and supply as follows,

$$J = \min \quad E\left[\sum_{i\in I}\sum_{t\in T} C_i(p_{i,t})\right] \tag{5.1}$$
$$\text{subject to} \quad \sum_{i\in I} p_{i,t} = d(t) - r(t) \ , \forall t \in T \ , \ p_{i,t} \in \{0\} \cup [\underline{p_i}, \overline{p_i}]$$

where the two stochastic time series, $d(t)$ and $r(t)$, represent hourly forecasted demand and renewable energy supply respectively. Also note the addition of the expectation operator, $E[\cdot]$, in the objective function.

The reliability of the forecast will be most critical when high demand coincides with low renewable supply and visa versa. This would indicate a pronounced negative correlation between $d(t)$ and $r(t)$; further spinning reserves may be required to hedge the schedule in this case. In the case of a positive correlation between $d(t)$ and $r(t)$, the forecast reliability is improved on average, because the variation of one tends to cancel the other. One oft quoted case of positive correlation between demand and renewable production is that of high winds increasing the energy consumption through convective cooling while also spurring wind turbine output. An example of a negative correlation would be, for instance, photovoltaic surpluses being met by decreased demand from heating loads.

The correlation between demand and renewable output is recognized as a factor in system design and judicious decisions can influence this to some degree. For example, the amount of installed capacity as well as geographical and technological diversity of the intermittent resources are important considerations in limiting variability. For the purpose of this exercise, we assume that sufficient information permits a reasonable estimate of the statistical distributions f_d and f_r and their correlation $\gamma_{d,r}$. The net demand can then be forecasted, $\hat{D}_t = E[d(t) - r(t)]$, where $\hat{D}_t$ is the stochastic equivalent of D_t.

Most of the existing algorithms to solve the stochastic UC problem use unit-wise decomposition, typically Lagrangian Relaxation is involved. One outcome of this choice is that forecast variability must be approximated by actual realizations of the stochastic process. One can think of each of these realizations as a scenario. There are essentially an infinite number of scenarios that can be generated and the more of these that are considered, up to a point, the better the overall schedule can be expected to be on average.

An example of this approach is the method adopted in [17] to incorporate stochastic aspects to the UC problem. It is based on solving decomposed stochastic optimization problems; the Lagrangian approach is used to permit unit-wise decomposition. Random scenario trees are cast to fit the statistical properties of the system; deviations in demand as well as single unit failures are considered. The scenario trees form explicit representations of foreseeable outcomes and the Augmented Lagrangian method is applied to improve convergence. Improved convergence is sought because considering more outcomes necessarily adds to the computational demands of the algorithm. The schedule which minimizes cost over a finite subset of realizations is selected.

Similarly, the authors in [14] propose a stochastic solution based on 'scenario analysis', a general approach introduced by Rockafellar and Wets [28]. Uncertainty is modeled by a set of deterministic subproblems in the hopes of obtaining a well hedged solution to the original problem. The policy adopted must be such that if two outcomes A and B are foreseeable at time t and it cannot be distinguished which outcome will occur, then the decision made at t must be the same. In other words, there must be a single policy that maximizes utility for both outcomes. These scenarios are forced to a common solution, so called bundle constraints are introduced so that over the course of iterations the solutions of the separate deterministic problems converge to a common policy. The form introduced in [14], which is in fact very similar to that used in [15, 16, 17], notational preferences aside[1], is;

$$\begin{aligned} &\min_{x,u} \sum_{s=1}^{S} P_s \sum_{i\in I} \sum_{t\in T} f_i(x_t^{i,s}, u_{t-1}^{i,s}, u_t^{i,s}) \qquad (5.2) \\ \text{subject to} \quad &\sum_{i\in I} x_t^{i,s} = D_t^s, \quad \forall s \in S \\ \text{and} \quad &u_t^{i,s_1} = u_t^{i,s_2} = \ldots = u_t^{i,S}, \quad \forall i \in I, \forall t \in T \end{aligned}$$

The objective function in (5.2) is the sum of statistically weighted deterministic UC objectives. The usual equality constraints on power/demand are applied and a new set of equality constraints is added. These new constraints, $u_t^{i,s_1} = u_t^{i,s_2} = \ldots = u_t^{i,S}$, are the bundle constraints. The bundle constraints require that the ultimate commitment decisions be the same for all the scenarios so that a single policy is returned and not one for each scenario. Two decomposition steps can be applied to solve (5.2). The first dualizes the bundle constraint so that each scenario can be solved in near isolation. The second decomposition is the usual power balance dualization typically used in

[1] $f_i(x_t^{i,s}, u_{t-1}^{i,s}, u_t^{i,s})$ is the i^{th} generator cost, $x_t^{i,s}$ the i^{th} unit loading, s denotes a scenario in the set S, the u values represent the 0/1 commitment decisions and P_s is the statistical weighting of scenario s.

Lagrangian Relaxation. The authors in [14] are careful to note that the objective (5.2) is not convex, therefore convergence to a global optima is not certain. They go on to add,

> In order to obtain a policy that is of practical use, we need to provide the progressive hedging algorithm with scenarios that truly reflect all possible future demands. Furthermore, the probabilities assigned to these scenarios must be calculated carefully. Clearly, this is not an easy task. More research is needed in this area to develop better understanding of the demand randomness and the related factors. One thing which must be considered in regards to scenario generation is that the more scenarios are created the better the hedging policy of the algorithm. On the other hand, the execution time of the algorithm grows rapidly as the number of scenarios included increases and as their demands are more diverse.

