City University
Cass Business School
Specialist Masters Program 2003

Efficient Lattices for Market Calibrated Derivatives Valuation

Dennis Furey
under supervision of Nick Webber

Submitted in part fulfillment of the requirements for the degree of Master of Science in Mathematical Trading and Finance of City University and Cass Business School, September 2004

Abstract

A market calibrated generalization of the willow tree lattice model due to Curran for pricing derivative securities [Cur01] is further generalized to a new model, the *crossover lattice*, which scales more efficiently and accurately to large state spaces through a hierarchical method of decomposition. In this way, certain practical disadvantages of the standard willow tree model are addressed, specifically its dependence on the assumption of geometric Brownian motion, and its proneness to instability or infeasibility where large state spaces are concerned.

This technique is combined with a novel method of moment matching discretization of continuous probability density functions to a freely specified number of state price points. In contrast with many existing methods, the proposed construction thereby provides for convenient and flexible decoupled selection of arbitrary non-constant time intervals and state space sizes, and does not depend on any restrictive assumptions about the underlying marginal distributions.

Moreover, the model can be generalized to multifactor valuation problems, where the payoff of a derivative is contingent on more than one risky asset or stochastic market variable. In contrast with higher dimensional versions of traditional lattice models, state explosion is not inevitable for this model, because state space sizes are chosen independently. In this context, it may provide a practical alternative for derivatives valuation applications that are otherwise intractable.

Contents

1 Introduction **6**

1.1 Background on derivatives pricing 6

1.2 Aims and scope 7

1.2.1 Variably branched implied trees 8

1.2.2 Moment matching discretization 8

1.2.3 Efficient multifactor valuations 8

1.3 Related work 9

1.3.1 Historical interest 9

1.3.2 Implied trees 10

1.3.3 Willow trees 12

1.3.4 Multifactor models 13

1.3.5 Non-recombining lattices 14

2 A novel class of lattices **16**

2.1 Inputs for an implied willow tree model 16

2.2 Exercising the implied willow tree model 18

2.2.1 Verification of probability and martingale properties . . . 20

2.2.2 Problems with the implied willow tree 20
2.3 Scaling up the implied willow tree 25
2.3.1 Intuitive overview 25
2.3.2 Explicit construction 26
2.3.3 Subproblems to be solved 29
2.4 Informal complexity analysis 31
2.5 Decomposition strategies and branching patterns 33
2.5.1 Warming up with a willow tree 33
2.5.2 A simple decomposition – the cut 35
2.5.3 A fairer decomposition - the deal 40
2.5.4 Cutting a deal for nested decomposition 43
2.5.5 A composable decomposition – the skim 43
2.6 The crossover lattice decomposition strategy 49
2.6.1 A recursive algorithm 49
2.6.2 Exercising the crossover lattice 51
2.6.3 Assessing the efficacy of decomposition 51
2.7 Summary 57

3 Verification of Convergence **58**

3.1 Discrete approximation of continuous distributions 58
3.1.1 Overview of discretization 59
3.1.2 A motivating example 60
3.1.3 The discretization method 62

3.1.4 The calibration space 63
3.2 Marginal distributions under Black-Scholes assumptions 65
3.2.1 Black-Scholes prices 67
3.2.2 Crossover lattice prices 67
3.3 Results 68
3.3.1 Varying the number of time steps 68
3.3.2 Varying the number of states 68
3.3.3 General observations and conclusions 68

4 Multifactor Models **73**
4.1 Inputs for a multifactor crossover lattice 73
4.2 Simple cases 74
4.3 Decomposable cases 77
4.3.1 Selection of subsets 77
4.3.2 Solving for weights 79
4.4 Summary 80

5 Concluding Remarks **81**
5.1 Decomposition strategies 81
5.2 Distribution calibration parameters 82
5.3 Convergence properties 83

A Supplementary data **84**

Chapter 1

Introduction

The valuation of contingent claims is one of the central themes of mathematical finance, encompassing a diverse literature of analytical and numerical methods inspired by the endless variety and sophistication of derivative instruments in the market. However, trustworthy valuation methods are more than a matter of academic interest, being vital to all market participants trading derivatives, whether for hedging, speculation, arbitrage, or risk management, in order to make informed decisions consistent with their views and objectives.

In this chapter, some introductory background material on the nature of the problems addressed and the aims and scope of this work is set out. This discussion is concluded with a review of the relevant literature.

1.1 Background on derivatives pricing

A derivative or contingent claim in the most general sense is an agreement stipulating an exchange of value between two parties at some future date in an amount determined by specified market conditions or events. The problem of derivatives valuation ultimately reduces to the simple question of deciding the fair value of such a contract at its inception, based on uncertain knowledge of the future course of the market.

This problem is indeed solvable, with one solution being the current market price of a so called replicating portfolio. This contains a collection of traded instru-

ments of a nature and quantity chosen to ensure that their combined net value will equal the payoff of the derivatives contract regardless of how the market moves. The fair price of the contract is therefore no more or less than that of the replicating portfolio.

Although pricing by replication is an ideal solution, it is not always achievable because the necessary portfolio instruments may not exist, so alternative pricing methods are needed. Those that are commonly used can be broadly classified among Monte Carlo simulation, partial differential equations, and lattice models.

- Monte Carlo simulation involves equating the value of a derivative contract with its average payoff over a representative sample of randomly generated evolutions of the market.
- Partial differential equations are based on an attempt to find a closed form solution for a price directly in terms of such continuous parameters as diffusion and drift that are assumed to capture the essential features of the market.
- Lattice models are a discrete analogue to partial differential equations aimed at approximate numerical solutions rather than closed forms.

Lattice models can be further subdivided between finite difference methods and trees, but finite difference methods are not discussed further. Strictly speaking, it would be more appropriate in most cases to refer to them as rooted graphs than as trees, but the indiscriminate usage of trees is entrenched in the financial literature, along with the oxymoronic "recombining tree".

By way of some further background, time in a lattice is divided into discrete steps, and at each step there is a discrete set of states consisting of prices or other underlying market variables on which a derivative contract's payoff may depend. Branches connect states in each time step to selected states in the subsequent step, with the intuition being that the market chooses a path through them as time proceeds. A price for the derivative is then inferred by averaging the payoff over all possible paths.

1.2 Aims and scope

While each of these approaches is known to have its strengths and weaknesses, this dissertation focuses exclusively on lattice models. The main ideas and con-

tributions are summarized below.

1.2.1 Variably branched implied trees

The brief description of lattices in Section 1.1 necessarily neglects some important issues about whether a given lattice faithfully represents the conditions of the actual market. Dubious simplifying assumptions made in the name of academic rigor or elegance are not unknown in the finance literature, while other schools of thought favor the so called market calibrated or implied tree pricing methodologies. The present work staunchly adheres to the latter orientation, and aims to develop a lattice based pricing method that is independent of any assumptions about the underlying statistical properties of the market.

Whereas much if not all of the existing work to date on implied trees relies on a fixed branching pattern, the present work relaxes this requirement. The resulting models are found to be parsimonious, scalable, and versatile.

1.2.2 Moment matching discretization

Because no assumptions are made about the underlying marginal distributions, they need to be specified as inputs to the model. If the marginals are known only in continuous form, then a discrete approximation for them must be found. An original technique that allows a discrete approximation to be selected and optimizes the mean and variance to match those of the continuous distribution is developed in Chapter 3.

1.2.3 Efficient multifactor valuations

An important benefit of unconstrained branching patterns is that it becomes trivial to specify the granularity of the space and time discretizations independently of each other. This feature has a major implication for multifactor derivatives valuations, where the payoff of a derivative depends on more than one underlying risky assets or stochastic market variable. Conventional lattices are constrained to increase exponentially in size with the number of time steps in multifactor versions, which severely curtails their ability to be used on problems of any significant size. The one proposed in this work does not have this drawback.

1.3 Related work

The subject of derivatives valuation using lattices has been investigated extensively in the financial literature. The main areas that are of some relevance to the present work are the implied tree models and the multifactor or higher dimensional lattices, which are discussed below. A particularly relevant contribution in a class by itself is the willow tree model, whose discussion follows.

1.3.1 Historical interest

A simple numerical valuation using recombining lattices for single factor options in a Black-Scholes economy was pioneered by Cox, Ross, and Rubinstein [CRR79], with later enhancements due to [JR82], [Tia93], and [LR96] among others. This work is of interest because its influence persists in more recent contributions which retain many of the same ideas.

For example, generalizations remaining within this framework are considered in [AW01], which advocates a heptanomial version of the Jarrow and Rudd model for optimum performance, and in [Sch97], where higher order branching patterns and underlying distributions other than log-normal are accommodated. In both of these single factor models, the recombining property is preserved.

In its most succinct formulation, the standard binomial tree estimates a price of

$$p = e^{-rT} \sum_{i=0}^{n} \binom{n}{i} p^i (1-p)^{n-i} f\left(S u^i d^{n-i}\right) \tag{1.1}$$

for a European option expiring at time T with payoff function f on an asset with spot price S, where r is the discount rate per unit time, and n is the number of time steps. From an assumed volatility σ follow the remaining parameters,

$$u = e^{\sigma\sqrt{\Delta t}} \tag{1.2}$$

$$d = e^{-\sigma\sqrt{\Delta t}} \tag{1.3}$$

$$p = \frac{e^{r\Delta t} - d}{u - d} \tag{1.4}$$

where $\Delta t = T/n$.

Given that the asset price in state i at time t is $Su^i d^{t-i}$, Equation 1.1 may be recognizable as the expected value of the payoff at expiration $f\left(Su^i d^{n-i}\right)$ with respect to a probability mass function, multiplied by a discount factor e^{-rT}. The probability mass function maps a terminal state number i to the probability of reaching it by traversing the lattice, and takes this form because at each step through the lattice a choice is possible between stepping up with probability p or down with probability $(1-p)$.

On this basis, Equation 1.4 follows directly from the martingale condition that the price of the asset in any state should coincide with its discounted expectation.

$$\begin{aligned} S &= e^{-r\Delta t}(pSu + (1-p)Sd) \\ e^{r\Delta t} &= pu + (1-p)d \\ pu + d - pd &= e^{r\Delta t} \\ p(u-d) &= e^{r\Delta t} - d \\ p &= \frac{e^{r\Delta t} - d}{u-d} \end{aligned}$$

As Equations 1.2 and 1.3 imply, $ud = 1$ by design, so a downward movement in price exactly cancels an upward movement, and any two neighboring state prices in a time step will have only three successors between them, being said thereby to recombine. This condition allows the number of states to grow only linearly with time rather than exponentially.

In summary, the binomial lattice model sets the example for pricing a derivative as a discounted expected payoff in a discrete idealization of time and space where martingale conditions hold sway. Everything about the binomial lattice is carefully designed to strike a delicate balance. The martingale condition, the linear growth rate of state spaces, the discount rates, the state transition probabilities, and the asset volatility are all simultaneously and inextricably secured.

1.3.2 Implied trees

The binomial lattice and related methods are based on the assumption that the underlying asset price follows a path described by geometric Brownian motion

$$dS = \mu S dt + \sigma S dz \tag{1.5}$$

where μ and σ are constant, but it is more realistic to relax this assumption slightly and allow a process of the form

$$dS = \mu(t)Sdt + \sigma(S,t)Sdz \tag{1.6}$$

One approach to doing so leads to the so called implied binomial trees of Derman and Kani [DK94], or their generalization to implied trinomial trees described in [DKC96]. Unlike an ordinary binomial tree, an implied tree has state transition probabilities that non-constant and prices that are not in constant ratios. These parameters are determined in such a way as to ensure that the tree will correctly price an arbitrary given set of call and put options.

To build the next level of an implied binomial tree after level n (where n starts from 0), there are $n+2$ asset prices $S_{n+1,i}$ to be determined, and $n+1$ transition probabilities $p_{n,i}$ from state (n,i) to state $(n+1,i+1)$. These $2n+3$ unknown quantities are solved from the known quantities of $n+1$ forward prices $F_{n,i} = S_{n,i}e^{r\Delta t}$ and $n+1$ prices for options expiring at T_{n+1}. The remaining degree of freedom is used for a centering condition similar to the standard binomial tree of Cox, Ross, and Rubinstein [CRR79].

The solution determined by these conditions for prices above the central node is

$$S_{n+1,i+1} = \frac{S_{n+1,i}(e^{r\Delta t}c(S_{n,i},T_{n+1}) - \Sigma) - \lambda_{n,i}S_{n,i}(F_{n,i} - S_{n+1,i})}{(e^{r\Delta t}c(S_{n,i},T_{n+1}) - \Sigma) - \lambda_{n,i}(F_{n,i} - S_{n+1,i})} \tag{1.7}$$

where $c(S_{ni},T_{n+1})$ is the price of a European call with strike $S_{n,i}$ and T_{n+1} as its time to expiration, $\lambda_{n,i}$ is the solution of

$$e^{r\Delta t}\lambda_{n+1,i} = \begin{cases} p_{n,n}\lambda_{n,n} & \text{if} \quad i = n+1 \\ p_{n,i-1}\lambda_{n,i-1} + (1-p_{n,i})\lambda_{n,i} & \text{if} \quad 1 \le i \le n+1 \\ (1-p_{n0})\lambda_{n,0} & \text{if} \quad i = 0 \end{cases} \tag{1.8}$$

or the so called Arrow-Debreu price, and $\Sigma = \sum_{j=i+1}^{n} \lambda_{nj}(S_{ni} - F_{nj})$. For prices below the central node, the solution is

$$S_{n+1,i} = \frac{S_{n+1,i+1}(e^{r\Delta t}p(S_{n,i},T_{n+1}) - \Sigma) + \lambda_{n,i}S_{n,i}(F_{n,i} - S_{n+1,i+1})}{(e^{r\Delta t}p(S_{n,i},T_{n+1}) - \Sigma) + \lambda_{n,i}(F_{n,i} - S_{n+1,i+1})} \tag{1.9}$$

where $\Sigma = \sum_{j=0}^{i-1} \lambda_{n,j}(S_{n,j} - F_{n,j})$ and $p(S_{n,i},T_{n+1})$ is the price of a European put with strike $S_{n,i}$ and time to maturity T_{n+1}. The implied transition probabilities at any state are

$$p_{n,i} = \frac{F_{n,i} - S_{n+1,i}}{S_{n+1,i+1} - S_{n+1,i}} \tag{1.10}$$

A point noted by many authors and easily verified by any attempt to implement these implied trees is that they are prone to anomalous results due to negative probabilities.[1] Nor is this issue resolved by the trinomial version.

This outcome is to be expected inasmuch as the derivation of the model never insists mathematically on probabilities being between 0 and 1. A proper solution taking this requirement into account makes it an optimization problem rather than a straightforward system of linear equations.

The implied tree may be seen as an attempt to escape the restrictive assumptions of the standard binomial tree, but in so doing it disturbs the balance mentioned at the end of Section 1.3.1.

1.3.3 Willow trees

One attempt to construct a lattice model where negative probabilities are specifically excluded is the so called Willow tree due to Curran [Cur01]. He derives transition probabilities p_{ij} by solving a linear programming problem. The problem is to minimize

$$\sum_{i=1}^{n}\sum_{j=1}^{n} p_{ij}\left|\sqrt{1+\alpha}z_i - z_j\right|^3 \tag{1.11}$$

subject to the constraints

$$\sum_{i=1}^{n} p_{ij} = 1 \tag{1.12}$$

$$\sum_{i=1}^{n} q_i p_{ij} = q_j \tag{1.13}$$

$$\sqrt{1+\alpha}\sum_{j=1}^{n} p_{ij} z_j = z_i \tag{1.14}$$

$$\sum_{j=1}^{n} p_{ij}\left(z_j\sqrt{1+\alpha}\right)^2 - z_i^2 = \alpha \tag{1.15}$$

$$p_{ij} \geq 0 \tag{1.16}$$

[1] One can only sympathize with students required to get reasonable results from it for an assignment.