Using a UC solution without unit-wise decomposition largely avoids the need to generate distinct scenarios. The requirement of a vastly expanded optimization is therefore avoided. The cost curve aggregation algorithm returns the parameters of the primal cost curves instead of individual points along these curves. This is helpful when calculating the expected cost of generation. For example, the expected cost for a given cost function, $C(p)$ and a uniform distribution function of power demand, f_p, with bounds $\begin{bmatrix} \underline{D} & \overline{D} \end{bmatrix}$ can be expressed:

$$E[C] = \int_{\underline{D}}^{\overline{D}} C(p) f_p dp \tag{5.3}$$

The expected cost obtained using (5.3) is the same as what one would obtain by running an arbitrarily large number of scenarios. The choice of the statistical distribution function will determine if the integral can be evaluated analytically. Nonetheless, numerical integration can be applied to any of the potential distributions. The advantages of the primal formulation in solving the stochastic problem are therefore retained.

The Dynamic Programming formulation for the primal problem requires only slight modifications to account for demand uncertainty. Only the cost matrix need be modified; for a T-hour problem with N unit combinations as candidates the cost matrix is

$$A = \begin{bmatrix} E[C_{1,1}] & E[C_{1,2}] & E[C_{1,3}] & \ldots & E[C_{1,T}] \\ E[C_{2,1}] & E[C_{2,2}] & E[C_{2,3}] & \ldots & E[C_{2,T}] \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ E[C_{N,1}] & E[C_{N,2}] & E[C_{N,3}] & \ldots & E[C_{N,T}] \end{bmatrix} \tag{5.4}$$

where $E[C_{1,2}]$ is the expected cost of the first commitment group at the second hour. This expected cost is obtained by integrating the product of the statistical weighting function of demand at the second period and the cost function of the first generating group.

Even a simple statistical distribution has the potential to improve the UC load model. A typical UC optimization interprets the load forecast as a set of steps in demand. This is necessarily a rough approximation mandated by the desire to keep the model manageable. Conceptually a uniform f_p for each hour smooths the steps into ramps prior to optimization, which may in fact be more realistic.

5.2 Application of Stochastic Scheduling

Consider the case of a market participant with the twenty conventional generators in [27]. The market participant also disposes of a given ratio of installed intermittent generation, say 20%. For the day ahead, the hourly marginal cost of power is anticipated or given. The market participant is then asked to submit it's aggregate power contribution for each period and will be charged an imbalance penalty if the actual contribution falls short of the scheduled quantity. The imbalance penalties used for the present example are represented by a 'dummy' unit that has unlimited generating capacity but that is much more expensive than the real generators. The cost function for the dummy unit is provided in Table B.5. The day ahead marginal prices are found in Table B.4.

The problem of formulating an optimal bid can be expressed as

$$\begin{aligned} \min_{p_{i,t},\mathcal{P}_t} \sum_{i \in I} \sum_{t \in T} f(p_{i,t}) - \lambda_t \left(\mathcal{P}_t\right) \qquad (5.5) \\ \text{subject to} \quad \sum_{i \in I} p_{i,t} + r(t) = \mathcal{P}_t, \forall t \in T \end{aligned}$$

where $\mathcal{P}_t$ represents the power bid for the t^{th} instant. The problem in (5.5) is also subject to the usual unit-wise constraints. The aggregate contribution from the intermittent units, $r(t)$, is a time series forecast. The first stage of the problem is to choose the market bid, $\mathcal{P}_t$, to optimize expected revenue. The power balance constraint is then applied in real-time to match the delivered power to the market offer. Deviations from the forecast require the conventional generators to be adjusted to regulate the supply to the original bid based on the actual performance of the intermittent units.

For the purposes of the example, the time series forecast, $r(t)$, is chosen to be a second-order auto regressive exogenous (ARX) model. The ARX model is a suitable, if basic, representation. It reflects inter-period correlation but also the non-deterministic nature of the problem; the form used is

$$x_t = a_1 x_{t-1} + a_2 x_{t-2} + e_t \tag{5.6}$$

$$r(t) = B + x_t \tag{5.7}$$

where a_1, a_2 are the AR parameters, x is an intermediate variable and B a bias used to compute the forecast. The forecast error, e_t, is a continuous random variable with an expected value of zero and a normal distribution. The parameters used to generate $r(t)$ for the example are summarized in Table B.3. Using the estimated standard deviation, σ_e, and the time series equation in (5.6) the confidence bands for a forecast of this form can be evaluated. The resulting forecast with five realizations is shown in Figure 5.1, the dashed lines show the bounds of integration used by the stochastic algorithm. The bounds were obtained by setting the exogenous term, e_t, to plus or minus one standard deviation.

To study the benefits of a stochastic methodology, three algorithms are compared using Monte-Carlo trials. The algorithms are:

1. deterministic scheduling based on the Lagrangian Relaxation method and expected output from the intermittent units,

2. deterministic scheduling based on the Lagrangian Relaxation method with a perfect forecast, and

3. primal/stochastic approach based on the aggregate cost curves, without price-based decomposition.

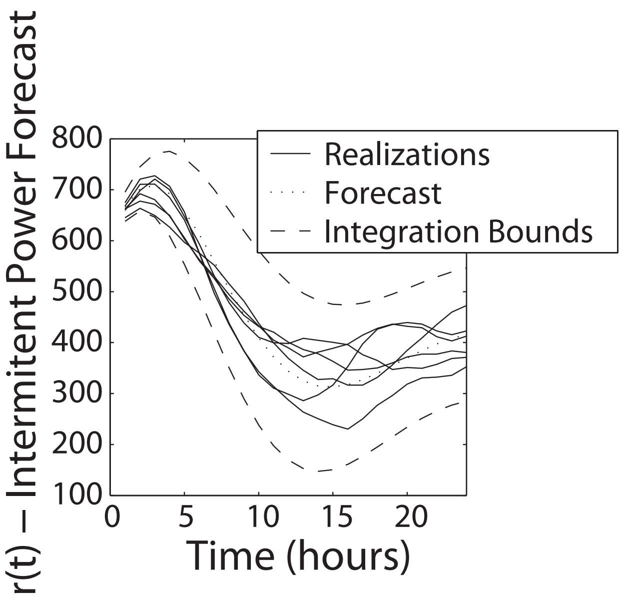

Figure 5.1: Time series forecast, 5 realizations and integration bounds.