Where q_i is the unconditional probability of state i assuming geometric Brownian motion, z_i is the corresponding normal variate, and α is described as a scaling parameter.

These equations are constructed to match the moments of the underlying distributions assuming lognormality, although the cost criterion in 1.11 is more of an arbitrary choice. Extensibility of the willow tree to other underlying process models incorporating a volatility smile or skew is claimed but no formal specification is given.

The use of linear programming to solve for transition probabilities in the willow tree model is a powerful and innovative technique. It is potentially even more useful for implied tree models where negative probabilities are a problem than those based on idealized assumptions for which simpler solutions are known, as the willow tree is. However, such approaches appear to constitute somewhat of a gap in the literature.

1.3.4 Multifactor models

The above methods are restricted to economies that can be characterized by a single source of risk, for example the price movement of a single underlying equity, and are known as single factor models. These constrast with so called multifactor models, where the payoff of a derivative depends on more than one risky asset or stochastic market variable.

One of the early references on multifactor lattice models is [Rub94], which describes a direct generalization of the standard binomial tree to cope with two assets. A related alternative is a multifactor generalization of the trinomial tree described in [CL95].

For the binomial version, in each state, asset 1 can move up by a ratio u or down by d. If asset 1 moves up, asset 2 can change by ratios of either A or B, and if it moves down, asset 2 change by ratios C or D. If the assets are lognormal with correlation ρ, then the ratios for asset 1 are

$$u = \exp(\mu_1 \Delta t + \sigma_1 \sqrt{\Delta t}) \tag{1.17}$$

$$d = \exp(\mu_1 \Delta t - \sigma_1 \sqrt{\Delta t}) \tag{1.18}$$

where the drifts are given by

$$\mu_1 = b_1 - \sigma_1^2/2 \tag{1.19}$$
$$\mu_2 = b_2 - \sigma_1^2/2 \tag{1.20}$$

for costs of carry b and volatilities σ, and the ratios for asset 2 are

$$A = \exp\left(\mu_2 \Delta t + \sigma_2 \sqrt{\Delta t}\left(\rho + \sqrt{1-\rho^2}\right)\right) \tag{1.21}$$
$$B = \exp\left(\mu_2 \Delta t + \sigma_2 \sqrt{\Delta t}\left(\rho - \sqrt{1-\rho^2}\right)\right) \tag{1.22}$$
$$C = \exp\left(\mu_2 \Delta t - \sigma_2 \sqrt{\Delta t}\left(\rho - \sqrt{1-\rho^2}\right)\right) \tag{1.23}$$
$$D = \exp\left(\mu_2 \Delta t - \sigma_2 \sqrt{\Delta t}\left(\rho + \sqrt{1-\rho^2}\right)\right) \tag{1.24}$$

Each state therefore has four successors rather than two. Although the two factor binomial tree grows faster with time than the ordinary one, it can at least be made to recombine if the underlying processes conform to the assumption of lognormality. In some cases they do not, making this approach inapplicable.

1.3.5 Non-recombining lattices

In a non-recombining lattice, the number of states grows exponentially with time because states may have disjoint sets of successors. This type of lattice may occur in certain multifactor derivatives valuation applications, particularly in fixed income.

Several approaches have been tried for coping with the state explosion in non-recombining lattices. These can be classified roughly into pruning the lattice, interpolation, and using a more carefully considered choice representative underlying asset prices. Of these three approaches, only the second actually avoids exponential complexity, and would therefore seem to be the most promising, but it is shown in this dissertation that improvements on the other two may be possible.

This more recent work attains a maturity and mathematical sophistication that resists simple summaries in a few lines. The reader is referred to the original references for the details.

Pruning the lattice

Pruning the lattice consists essentially of choosing to neglect low probability states at each time step and refraining from enumerating their descendents. This technique can be compared to a sophisticated form of curtailed range methods [AWDN04], which are good practice even with recombining trees. An example based primarily on this approach is in [TL01], wherein only the states coinciding with modes (that is, local maxima) of the underlying probability distribution are selected for further elaboration into the next time step. A proof of convergence is given.

Using interpolation

The interpolation approach is taken by [SW03]. Essentially the positions of the nodes in the lattice in a multi-factor model are adjusted to allow for recombination, and drift errors incurred in the course of this step are stored, and the algorithm for valuing derivative is modified accordingly to compensate for the drift errors.

Munging the branching patterns

The remaining classification is exemplified by [Jäc00] and [MW02], which advocate the choice of a regular simplex in the factor space and alternating orientations with time steps, in order to achieve closer to optimal coverage of the underlying probability distribution. The method in [Jäc00] is amenable to a concurrent implementation.

Chapter 2

A novel class of lattices

Every branch in me that beareth not fruit he taketh away . . .

— John 15:2

This chapter begins with a straightforward generalization of Curran's willow trees [Cur01] to arbitrary marginal distributions and time varying discount rates and volatilities, allowing them to serve in a comparable capacity to the so called implied trees of Derman, Kani, and Chriss [DK94, DKC96]. Subsequently a further enhancement is introduced, namely the *crossover lattice*, to provide an attractive alternative in applications where state explosion might otherwise be prohibitive.

2.1 Inputs for an implied willow tree model

Let the price of an asset S at time k in state i be denoted by S_i^k, where i takes a discrete value from 0 to N_k and k varies discretely from 0 to $\mathcal{T}$, with 0 corresponding to the present time. Denote the unconditional risk neutral probability of state i occurring at time k by q_i^k.

Values of k, S_i^k and q_i^k may be calibrated to market data, adjusted for a proprietary market view, or tailored to particular features of the instrument to be valued (for example barriers and exercise dates). However, consistency with these definitions

requires that probabilities sum to unity

$$\forall k \leq \mathcal{T},\ \sum_{i=0}^{N_k} q_i^k = 1 \tag{2.1}$$

and marginal variances are monotonically increasing

$$\forall k < \mathcal{T},\ V\left[S^k\right] < V\left[S^{k+1}\right] \tag{2.2}$$

where the variances are based on the given risk neutral distributions.

$$V\left[S^k\right] = \sum_{i=0}^{N_k} q_i^k \left(S_i^k\right)^2 - \left(\sum_{i=0}^{N_k} q_i^k S_i^k\right)^2 \tag{2.3}$$

Valuation of a derivative security based on this model requires the construction of a lattice that associates a state transition probability with each possible pair of consecutive states. Let p_{ij}^k denote the probability of changing from state i at time k to state j at time $k+1$, given that state i prevails at time k. Consistency with the unconditional probabilities q_i^k implies these constraints.

$$\forall k < \mathcal{T},\ \forall i \leq N_k,\ \sum_{j=0}^{N_{k+1}} p_{ij}^k = 1 \tag{2.4}$$

$$\forall k < \mathcal{T},\ \forall j \leq N_{k+1},\ \sum_{i=0}^{N_k} p_{ij}^k q_i^k = q_j^{k+1} \tag{2.5}$$

The martingale property requires the discounted conditional expectation of the asset price in any state to be equal to the price in that state. The discount factor d_k at time k may be inferred from the fact that these distributions are risk neutral and therefore will have their means drifting at the risk free rate.

$$d_k = \frac{E\left[S^k\right]}{E\left[S^{k+1}\right]} = \frac{\sum_{i=0}^{N_k} q_i^k S_i^k}{\sum_{j=0}^{N_{k+1}} q_j^{k+1} S_j^{k+1}} \tag{2.6}$$

The martingale condition imposed on the transition probabilities is as follows.

$$\forall k < \mathcal{T},\ \forall i \leq N_k,\ d_k \sum_{j=0}^{N_{k+1}} p_{ij}^k S_j^{k+1} = S_i^k \tag{2.7}$$

Equations 2.4, 2.5, and 2.7 do not determine unique transition probabilities for lattices with more than three states per time step, so some consideration of further conditions is needed.

Curran proposes a fourth constraint to match the conditional variances in the lattice with the underlying geometric Brownian motion, and chooses the transition probabilities subject to a cost criterion favoring small price movements insofar as any choice remains. The cost he proposes for a transition probability increases as the cube of the difference between the original and discounted terminal price. This choice is justified on the basis that any lower order cost would be redundant in combination with the variance matching condition.

In the more general case presently under consideration, variance matching is arguably ill-posed and certainly not straightforward, but a comparable effect is achieved by using a cost function that weights each transition probability proportionately with the discounted return.

$$C = \sum_{k=0}^{\mathcal{T}-1} \sum_{i=0}^{N_k} \sum_{j=0}^{N_{k+1}} p_{ij}^k \left| \ln \frac{d_k S_j^{k+1}}{S_i^k} \right|^\alpha \tag{2.8}$$

In this equation, the exponent α is a tunable parameter which has been fixed at 2. Higher values of α impose a greater cost on non-zero probabilities for extreme returns, and express a belief in lower conditional volatility of the underlying asset. Anecdotal evidence suggests that lower values of α than 2 are conducive to non-zero probabilities for occasional long jumps. These can only be regarded as spurious in the context of a pure diffusion process, but may have applications to other underlying process models. Higher values of α than 2 do not noticeably alter the basic structure of the lattice.

2.2 Exercising the implied willow tree model

The transition probabilities are obtained by minimizing the cost given by Equation 2.8, subject to the constraints given by Equations 2.4, 2.5, and 2.7. This task can be expressed as a linear programming problem, which can be solved by standard software packages such as `lp_solve`, `PCx`, and the GNU Linear Programming kit, `glpk`.

Table 2.1 shows a data set consisting of state prices and their probabilities for five

	time $k=1$		time $k=2$		time $k=3$		time $k=4$		time $k=5$	
i	S_i^1	q_i^1	S_i^2	q_i^2	S_i^3	q_i^3	S_i^4	q_i^4	S_i^5	q_i^5
0	91.17	0.000002	88.45	0.000004	85.74	0.000005	72.91	0.000004	65.15	0.000003
1	93.03	0.000240	92.88	0.000165	91.55	0.000181	81.41	0.000202	75.36	0.000275
2	94.92	0.004450	97.54	0.005129	97.77	0.004596	90.89	0.007147	87.16	0.007082
3	96.85	0.053562	102.43	0.044962	104.40	0.045031	101.49	0.049553	100.82	0.073159
4	98.82	0.208810	107.57	0.182657	111.49	0.335178	113.31	0.133308	116.62	0.170231
5	100.83	0.392544	112.96	0.473904	119.05	0.344819	126.52	0.403494	134.89	0.414832
6	102.88	0.277615	118.62	0.243944	127.13	0.180858	141.26	0.352745	156.03	0.291693
7	104.97	0.054955	124.57	0.044528	135.75	0.079913	157.73	0.047179	180.47	0.037244
8	107.11	0.007659	130.81	0.004465	144.97	0.009103	176.11	0.006223	208.75	0.005220
9	109.29	0.000159	137.37	0.000238	154.80	0.000312	196.63	0.000143	241.46	0.000258
10	111.51	0.000003	144.25	0.000003	165.31	0.000004	219.55	0.000004	279.29	0.000003

Table 2.1: asset prices and probabilities expressing a particular market view

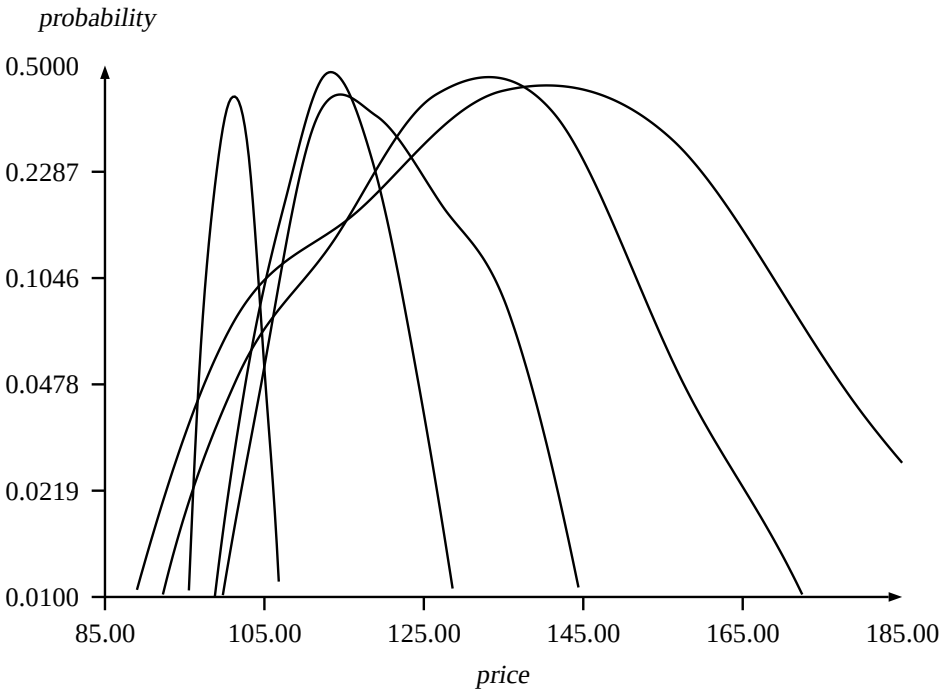

Figure 2.1: graphical representation of the marginal distributions interpolated from Table 2.1, illustrating irregular drift and increasing diffusion with time

consecutive time steps, and Figure 2.1 provides a graphical representation. The data are meant to be representative of a typical set of asset prices with risk neutral probabilities derived from econometric analysis of market data, and possibly adjusted to suit a particular market view. For illustrative purposes, these data were simulated from a random series of roughly bell shaped distributions. The data are consistent with the conditions of monotonically increasing variance and probabilities summing to unity. Although eleven states were chosen for each time step, varying numbers could have been used within the same model.

The state transition probabilities were obtained using `glpk` and are shown in Table 2.2. It is notable that a parsimonious model is automatically determined by the cost criterion, with the majority of possible state transitions assigned a zero probability, and not necessarily a fixed branching pattern. This example is typical insofar as it leads to a nearly binomial recombining tree, with minor corrections by additional branches originating from the most probable one or two states at each time step.

2.2.1 Verification of probability and martingale properties

As a spot check on the accuracy of these solutions, some numbers from the time $k = 3$ solution are displayed with higher precision in Tables 2.3 and 2.4. It can be seen that the probability constraints and martingale condition are accurate up to ten digits or more, with discrepancies due only to limited machine precision.

2.2.2 Problems with the implied willow tree

Although it has been established in principle that implied willow trees can be extended to other contexts than geometric Brownian motion, some issues remain to be addressed.