The first algorithm, based on the Lagrangian approach, is to use the day-ahead prices to solve a set of dynamic problems, one for each conventional unit. By then summing the overall controllable output with forecasted production from the intermittent units the day-ahead energy bids are obtained. Given a perfect forecast of the intermittent contribution the dual solution for this bid-based problem is optimal and can be used to evaluate the expected value of perfect information, the second item in the preceding enumeration. The last algorithm uses the aggregate cost/power curves for a selected set of generator groupings. The optimal bids are based on the aggregate costs for each generator grouping and on the imbalance penalty. Dynamic Programming is used to obtain a feasible, profit maximizing, multi-hour schedule.

The Lagrangian Relaxation method has a tendency to schedule generators close to their rated output to offset the fixed costs of generation. In our example the deterministic solution uses the forecasted contribution directly. This can lead to a poorly hedged strategy because very little regulating margin is provided. Intermittent production below the forecast can easily lead to an inability to deliver the scheduled quantity and to the imbalance penalty being imposed. A more conservative option would be to set $\mathcal{P}_t$ to the production of the conventional units with zero contribution expected from the intermittent units. The conventional units are then backed off to accommodate the renewables. The cost associated with this policy is a lost opportunity to sell more when prices are favorable and the intermittent component is present. However, this conservative policy faces very little risk of being charged the imbalance penalty. Obviously, the optimal bid factoring in the intermittent component is somewhere in between these two extremes and will largely depend on the severity of the imbalance penalty.

Although unit-wise decomposition greatly facilitates the selection of the generator schedules it is of no use in optimizing the regulating margin. As in the previous discussion of Lagrangian Relaxation applied to the stochastic Unit Commitment problem, the only option to obtain a well hedged schedule is to generate a representative number of scenarios and, based on these, to experimentally determine the optimal regulating margin. Alternatively, if unit-wise decomposition is not used, one can directly formulate a bid with stochastic considerations without the need for a scenario-based analysis. First, consider (5.5) rewritten for a single hour, single unit minimization;

$$J(\mathcal{P}) = \min_{p} a + bp + cp^2 - \lambda \mathcal{P} \qquad (5.8)$$
$$\text{subject to} \quad p + r = \mathcal{P}$$

Note that (5.8) represents a single generator, but if we generalize to a piece-wise quadratic function the aggregate function of many units can be represented. The algebra in Appendix A is a convenient way of calculating the piece-wise quadratic parameters of the aggregate power/cost function for a given commitment group. The expected net cost of $J(\mathcal{P})$ is

$$E[J(\mathcal{P})] = \int_{\mathcal{P}-\overline{r}}^{\mathcal{P}-\underline{r}} f_p \left(a + bp + cp^2\right) dp - \lambda\mathcal{P} \tag{5.9}$$

where $\underline{r}$ and $\overline{r}$ are respectively the smallest and largest foreseeable stochastic contributions and f_p is the statistical weighting function for the power required from the conventional units to regulate the supply to the original bid. For values of $\mathcal{P}$ which cause the range $[\mathcal{P}-\overline{r} \quad \mathcal{P}-\underline{r}]$ to span more than a single section of a piece-wise-quadratic function,(5.9) is re-written,

$$\begin{aligned} E\left[J(\mathcal{P})\right] = &\int_{\mathcal{P}-\overline{r}}^{\overline{p_i}} f_p \left(a_1 + b_1 p + c_1 p^2\right) dp + \int_{\underline{p_2}}^{\overline{p_2}} f_p \left(a_2 + b_2 p + c_2 p^2\right) dp \\ &+ \ldots + \int_{\underline{p_i}}^{\overline{p_i}} f_p \left(a_i + b_i p + c_i p^2\right) dp + \ldots \\ &+ \int_{\underline{p_n}}^{\mathcal{P}-\underline{r}} f_p \left(a_n + b_n p + c_n p^2\right) dp - \lambda\mathcal{P} \end{aligned} \tag{5.10}$$

The last integration in (5.10) can be made to factor in the imbalance penalty, the boundary conditions of $E\left[J(\mathcal{P})\right]$ are then fully formed.

As a simplifying assumption, the stochastic algorithm for the present example assumes a uniform distribution over $[\mathcal{P}-\overline{r} \quad \mathcal{P}-\underline{r}]$. This is an approximation of the normal distribution used to generate the $r(t)$ time series for the Monte Carlo trials. Incremental gains in the expected profit of the stochastic algorithm could be realized by evaluating (5.10) numerically with the normal distribution. Using an alternate distribution shows the effectiveness of the stochastic approach despite modeling uncertainty in the power forecast, an inherent fact in real-world applications.

Under the approximation of a uniform distribution, the first differential of (5.10) with respect to $\mathcal{P}$ is

$$\frac{dE\left[f(\mathcal{P})\right]}{d\mathcal{P}} = f_p\left[a_n - a_1 + b_n(\mathcal{P} - \underline{r}) - b_1(\mathcal{P} - \overline{r}) + c_n(\mathcal{P} - \underline{r})^2 - c_1(\mathcal{P} - \overline{r})^2\right] - \lambda \tag{5.11}$$

By equating the above to zero and solving for the values of $\mathcal{P}^\star$ and evaluating $E\left[J(\mathcal{P}^\star)\right]$ the optimal bid can be determined. The intersections of piece-wise cubic sections should also be evaluated as they could also be the optimal point. Note that the terms in all but the first and last integrations in (5.10) are eliminated after differentiation. This convenient result is associated with the choice of a uniform distribution, the integrations away from the boundaries are not functions of $\mathcal{P}$ and are therefore eliminated following differentiation.

From this result, the bid which minimizes expected net cost can be computed for a given commitment group. The cost matrix for Dynamic Programming is filled in with $E[J(\mathcal{P}^\star)]$ for each commitment group at each hour and the same transition matrix as in the deterministic UC form is used. The choice of the generator combinations to be included can be based on the cost curve aggregation method and also on the dual solution. This choice provides generator groupings that are optimal in the static sense, from the cost curve aggregation method, good dynamic combinations are returned by the dual solution.