- The model can give rise to an infeasible linear programming problem if the data are insufficient to determine a solution.
- The number of variables in the linear programming problem increases as the product of the number of states in consecutive time steps, making it impractical for large state spaces.

time $k=1$			time $k=2$			time $k=3$			time $k=4$		
states			states			states			states		
i	j	p_{ij}^k	i	j	p_{ij}^k	i	j	p_{ij}^k	i	j	p_{ij}^k
0	2	0.016132	0	1	0.822235	0	2	0.706576	0	1	0.877220
0	3	0.983868	0	2	0.177765	0	3	0.293424	0	2	0.122780
1	3	0.609943	1	1	0.074579	1	2	0.104417	1	1	0.119112
1	4	0.390057	1	2	0.925421	1	3	0.895583	1	2	0.880888
2	3	0.196280	2	2	0.334595	2	3	0.517615	2	2	0.371179
2	4	0.803720	2	3	0.665405	2	4	0.482385	2	3	0.628821
3	4	0.784981	3	3	0.590295	3	4	0.912797	3	3	0.614499
3	5	0.215019	3	4	0.409705	3	5	0.087203	3	4	0.385501
4	4	0.374877	4	4	0.841750	4	0	0.000013	4	4	0.849374
4	5	0.625123	4	5	0.158250	4	1	0.000602	4	5	0.150626
						4	2	0.021254			
						4	3	0.140255			
						4	4	0.268475			
						4	5	0.374113			
						4	6	0.173367			
						4	7	0.005331			
						4	8	0.016155			
						4	9	0.000423			
						4	10	0.000011			
5	0	0.000011	5	0	0.000011	5	5	0.728228	5	0	0.000008
5	1	0.000422	5	1	0.000347	5	6	0.271772	5	1	0.000612
5	2	0.013067	5	2	0.005752				5	2	0.010536
5	3	0.111937	5	3	0.031814				5	3	0.094710
5	4	0.149445	5	4	0.343963				5	4	0.093930
5	5	0.435861	5	5	0.479963				5	5	0.680077
5	6	0.169439	5	7	0.120029				5	6	0.078488
5	7	0.107833	5	8	0.017457				5	7	0.031153
5	8	0.011374	5	9	0.000654				5	8	0.009846
5	9	0.000605	5	10	0.000009				5	9	0.000632
5	10	0.000007							5	10	0.000008
6	5	0.551938	6	5	0.362610	6	5	0.127538	6	5	0.341167
6	6	0.448062	6	6	0.630026	6	6	0.872462	6	6	0.658833
			6	7	0.007363						
7	5	0.137094	7	6	0.610113	7	6	0.539728	7	6	0.585529
7	6	0.862906	7	7	0.389887	7	7	0.460272	7	7	0.414471
8	6	0.727475	8	7	0.861239	8	7	0.933946	8	7	0.821409
8	7	0.272525	8	8	0.138761	8	8	0.066054	8	8	0.178591
9	6	0.316201	9	7	0.115537	9	7	0.347215	9	7	0.056797
9	7	0.683799	9	8	0.884463	9	8	0.652785	9	8	0.943203
10	7	0.901501	10	8	0.374874	10	8	0.749827	10	8	0.311027
10	8	0.098499	10	9	0.625126	10	9	0.250173	10	9	0.688973

Table 2.2: transition probabilities computed from Table 2.1, with zeros not shown

i	$\sum_{j=0}^{N_{k+1}} p_{ij}^k$	S_i^k	$d_k \sum_{j=0}^{N_{k+1}} p_{ij}^k S_j^{k+1}$
0	0.99999999998683164	85.7366852696775794	85.7366852658526284
1	1.00000000000000022	91.5544007900033989	91.5544007900034273
2	1.00000000000000089	97.7668809757577719	97.7668809757578856
3	1.00000000000000022	104.4009128261548511	104.4009128261548938
4	0.99999999999999978	111.4851009887185853	111.4851009887185711
5	1.00000000000000000	119.0499910969269450	119.0499910969269877
6	1.00000000000000000	127.1282014770078632	127.1282014770078916
7	1.00000000000000022	135.7545637917808108	135.7545637917808676
8	0.99999999999999967	144.9662732279726640	144.9662732279726640
9	0.99999999999994271	154.8030488745866080	154.8030488745779394
10	0.99999999999891409	165.3073049838433803	165.3073049836589803

Table 2.3: verification of Equations 2.4 and 2.7 with respect to the time $k = 3$ solution from Tables 2.1 and 2.2, showing that the outgoing probabilities sum nearly to 1 and the discounted expected payoffs on the asset are nearly its current price

j	q_j^{k+1}	$\sum_{i=0}^{N_k} p_{ij}^k q_i$
0	$4.4237809946452631 \times 10^{-6}$	$4.4237809946484310 \times 10^{-6}$
1	$2.0169951651707697 \times 10^{-4}$	$2.0169951651719537 \times 10^{-4}$
2	$7.1465979111240792 \times 10^{-3}$	$7.1465979111240176 \times 10^{-3}$
3	$4.9552675299094996 \times 10^{-2}$	$4.9552675299094906 \times 10^{-2}$
4	$1.3330807417635945 \times 10^{-1}$	$1.3330807417635943 \times 10^{-1}$
5	$4.0349400580737660 \times 10^{-1}$	$4.0349400580737671 \times 10^{-1}$
6	$3.5274452980974474 \times 10^{-1}$	$3.5274452980974486 \times 10^{-1}$
7	$4.7178577180138140 \times 10^{-2}$	$4.7178577180138070 \times 10^{-2}$
8	$6.2226637166722361 \times 10^{-3}$	$6.2226637166722413 \times 10^{-3}$
9	$1.4296803247445655 \times 10^{-4}$	$1.4296803247445655 \times 10^{-4}$
10	$3.7847695033692714 \times 10^{-6}$	$3.7847695033557832 \times 10^{-6}$

Table 2.4: verification of Equation 2.5 with respect to the time $k = 3$ solution from Tables 2.1 and 2.2, showing that the total probability of reaching each state is nearly the sum of the probabilities of all mutually exclusive ways of reaching it

	time $k=1$		time $k=2$		time $k=3$		time $k=4$		time $k=5$	
i	S_i^1	q_i^1	S_i^2	q_i^2	S_i^3	q_i^3	S_i^4	q_i^4	S_i^5	q_i^5
0	68.93	0.054318	69.98	0.027770	67.78	0.033288	61.60	0.023521	57.68	0.030632
1	81.96	0.054103	83.49	0.072917	82.76	0.057220	78.48	0.066738	75.19	0.092462
2	97.45	0.202844	99.62	0.169463	101.05	0.078315	100.00	0.112241	98.02	0.161836
3	115.87	0.164778	118.86	0.219134	123.39	0.213272	127.42	0.255686	127.77	0.104864
4	137.76	0.155419	141.82	0.173575	150.67	0.204640	162.35	0.147635	166.57	0.157682
5	163.80	0.135806	169.21	0.119319	183.98	0.201772	206.85	0.107860	217.14	0.209700
6	194.75	0.101718	201.89	0.091814	224.65	0.088240	263.56	0.183201	283.06	0.122458
7	231.55	0.088998	240.89	0.089530	274.30	0.084972	335.81	0.049349	369.00	0.096204
8	275.31	0.042015	287.42	0.036478	334.94	0.038280	427.87	0.053768	481.03	0.024162

Table 2.5: ordinary looking asset prices and probabilities with infeasible willow trees

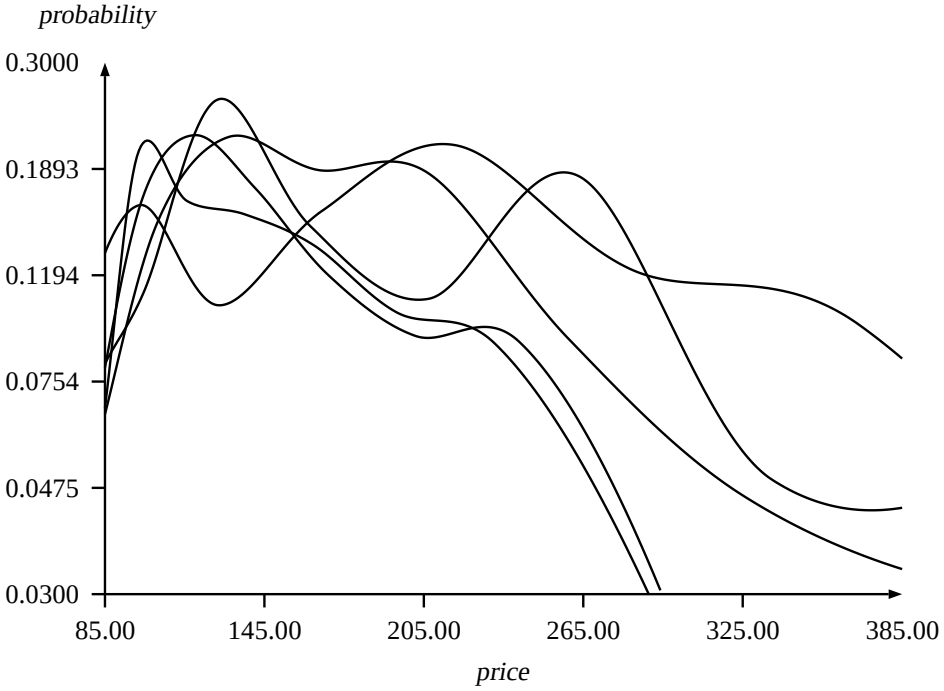

Figure 2.2: graphical representation of the marginal distributions interpolated from Table 2.5, illustrating distributions of an unusual shape

- For a competent linear programming solver using sparse matrix techniques, the time complexity becomes a more severe limitation than space.

An example of a data set whose implied willow tree transitions have no feasible solution despite monotonically increasing variance is shown in table 2.5, with some plots shown in Figure 2.2. Similar examples are not difficult to construct, and they become more the rule than the exception as state space sizes increase.

A rigorous quantitative study of infeasibility in linear programming problems and the conditions necessary to avoid it is beyond the scope of this dissertation, but to understand it on an intuitive level, one may consider that the limiting constraint in any system of transition probabilities is the martingale condition given by Equation 2.7. In order for it to be met, the rising and falling transitions from each initial point have to balance one another by landing on points above and below it whose unconditional probability masses are large enough not to preclude sufficiently probable transitions to them.

On this basis, one might anticipate that no solution is possible for transitions from the first to the second time step in Table 2.5, because transitions from the state with the lowest price of 68.93 at time $k = 1$ have nowhere to go but up, with the lowest price at time $k = 2$ being 69.98. However, a situation like this one does not automatically guarantee infeasibility, because the latter price could be lower in drift adjusted terms based on the rest of the distributions.

On the other hand, even if all prices in the earlier state space have drift adjusted counterparts above and below them in the later one, infeasibility could occur if the unconditional probabilities of the extreme points in the later state space are too small. In this case, the probabilities of any transitions leading to the extreme points are constrained not to push the unconditional probabilities of the extreme points up too high, so they can be of little help in balancing the alternative transitions from their points of origin in order to meet the martingale condition.

Anecdotal evidence suggests that infeasibility is rare if the state prices are derived from lognormal or qualitatively similar continuous marginal distributions in the manner set out in Section 3.1, which can always be done automatically even for distributions with no explicit formula. Infeasibility is a more frequent problem when the state prices are chosen manually without due care and attention to the martingale constraint. The flexibility to choose the state price points manually is nevertheless an option if this additional responsibility is acceptable.

2.3 Scaling up the implied willow tree

Lattices with larger state spaces are necessary for achieving greater accuracy in derivatives valuation problems, but as noted above, the time and memory requirements for the implied willow tree become increasingly burdensome as the size of the state spaces increases. In this section, the crossover lattice method is developed for coping with larger state spaces more efficiently than possible by implied willow trees alone.

2.3.1 Intuitive overview

Because the size of the linear programming problem needing to be solved depends crucially on the number of state prices at each time step, it is natural to consider decreasing it by splitting the state spaces somehow into smaller systems that can be solved separately and then combined. An obvious choice would be to solve for the transition probabilities separately between the high prices at time k and the high prices at time $k + 1$, and similarly for the low prices. However, this approach in itself would be unlikely to yield satisfactory results, imposing as it does a global partition on the state space with zero probability of crossing it.

One may envision overcoming this drawback by a more sophisticated scheme whereby the state spaces are divided into nearly disjoint subsets but allowed some small non-empty intersection between them. This approach in itself would lead immediately to the difficulty of reconciling the different sets of outgoing transition probabilities obtained for each state belonging to multiple subsets. However, building an implied willow tree requires solving for the transition probabilities between each pair of consecutive state spaces in a series, and effective intermingling of the states can be achieved by taking a disjoint partition on each state space when it is the source but an intersecting one when it is the target. With unique outgoing transition probabilities from each state thus obtained, it would remain only to ensure that the incoming ones do not conflict.

To this end, the lattice can be compared to an electrical circuit or network flow diagram, with the transitions corresponding to the wires or conduits, and the states being the nodes. The transition probabilities in the lattice are proportional to a current or flux of a fluid through it that is collected and dispersed at each node but not created or destroyed, and the unconditional probability of each state is proportional to the total current or flux through it.

If a target state ω_j belongs to two subsets, one could therefore obtain incoming transition probabilities p_{ij} consistent with its total probability q_j by solving each subproblem as if q_j were half of its actual value, so that the combined flux from both solutions would add up to the correct value. It would be equally effective to assume one third the probability in one case and two thirds in the other, or any other pair of fractions summing to unity. This freedom of choice may be put to good use in meeting other constraints yet to be considered.

Returning to the network flow analogy, one might also surmise correctly that the total probability flowing into all of the earlier states (normally unity) must always equal the total out of the later states if any set of transition probabilities consistent with them is to exist. With the earlier states separated into disjoint sets, the total probability of the states in each earlier set is fixed, and only the later sets are adjustable. Therefore, the total probability of the states in the later sets must be adjusted to match the earlier sets by judiciously weighting the probabilities of the states in their intersection, for which there is some freedom of choice as noted above.

A final consideration is that of the martingale condition. It is a requirement that the discounted expected value of the asset in each state should be its present price. This constraint is normally met by construction when the entire state space is solved as a willow tree, but will be met when a problem is subdivided only if the drifts in each subproblem are matched to that of the whole. This matching may also be achieved by suitable allocation of probabilities on the states that belong to more than one subset.

2.3.2 Explicit construction

It is now possible to consider developing this line of reasoning along more quantitative lines. Although the previous section begins with the idea of dividing the state spaces between two sets, namely the high and low prices, it can be shown that this approach leaves the problem of choosing the weights overconstrained, with no solution under general conditions. It is therefore necessary to choose at least three subsets for each state space, as shown in Figure 2.3, with the earlier subsets disjoint and the later ones intersecting as discussed above.

Suppose three sets of states are chosen at time k and three at time $k + 1$ with cardinalities as nearly equal as possible, and consecutive prices for the states in each. Leaving aside momentarily the question of whether this choice is best, let