The schedules obtained under the Lagrangian/deterministic and primal/stochastic algorithms are adjoined in Appendix B. Figure 5.2 shows the sample average expected profit as a function of the Monte Carlo iteration for 200 iterations. This plot confirms that scheduling with perfect information yields the highest expected profit. Although this result cannot be achieved if the problem is truly stochastic, it helps gauge the success of the algorithms and demonstrates the incremental gains that could result from improving the forecast.

Over the 200 trials, the expected increase in net profit from a perfect forecast is estimated at roughly 4.9% over the deterministic solution. The improvement from the stochastic solution is in the order of 2.6% over the deterministic approach. The aforementioned tendency of generators to get scheduled close to their operating limits by the Lagrangian method resulted in the imbalance penalty being imposed fairly often on the deterministic algorithm. The bid quantity is optimized in the stochastic approach and objectively balances the opportunity to reap higher profits with the risk of incuring the imbalance penalty. The improvement in expected profit from the stochastic solution is a little over half of the prospective improvement from a perfect forecast.

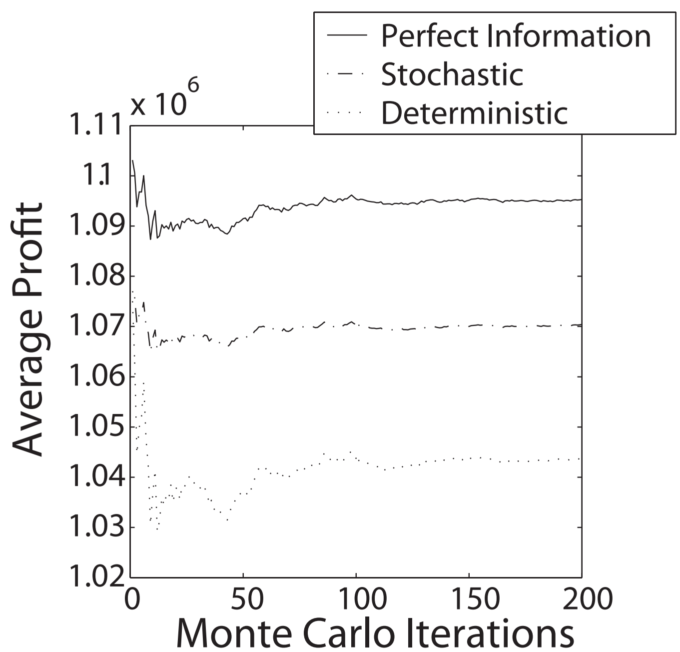

Figure 5.2: Sample mean of profit as a function of Monte Carlo iteration.

Chapter 6

Conclusions

The scheduling of electrical generators with a special consideration of distributed generation was studied. Distributed generation provides specific challenges to the dispatch of electrical energy, the main complication being the potential for a dramatic increase in the number of generators to be dispatched. Another difficulty stems from the inclusion of generators with a take-or-leave power characteristic such as wind turbines and photo-voltaic panels.

The Lagrangian Relaxation algorithm is widely applied to the scheduling of electric generators. This approach was implemented and studied in detail. It is based on the disaggregation of generator decisions through the use of coordinating dual variables that are analogous to costs. Limited convergence of the Lagrangian Relaxation algorithm was observed as in other research. Poor convergence stems from the discontinuities in the objective function of the Unit Commitment problem. The discontinuities are a direct result of the fixed cost that must be assumed to keep a generator on-line.

It remains that the Lagrangian Relaxation algorithm is very capable at incorporating generator dynamic constraints into its formulation. In large part this is because it makes use of generator-wise decomposition which yields single unit multi-hour problems which are readily solved using dynamic programming. However, the limited inter-period coupling of decisions inherently encountered for small machines largely minimizes this advantage for scheduling of distributed generators.

Two extensions to the classic Lagrangian Relaxation algorithm, in the form of the Augmented Lagrangian Relaxation, have been implemented and analyzed.

The first such method adds weight to the power balance constraint by including a quadratic penalty term in the objective. The discontinuities in the objective become locally smooth for a large enough penalty, thus convergence becomes possible. The penalty term makes decomposing the problem into individual generator sub-problems more difficult but does not preclude this option. The nature of the discontinuities in the UC problem make scaling the quadratic penalty difficult as arbitrarily large penalties may be required to make the function locally smooth. These large penalties eventually impede convergence of the algorithm.

The second Augmented Lagrangian Relaxation method studied is the Variable Split Augmented Lagrangian algorithm. In this approach, dynamic constraints and static constraints, which normally bind the problem variables simultaneously are applied separately to new independent variables. The new independent variables are then coordinated through a set of dual variables and of quadratic penalty terms so that the variables eventually converge to a common state meeting both static and dynamic constraints. This approach was found to have improved convergence properties over the other two LR methods, although it proved to be prone to converging to false optima.

A novel solution method to the Unit Commitment problem, the cost curve aggregation method, has been developed. This new algorithm was shown experimentally to return the optimal solution to the static Unit Commitment problem in polynomial time. The optimality of the method was demonstrated using randomly generated networks of up to twenty generators. The cost curve aggregation method provides a listing of the lowest cost generator combinations and is not hindered by the discontinuities in the UC problem.

The cost curve aggregation algorithm does not directly incorporate inter-period costs and constraints. This is much less of a problem in distributed generator scheduling than for conventional units. A key factor which justifies this assumption is the limited energy and time required to start and stop the smaller units. In comparison, significant energy is stored and dissipated when conventional thermal units are brought up to generate, then left to cool waiting for the next peak period. Not only is it wasteful to rapidly change the state of conventional generators it is also often infeasible, this constraint can also be present for distributed generators but, generally, on a much reduced time scale.