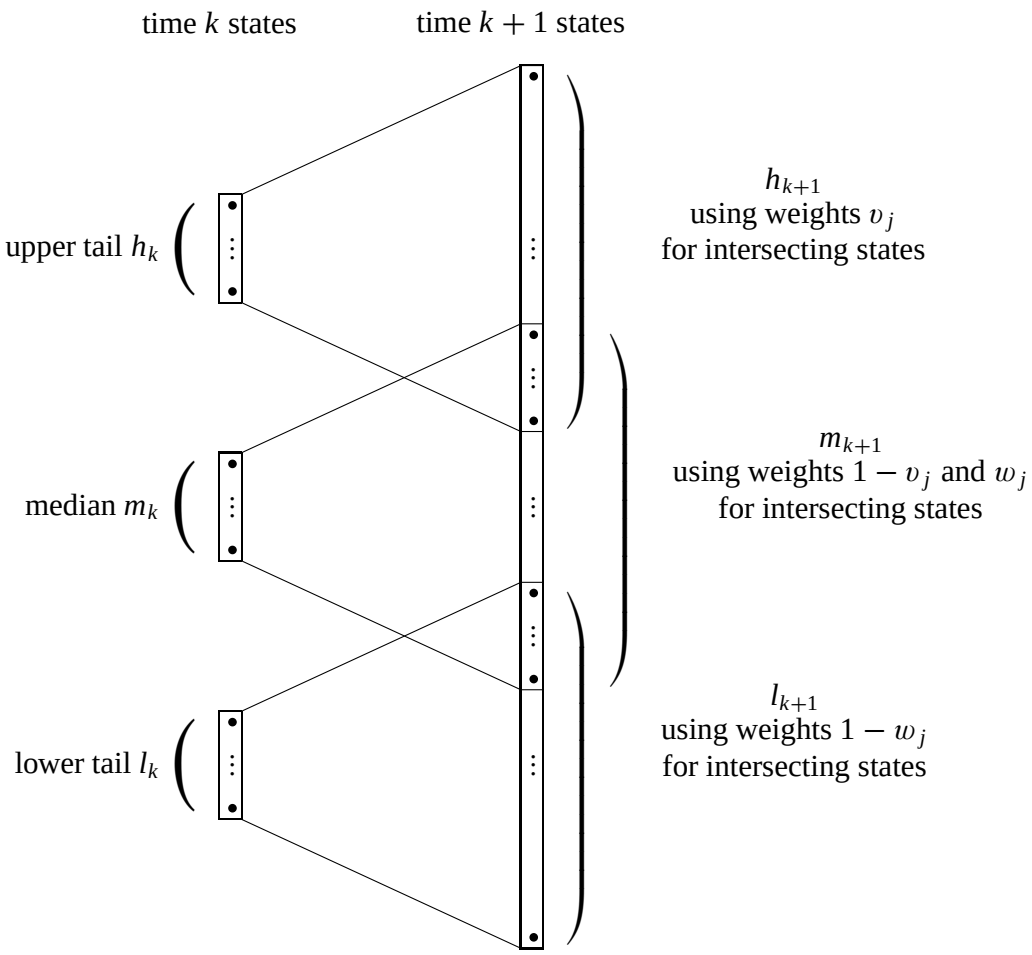

Figure 2.3: diagrammatic representation of the way a pair of consecutive state spaces is decomposed into three subproblems solvable independently for state transition probabilities

$l_k \subset \mathbb{N}$ be the set of indices of states belonging to the subset with low prices at time k, let $h_k \subset \mathbb{N}$ be that of the subset of states with high prices at time k, and let m_k contain the remaining indices at time k. A similar convention applies to l_{k+1}, m_{k+1} and h_{k+1}, except that

$$l_k \cap m_k = m_k \cap h_k = l_k \cap h_k = \emptyset \tag{2.9}$$

whereas

$$l_{k+1} \cap m_{k+1} \neq \emptyset \tag{2.10}$$
$$m_{k+1} \cap h_{k+1} \neq \emptyset \tag{2.11}$$
$$l_{k+1} \cap h_{k+1} = \emptyset \tag{2.12}$$

and the issue of how the intersections are chosen is also temporarily deferred (but see Section 2.5.2, page 38). The sets of weights

$$\{v_j \in [0,1] \mid j \in h_{k+1} \cap m_{k+1}\} \tag{2.13}$$
$$\{w_j \in [0,1] \mid j \in l_{k+1} \cap m_{k+1}\} \tag{2.14}$$

are required to make the total probabilities q_i^k and q_j^{k+1} satisfy

$$\sum_{i \in h_k} q_i^k = \sum_{j \in h_{k+1} \cap m_{k+1}} v_j q_j^{k+1} + \sum_{j \in h_{k+1} - m_{k+1}} q_j^{k+1} \tag{2.15}$$

$$\sum_{i \in m_k} q_i^k = \sum_{j \in h_{k+1} \cap m_{k+1}} (1 - v_j) q_j^{k+1} + \sum_{j \in m_{k+1} - h_{k+1} - l_{k+1}} q_j^{k+1} + \sum_{j \in m_{k+1} \cap l_{k+1}} w_j q_j^{k+1}$$

$$\sum_{i \in l_k} q_i^k = \sum_{j \in m_{k+1} \cap l_{k+1}} (1 - w_j) q_j^{k+1} + \sum_{j \in l_{k+1} - m_{k+1}} q_j^{k+1} \tag{2.16}$$

which is to say that the total probability of each set of states on the left side of Figure 2.3 must equal the total probability of each corresponding set on the right when the weights of the probabilities are taken into account on the states belonging to their intersections. These weights must also make the drifts between each pair of subsets match the prevailing discount rate d_k,

$$\frac{1}{d_k} \sum_{i \in h_k} q_i^k S_i^k = \sum_{j \in h_{k+1} \cap m_{k+1}} v_j q_j^{k+1} S_j^{k+1} + \sum_{j \in h_{k+1} - m_{k+1}} q_j^{k+1} S_j^{k+1} \tag{2.17}$$

$$\frac{1}{d_k} \sum_{i \in m_k} q_i^k S_i^k = \sum_{j \in h_{k+1} \cap m_{k+1}} (1 - v_j) q_j^{k+1} S_j^{k+1} + \sum_{j \in m_{k+1} - h_{k+1} - l_{k+1}} q_j^{k+1} S_j^{k+1} + \sum_{j \in m_{k+1} \cap l_{k+1}} w_j q_j^{k+1} S_j^{k+1}$$

$$\frac{1}{d_k} \sum_{i \in l_k} q_i^k S_i^k = \sum_{j \in m_{k+1} \cap l_{k+1}} (1 - w_j) q_j^{k+1} S_j^{k+1} + \sum_{j \in l_{k+1} - m_{k+1}} q_j^{k+1} S_j^{k+1} \tag{2.18}$$

where d_k is given by Equation 2.6.

Every term in these equations except w_j and v_j is a known constant, making them simply a linear system of equations in the weights. However, the constraint that the weights must each lie between zero and one makes it a linear programming problem. Any linear programming problem needs a cost to minimize. If these conditions do not determine the weights w_j and v_j completely, then it would be desirable to minimize the cost

$$C_{vw} = \sum_{j \in h_{k+1} \cap m_{k+1}} v_j - \sum_{j \in m_{k+1} \cap l_{k+1}} w_j \tag{2.19}$$

because low values of v_j and high values of w_j mean less of the modal probability gets mixed into the tails, leading to lower skew in the tails and subproblems more similar to the original problem. Another variation is to assess a cost inversely proportional to the probability. In practice the cost function appears to make no significant difference, suggesting that that there is very limited discretion available in the choice of the weights after these conditions are met.

2.3.3 Subproblems to be solved

When the weights v_j and w_j are known, it is straightforward to set up the three linear programming problems to be solved for the transition probabilities in lieu of the original one. For the transition probabilities originating from states in the upper tail h_k, it is necessary to minimize

$$C_h = \sum_{i \in h_k} \sum_{j \in h_{k+1}} p_{ij}^k \left| \ln \frac{d_k S_j^{k+1}}{S_i^k} \right|^\alpha \tag{2.20}$$

(cf. Equation 2.8) subject to the constraints

$$\forall i \in h_k, \sum_{j \in h_{k+1}} p_{ij}^k = 1 \tag{2.21}$$

$$\forall j \in h_{k+1} - m_{k+1}, \sum_{i \in h_k} p_{ij}^k q_i^k = q_j^{k+1} \tag{2.22}$$

$$\forall j \in h_{k+1} \cap m_{k+1}, \sum_{i \in h_k} p_{ij}^k q_i^k = v_j q_j^{k+1} \tag{2.23}$$

$$\forall i \in h_k, \; d_k \sum_{j \in h_{k+1}} p_{ij}^k S_j^{k+1} = S_i^k \tag{2.24}$$

where the probabilities p_{ij}^k are of course also constrained to be non-negative. For the lower tail, it is necessary to minimize

$$C_l = \sum_{i \in l_k} \sum_{j \in l_{k+1}} p_{ij}^k \left| \ln \frac{d_k S_j^{k+1}}{S_i^k} \right|^\alpha \tag{2.25}$$

subject to the constraints

$$\forall i \in l_k, \sum_{j \in l_{k+1}} p_{ij}^k = 1 \tag{2.26}$$

$$\forall j \in l_{k+1} - m_{k+1}, \sum_{i \in h_k} p_{ij}^k q_i^k = q_j^{k+1} \tag{2.27}$$

$$\forall j \in l_{k+1} \cap m_{k+1}, \sum_{i \in h_k} p_{ij}^k q_i^k = (1 - w_j) q_j^{k+1} \tag{2.28}$$

$$\forall i \in l_k, \; d_k \sum_{j \in l_{k+1}} p_{ij}^k S_j^{k+1} = S_i^k \tag{2.29}$$