The applicability of the cost curve aggregation algorithm to conventional generator scheduling has been investigated by combining it with dynamic programming. The cost curve aggregation method combines well with dynamic programming applications without unit-wise decomposition because it provides much needed least-cost unit combinations. It should be noted that because the cost curve aggregation algorithm is basically a static method it does not return all the optimal unit combinations required to minimize cost for a dynamic power demand. Other approaches should be used to increase the number of unit combinations considered, as discussed.

Finally, the stochastic Unit Commitment problem was considered and solved without resort to unit-wise decomposition, in contrast with previous implementations. The advantage of the proposed approach is that it does not require discrete scenarios to model forecast uncertainty. The methodology was demonstrated on an optimal bid problem with intermittent generation included. The stochastic algorithm was shown to outperform significantly a deterministic algorithm that was otherwise optimal but that did not factor in forecast uncertainty.

The new algorithm, unlike the three others studied does not resort to an iterative search and therefore convergence is never an issue. Of course, the one limitation of the three Lagrangian methods is convergence, despite that the Augmented methods were introduced to alleviate this they are not panacea. It remains that the generator-wise decomposition that the Lagrangian methods affords puts these algorithms in very good standing for solving UC problems with complex dynamic constraints especially when compared to the cost curve aggregation algorithm.

However, the cost curve aggregation algorithm appears to be the first optimal, sub-exponential complexity algorithm to successfully solve the static Unit Commitment problem. Although this is not a complete solution to the generalized UC problem it represents an interesting direction to take in future research on the topic, particularly in the case of DG scheduling. The approach meshes well with the need to dispatch distributed generators because it provides a single, aggregated cost representation of many units. The de-emphasis on transient costs and restrictions, such as start-up and must run periods fits well with DG because these issues are much less prevalent for these machines when compared to the larger units.

Bibliography

[1] George E.P. Box, Gwilym M. Jenkins, Gregory C. Reinsel, *Time Series Analysis, Forecasting and Control*, Prentice Hall, 1994.

[2] Alex D. Papalexopoulos, Timothy C. Hesterberg, *A regression-based approach to short-term system load forecasting*, IEEE Transactions on Power Systems, Vol. 5, No. 4, November 1990, pp. 1535-1550.

[3] D.C. Park, *Electric Load Forecasting Using an Artificial Neural Network*, IEEE Transactions on Power Systems, Vol. 6, No. 2, 1991, pp. 442-449.

[4] Saied S. Sharif, James H. Taylor, *Real-Time Load Forecasting by Artificial Neural Networks*, Proceedings IEEE Power Engineering Society Summer Meeting 2000.

[5] Gregor Giebel, *The State-Of-The-Art in Short-Term Prediction of Wind Power, A Literature Overview*, http://anemos.cma.fr/, 2003.

[6] Gregor Giebel et al., *Wind Power Prediction using Ensembles*, Risø National Laboratory, Roskilde Denmark, September 2005.

[7] Allen J. Wood, Bruce F. Wollenberg, *Power Generation Operation & Control* John Wiley & Sons, 1984.

[8] Narayana Prasad Padhy, *Unit Commitment - A Bibliographical Survey*, IEEE Transactions on Power Systems, Vol. 19, No. 2, May 2004.

[9] F.N. Lee, *Short-term unit commitment - a new method*, IEEE Transactions on Power Systems, vol. 3, pp. 421-428, May 1988.

[10] John A. Muckstadt, Sherri A. Koenig, *An Application of Lagrangian Relaxation to Scheduling in Power Generation Systems*, Operations Research, Vol. 25, No. 3, pp. 387-403, May-June 1977.

[11] Jonathan F. Bard, *Short-Term Scheduling of Thermal-Electric Generators Using Lagrangian Relaxation*, Operations Research, Vol. 36, No. 5, pp. 756-766, September-October 1988.

[12] Ross Baldick, *The Generalized Unit Commitment Problem*, IEEE Transactions on Power Systems, Vol. 10, No. 1, February 1995.

[13] S.J. Wang et al., *Short-Term Generation Scheduling with Transmission and Environmental Constraints using an Augmented Lagrangian Relaxation*, IEEE Transactions on Power Systems, Vol. 10, No. 3, August 1995.

[14] Samer Takriti, John R. Birge, Erik Long, *A Stochastic Model for the Unit Commitment Problem*, IEEE Transactions on Power Systems, Vol. 11, No. 3, August 1996.

[15] U. Aytun Ozturk, Mainak Mazumdar, Bryan A. Norman,*A Solution to the Stochastic Unit Commitment Problem Using Chance Constrained Programming*, IEEE Transactions on Power Systems, Vol. 19, No. 3, August 2004.

[16] U. Aytun Ozturk, *The Stochastic Unit Commitment Problem: A Chance Constrained Programming Approach Considering Extreme Multivariate Tail Probabilities*, PHD thesis at U. of Pittsburgh, 2003.

[17] P. Carpentier, G. Cohen, J.-C. Culioli, A. Renaud, *Stochastic Optimization of Unit Commitment: a New Decomposition Framework*, IEEE Transactions on Power Systems, Vol. 11, No. 2, May 1996.

[18] Chao-an Li, Raymond B. Johnson, Alva J. Svoboda, *A New Unit Commitment Method* IEEE Transactions on Power Systems, Vol. 12, No. 1, February 1997.

[19] Nigel Lucas, Peter Taylor, *Structural deficiencies in electricity pricing in the pool*, Utilities Policy, pp. 72-76, January 1994.

[20] Marcelino Madrigal, Victor H. Quintana, *Existence and Determination of Competitive Equilibrium in Unit Commitment Power Pool Auctions: Price Setting and Scheduling Alternatives*, IEEE Transactions on Power Systems, Vol. 16, No. 3, August 2001.

[21] Arnaud Renaud, *Daily Generation Management at Electricité de France: From Planning Towards Real Time*, IEEE Transactions on Automatic Control, Vol. 38, No. 7, July 1993.

[22] J. Batut, A. Renaud, *Daily Generation Scheduling Optimization with Transmission Constraints a New Class of Algorithms*, IEEE Transactions on Power Systems, Vol. 7, No. 3, August 1992.

[23] R. Tyrrell Rockafellar, *A Dual Approach to Solving Nonlinear Programming Problems by Unconstrained Optimization*, Mathematical Programming 5, pp. 354-373, 1973.

[24] R. Bellman, A. O. Esogbue, I. Nabeshima, *Mathematical Aspects of Scheduling & Applications*, Pergamon Press 1982.

[25] Guy Cohen, *Optimization des Grands Systèmes*, Notes de cours, CERMICS-École Nationale des Ponts et Chaussées, 2004.

[26] Guy Cohen, *Optimization by Decomposition and Coordination: A Unified Approach* IEEE Transactions on Automatic Control,Vol. AC-23, No. 2, April 1978.

[27] A.B. Bakirtzis, C.E. Zoumas, *Lambda of Lagrangian Relaxation Solution to Unit Commitment Problem*, Generation, Transmission and Distribution,IEE Proceedings-, Vol. 147, No 2, pp. 131-136, March 2000.

[28] R.T. Rockafellar, J.B. Wets, *Senarios and policy aggregation in optimization under uncertainty*, Mathematics of Operations Research, 16(1):119-147, 1991.

[29] B.C. Ummels, M. Gibescu, W. L. Kling, G. C. Paap , *Integration of wind power in the liberalized Dutch electricity market*, Wind Energy, Volume 9, Issue 6, November/December 2006.

[30] Gerard B. Sheble, George N. Fahd, *Unit Commitment Literature Synopsis*, IEEE Transactions on Power Systems, Volume 9, No. 1, February 1994.

Appendix A

Analytic Economic Dispatch

To obtain solution to the stochastic Unit Commitment problem without unit-wise decomposition, the analytic cost functions of generator groupings are useful. The cost curve aggregation approach itself provides part of these cost functions, but necessarily, it does not retain all the cost information required, this is a pragmatic feature of the algorithm as retaining more information than strictly necessary to return the optimal curve would needlessly deplete computational resources. To populate the matrix in (5.4) it becomes necessary to evaluate generator combinations which are not necessarily optimal in the static sense considered by the cost curve aggregation approach.

The ED problem is to determine the cost-optimal loading of each unit in a generating group; the commitment decisions are set, which vastly simplifies the solution compared to the UC problem. Generally, to solve an economic dispatch problem an iterative solution is used to minimize generation cost. The same decomposition strategy is used as in the Lagrangian Relaxation solution for UC. The solution is fast and does not suffer from any convergence problems, as the problem domain is convex and smooth specifically because on/off decisions are not within the scope of the problem.

A typical ED solution only returns the cost at a single point along the loading curves of the generators. The analytical version is composed $2N + 1$ parabolas where the number of generators in the dispatch is N. The approach certainly uses more memory and is slightly more involved than a classic ED but the additional information can be used to advantage in stochastic scheduling problems.

In order to draw the ED curve, we fall back to the same paradigm as for a classic ED. For a fixed commitment we separate the generator subproblems by dualizing the power balance constraint. Following this, we can find for what price the generator moves out of its low constraint and also for which value it reaches its high constraint;

$$\lambda_{i,low} = 2c_i\underline{p_i} + b_i \tag{A.1}$$

$$\lambda_{i,high} = 2c_i\overline{p_i} + b_i \tag{A.2}$$

from 4.2 with $p = \underline{p_i}$ and $p = \overline{p_i}$.

For a given value of λ we can separate the generators into three sets, low constrained I_{LC}, not constrained I_{NC} and high constrained I_{HC}. All the generators in the dispatch are represented without duplication in the union of these three sets $I = I_{LC} \cup I_{NC} \cup$

I_{HC}. The parameters of each parabola are then obtained from,

$$a = \sum_{i \in I_{NC}} a_i + \sum_{i \in I_{LC}} C_i(\underline{p_i}) + \sum_{i \in I_{HC}} C_i(\overline{p_i})$$
$$- \sum_{i \in I_{NC}} \frac{b_i^2}{4c_i} + \left(\sum_{i \in I_{NC}} \frac{1}{4c_i} \right) \frac{\left(\sum_{i \in I_{NC}} \frac{b_i}{2c_i} - P_{con} \right)^2}{\left(\sum_{i \in I_{NC}} \frac{1}{2c_i} \right)^2} \tag{A.3}$$
$$b = \frac{2 \sum_{i \in I_{NC}} \left(\frac{b_i}{2c_i} - P_{con} \right) \left(\sum_{i \in I_{NC}} \frac{1}{4c_i} \right)}{\left(\sum_{i \in I_{NC}} \frac{1}{2c_i} \right)^2} \tag{A.4}$$
$$c = \frac{\sum_{i \in I_{NC}} \frac{1}{4c_i}}{\left(\sum_{i \in I_{NC}} \frac{1}{2c_i} \right)^2} \tag{A.5}$$

where,

$$P_{con} = \sum_{i \in I_{HC}} \overline{p_i} + \sum_{i \in I_{LC}} \underline{p_i} \tag{A.6}$$

By sorting the values of λ which mark the bifurcations in unit behavior, the sets I_{LC}, I_{NC} and I_{HC} are found. The equations (A.3), (A.4) and (A.5) can then be evaluated to find the cost function which bridges a section where these behavior sets remain constant.

Appendix B

Tables from Examples

Table B.1: Cost curve aggregation/DP schedule for problem in [27].

	Unit									1	1	1	1	1	1	1	1	1	1	2
Hour	1	2	3	4	5	6	7	8	9	0	1	1	3	4	5	6	7	8	9	0
1	1	1	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0
2	1	1	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0
3	1	1	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0
4	1	1	0	1	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0
5	1	1	0	1	0	0	0	0	0	0	1	1	0	1	0	0	0	0	0	0
6	1	1	1	1	0	0	0	0	0	0	1	1	0	1	0	0	0	0	0	0
7	1	1	1	1	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0
8	1	1	1	1	1	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0
9	1	1	1	1	1	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0
10	1	1	1	1	1	1	1	0	0	0	1	1	1	1	1	0	0	0	0	0
11	1	1	1	1	1	1	1	0	0	0	1	1	1	1	1	1	0	0	0	0
12	1	1	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	1	0	0
13	1	1	1	1	1	1	0	0	0	0	1	1	1	1	1	1	0	0	0	0
14	1	1	1	1	1	1	0	0	0	0	1	1	0	1	1	0	0	0	0	0
15	1	1	1	1	1	0	0	0	0	0	1	1	0	1	1	0	0	0	0	0
16	1	1	1	1	1	0	0	0	0	0	1	1	0	1	1	0	0	0	0	0
17	1	1	1	1	1	0	0	0	0	0	1	1	0	1	1	0	0	0	0	0
18	1	1	1	1	1	0	0	0	0	0	1	1	0	1	1	0	0	0	0	0
19	1	1	1	1	1	1	0	0	0	0	1	1	1	1	1	1	0	0	0	0
20	1	1	1	1	1	1	0	0	0	0	1	1	1	1	1	1	0	0	0	0
21	1	1	1	1	1	1	0	0	0	0	1	1	1	1	1	1	0	0	0	0
22	1	1	1	1	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0
23	1	1	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0
24	1	1	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0

Table B.2: Lagrangian Relaxation schedule reported in [27].

	Unit									1	1	1	1	1	1	1	1	1	1	2
Hour	1	2	3	4	5	6	7	8	9	0	1	1	3	4	5	6	7	8	9	0
1	1	1	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0
2	1	1	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0
3	1	1	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0
4	1	1	0	1	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0
5	1	1	0	1	0	0	0	0	0	0	1	1	0	1	0	0	0	0	0	0
6	1	1	1	1	0	0	0	0	0	0	1	1	0	1	0	0	0	0	0	0
7	1	1	1	1	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0
8	1	1	1	1	1	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0
9	1	1	1	1	1	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0
10	1	1	1	1	1	1	0	0	0	0	1	1	1	1	1	1	0	0	0	0
11	1	1	1	1	1	1	1	0	0	0	1	1	1	1	1	1	0	0	0	0
12	1	1	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	1	0	0
13	1	1	1	1	1	1	1	0	0	0	1	1	1	1	1	1	0	0	0	0
14	1	1	1	1	1	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0
15	1	1	0	1	1	0	0	0	0	0	1	1	0	1	1	0	0	0	0	0
16	1	1	0	1	1	0	0	0	0	0	1	1	0	1	1	0	0	0	0	0
17	1	1	0	1	1	0	0	0	0	0	1	1	0	1	1	0	0	0	0	0
18	1	1	0	1	1	0	0	0	0	0	1	1	0	1	1	0	0	0	0	0
19	1	1	0	1	1	0	0	0	0	0	1	1	0	1	1	0	0	0	0	0
20	1	1	1	1	1	1	0	0	1	0	1	1	1	1	1	0	0	0	1	0
21	1	1	1	1	1	1	0	0	0	0	1	1	1	1	1	0	0	0	0	0
22	1	1	1	1	0	1	0	0	0	0	1	1	1	0	0	0	0	0	0	0
23	1	1	1	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0
24	1	1	1	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0

Table B.3: Parameters to model $r(t)$.

a_1	a_2	σ_e	B	x_0	x_{-1}
1.7387	-0.8100	10	400	100	200

Table B.4: Marginal cost of power for example.

hour	1	2	3	4	5	6	7	8
λ	19.76	22.512	31.9445	28.8995	17.3635	30.0265	27.657	32.6515
hour	9	10	11	12	13	14	15	16
λ	34.5485	35.469	34.6675	34.4505	33.4845	35.224	39.3435	35.728
hour	17	18	19	20	21	22	23	24
λ	36.0465	34.7795	33.509	37.3275	40.313	33.579	32.249	21.2275

Table B.5: Imbalance penalty unit.

LL	UL	a	b	c
0	∞	2500	69.7250	0.0178

Table B.6: Stochastic schedule.

	Unit									1	1	1	1	1	1	1	1	1	1	2
Hour	1	2	3	4	5	6	7	8	9	0	1	1	3	4	5	6	7	8	9	0
1	1	1	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0
2	1	1	1	1	1	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0
3	1	1	1	1	1	1	0	0	0	0	1	1	1	1	1	1	0	0	0	0
4	1	1	1	1	1	1	0	0	0	0	1	1	1	1	1	1	0	0	0	0
5	1	1	1	1	1	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0
6	1	1	1	1	1	1	0	0	0	0	1	1	1	1	1	1	0	0	0	0
7	1	1	1	1	1	1	0	0	0	0	1	1	1	1	1	1	0	0	0	0
8	1	1	1	1	1	1	1	0	0	0	1	1	1	1	1	1	1	0	0	0
9	1	1	1	1	1	1	1	0	0	0	1	1	1	1	1	1	1	0	0	0
10	1	1	1	1	1	1	1	0	0	0	1	1	1	1	1	1	1	0	0	0
11	1	1	1	1	1	1	1	0	0	0	1	1	1	1	1	1	1	0	0	0
12	1	1	1	1	1	1	1	0	0	0	1	1	1	1	1	1	1	0	0	0
13	1	1	1	1	1	1	1	0	0	0	1	1	1	1	1	1	1	0	0	0
14	1	1	1	1	1	1	1	0	0	0	1	1	1	1	1	1	1	0	0	0
15	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	0	0
16	1	1	1	1	1	1	1	0	0	0	1	1	1	1	1	1	1	0	0	0
17	1	1	1	1	1	1	1	0	0	0	1	1	1	1	1	1	1	0	0	0
18	1	1	1	1	1	1	1	0	0	0	1	1	1	1	1	1	1	0	0	0
19	1	1	1	1	1	1	1	0	0	0	1	1	1	1	1	1	1	0	0	0
20	1	1	1	1	1	1	1	1	0	0	1	1	1	1	1	1	1	0	0	0
21	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	0	0
22	1	1	1	1	1	1	1	0	0	0	1	1	1	1	1	1	1	0	0	0
23	1	1	1	1	1	1	1	0	0	0	1	1	1	1	1	1	1	0	0	0
24	1	1	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0

Table B.7: Lagrangian/deterministic schedule.

	Unit									1	1	1	1	1	1	1	1	1	1	2
Hour	1	2	3	4	5	6	7	8	9	0	1	1	3	4	5	6	7	8	9	0
1	1	1	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0
2	1	1	1	1	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0
3	1	1	1	1	1	1	0	0	0	0	1	1	1	1	1	1	0	0	0	0
4	1	1	1	1	1	1	0	0	0	0	1	1	1	1	1	1	0	0	0	0
5	1	1	1	1	1	1	0	0	0	0	1	1	1	1	1	1	0	0	0	0
6	1	1	1	1	1	1	0	0	0	0	1	1	1	1	1	1	0	0	0	0
7	1	1	1	1	1	1	0	0	0	0	1	1	1	1	1	1	0	0	0	0
8	1	1	1	1	1	1	0	0	0	0	1	1	1	1	1	1	0	0	0	0
9	1	1	1	1	1	1	1	0	0	0	1	1	1	1	1	1	1	0	0	0
10	1	1	1	1	1	1	1	0	0	0	1	1	1	1	1	1	1	0	0	0
11	1	1	1	1	1	1	1	0	0	0	1	1	1	1	1	1	1	0	0	0
12	1	1	1	1	1	1	1	0	0	0	1	1	1	1	1	1	1	0	0	0
13	1	1	1	1	1	1	1	0	0	0	1	1	1	1	1	1	1	0	0	0
14	1	1	1	1	1	1	1	0	0	0	1	1	1	1	1	1	1	0	0	0
15	1	1	1	1	1	1	1	1	0	0	1	1	1	1	1	1	1	1	0	0
16	1	1	1	1	1	1	1	0	0	0	1	1	1	1	1	1	1	0	0	0
17	1	1	1	1	1	1	1	0	0	0	1	1	1	1	1	1	1	0	0	0
18	1	1	1	1	1	1	1	0	0	0	1	1	1	1	1	1	1	0	0	0
19	1	1	1	1	1	1	1	0	0	0	1	1	1	1	1	1	1	0	0	0
20	1	1	1	1	1	1	1	0	0	0	1	1	1	1	1	1	1	0	0	0
21	1	1	1	1	1	1	1	1	0	0	1	1	1	1	1	1	1	1	0	0
22	1	1	1	1	1	1	1	0	0	0	1	1	1	1	1	1	1	0	0	0
23	1	1	1	1	1	1	0	0	0	0	1	1	1	1	1	1	0	0	0	0
24	1	1	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0

Appendix C

DP Example Continued

By applying (4.29) for the first two decision stages, one obtains:

$$\begin{bmatrix} P_1 & P_2 \end{bmatrix} = \begin{bmatrix} 2 & 2 \\ 2 & 2 \\ 2 & 3 \end{bmatrix} \tag{C.1}$$

Applying (4.28) and (4.29) a third time yields

$$F_3^T = \min \left\{ \begin{bmatrix} -4 & -4 & -4 \\ -2 & -2 & -2 \\ -2 & -2 & -2 \end{bmatrix} + \begin{bmatrix} 1 & 2 & 4 \\ 2 & 1 & 2 \\ 4 & 2 & 1 \end{bmatrix} + \begin{bmatrix} -2 & -3 & -2 \\ -2 & -3 & -2 \\ -2 & -3 & -2 \end{bmatrix} \right\} \tag{C.2}$$

$$F_3^T = \min \left\{ \begin{bmatrix} -5 & -5 & -2 \\ -2 & -4 & -2 \\ 0 & -3 & -3 \end{bmatrix} \right\} \tag{C.3}$$

$$F_3 = \begin{bmatrix} -5 \\ -5 \\ -3 \end{bmatrix} \quad , \quad P_3 = \begin{bmatrix} 1 \\ 1 \\ 3 \end{bmatrix} \tag{C.4}$$

The equations for the fourth step, the last one for this problem, are as follows:

$$F_4^T = \min \left\{ \begin{bmatrix} -5 & -5 & -5 \\ -5 & -5 & -5 \\ -3 & -3 & -3 \end{bmatrix} + \begin{bmatrix} 1 & 2 & 4 \\ 2 & 1 & 2 \\ 4 & 2 & 1 \end{bmatrix} + \begin{bmatrix} -4 & -1 & -6 \\ -4 & -1 & -6 \\ -4 & -1 & -6 \end{bmatrix} \right\} \tag{C.5}$$

$$F_4^T = \min \left\{ \begin{bmatrix} -8 & -4 & -7 \\ -7 & -5 & -9 \\ -3 & -2 & -8 \end{bmatrix} \right\} \tag{C.6}$$

$$F_4 = \begin{bmatrix} -8 \\ -5 \\ -9 \end{bmatrix} \quad , \quad P_4 = \begin{bmatrix} 1 \\ 2 \\ 2 \end{bmatrix} \tag{C.7}$$

The results are regrouped into a 'F' and a 'P' matrix as follows:

$$\begin{bmatrix} F_1 & F_2 & F_3 & F_4 \end{bmatrix} = \begin{bmatrix} 0 & -4 & -5 & -8 \\ -2 & -2 & -5 & -5 \\ -2 & -2 & -3 & -9 \end{bmatrix} \tag{C.8}$$

$$\begin{bmatrix} F_1 & F_2 & F_3 & F_4 \end{bmatrix} = P = \begin{bmatrix} 2 & 2 & 1 & 1 \\ 2 & 2 & 1 & 2 \\ 2 & 3 & 3 & 2 \end{bmatrix} \tag{C.9}$$

Vita

Candidate's full name: Etienne Jacques Dupuis

University attended: University of New Brunswick, B.Sc.E., 2004