For the modal region, it is necessary to minimize

$$C_m = \sum_{i \in m_k} \sum_{j \in m_{k+1}} p_{ij}^k \left| \ln \frac{d_k S_j^{k+1}}{S_i^k} \right|^\alpha \tag{2.30}$$

subject to the constraints

$$\forall i \in m_k, \sum_{j \in m_{k+1}} p_{ij}^k = 1 \tag{2.31}$$

$$\forall j \in m_{k+1} \cap h_{k+1}, \sum_{i \in m_k} p_{ij}^k q_i^k = (1 - v_j) q_j^{k+1} \tag{2.32}$$

$$\forall j \in m_{k+1} - h_{k+1} - l_{k+1}, \sum_{i \in m_k} p_{ij}^k q_i^k = q_j^{k+1} \tag{2.33}$$

$$\forall j \in l_{k+1} \cap m_{k+1}, \sum_{i \in m_k} p_{ij}^k q_i^k = w_j q_j^{k+1} \tag{2.34}$$

$$\forall i \in m_k, \; d_k \sum_{j \in m_{k+1}} p_{ij}^k S_j^{k+1} = S_i^k \tag{2.35}$$

The solution of these three systems of equations independently of one another will provide transition probabilities p_{ij}^k that are consistent with the original pair of state spaces from which these were derived.

This construction pertains to a single level of decomposition from one problem to three subproblems, but it is natural to generalize it to a hierarchical series of decompositions. One may first decompose a problem into three subproblems, then follow a similar procedure to decompose each of those into three more, and so on, perhaps stopping only when the size of the subproblems reaches some chosen upper bound. This essential technique allows crossover lattices to be constructed for large state spaces without requiring the solution of large linear programming problems.

2.4 Informal complexity analysis

The purpose of the exercise in the previous section has been to avoid solving a large linear programming problem by subdividing it into smaller ones, but it has turned out to require the solution of another linear programming problem just to effect the subdivision. It is prudent to ask whether or not the benefit outweighs the cost, and preferable if a credible answer can be given informally.

The number of variables in the linear programming problem that is solved for the weights is equal to the number of weights, namely

$$|m_{k+1} \cap l_{k+1}| + |m_{k+1} \cap h_{k+1}| \tag{2.36}$$

This number is bounded by the number of states at time $k+1$, and could be an order of magnitude less. If a feasible solution for the weights is found, one avoids solving a problem with

$$|h_k \cup m_k \cup l_k| \times |h_{k+1} \cup m_{k+1} \cup l_{k+1}| \tag{2.37}$$

variables, and instead solves three problems, with numbers of variables

$$|h_k| \times |h_{k+1}| \tag{2.38}$$
$$|m_k| \times |m_{k+1}| \tag{2.39}$$
$$|l_k| \times |l_{k+1}| \tag{2.40}$$

Given that h_{k+1}, m_{k+1} and l_{k+1} are non-empty, the right side of 2.37 must be strictly greater than any of those of 2.38 through 2.40. In the most optimistic scenario, it is approximately three times each of their individual sizes. With the cardinalities of h_k, m_k and l_k constructed to be one third that of their union, each of the subproblems therefore has one ninth as many variables. The result is a

considerable savings even though three of them have to be solved, especially in view of the fact that the time and memory required to solve a linear programming problem increases more than linearly with the number of variables.

In a more pessimistic scenario, $|h_{k+1}|$ and $|l_{k+1}|$ would each approach half of N_{k+1} as the boundaries of intersections between the tails and the mode in Figure 2.3 creep toward the middle. However, h_{k+1} and l_{k+1} are disjoint by construction (Equation 2.12) so their sum would not exceed it. In this case, two of the subproblems would have one sixth the number of variables of the original. The remaining subproblem could have up to a third the number of variables if the outer boundaries of the intersecting regions extend far into the tails, but it would be unlikely insofar as the outlying points have insignificant probabilities compared to the mode and therefore can have little effect in drift matching whatever their weights. One is still ahead by a wide margin.

In the case of a system of one hundred points in each state space, the original problem has 100^2 or ten thousand variables, the optimistic case has three problems with a bit more than a thousand each, and the pessimistic one has two problems closer to two thousand and one closer to three. For the case of two hundred points, these numbers increase by a factor of four, but there is of course a bigger difference between a forty-thousand and a four-thousand variable linear programming problem than between ten thousand and one thousand.

Returning to the original question in this section of whether the benefit of computing the weights is worth the cost, one may conclude that the cost of solving a linear programming problem with typically less than a hundred or two hundred variables is absolutely insignificant compared to the potential benefit achievable for state spaces of that size, and the advantage only increases as the state spaces grow larger.

2.5 Decomposition strategies and branching patterns

In Section 2.3.2, a set of weights v_j and w_j is derived to match the drifts and probabilities in a state space when subdividing it for easier computation of transition probabilities. This derivation presupposes a choice of disjoint subsets h_k, m_k and l_k on the earlier state space, and of overlapping subsets h_{k+1}, m_{k+1}, and l_{k+1} on the succeeding one. Although the exact choice of subsets is unspecified, a bad choice may preclude a solution due to an infeasible linear programming problem.

Alternatively, a choice that may allow for decomposition could preclude any feasible solution of transition probabilities in the subspaces. Yet another choice may allow for both decomposition and feasible subproblems, but not for any further decomposition, as one would prefer when constructing crossover lattices for very large state spaces. The matter of choosing subsets and points of intersection on the state spaces with a view to avoiding these pitfalls is the subject of this section.

In particular, three strategies for decomposing the state space are considered, designated as the cut, the deal, and the skim.

- The cut is most intuitive but least useful.
- The deal is sometimes more conducive to scalability than the cut and more efficient than the skim.
- The skim is used when all else fails, being least efficient but usually the most scalable.

2.5.1 Warming up with a willow tree

A numerical example is helpful for shedding light on these issues. Table 2.6 shows a pair of hypothetical state spaces for two consecutive time steps. The state prices are chosen in a geometric progression around means of roughly 100, with a slight upward drift and increased variance from one state space to the next. The unconditional probabilities are based on lognormal distributions according to the discretization method described in Section 3.1. Although more congenial than one might expect in practice, this example suffices for the present purpose of revealing any symmetries that may occur in the partitions or branching patterns.

It is reassuring to note that implied willow tree transition probabilities can be obtained for this pair of state spaces in the manner described in Section 2.2. As an aid to intuition, a diagrammatic representation of the branching pattern and transition probabilities is shown in Figure 2.4. The points on the left of the figure represent the earlier states, and the points on the right represent the later ones. States with lower prices are depicted toward the bottom of both columns. The points are plotted on a drift adjusted logarithmic scale, in that both means are at the same level. The apparently uniform vertical spacing in each column is a consequence of the state prices having been chosen in geometric progression. The lines represent transitions from earlier to later states, and their thickness is proportional to the transition probability as far as the print quality allows.

	sooner		later	
state	price	probability	price	probability
0	80.94752249	1.37501078×10^{-7}	74.30677854	1.37624154×10^{-7}
1	81.84720792	4.25570010×10^{-7}	75.47931600	4.27175214×10^{-7}
2	82.75689284	1.58413734×10^{-6}	76.67035574	1.59318346×10^{-6}
3	83.67668838	5.47966894×10^{-6}	77.88018971	5.52030371×10^{-6}
4	84.60670693	1.76139802×10^{-5}	79.10911448	1.77704994×10^{-5}
5	85.54706210	5.26143673×10^{-5}	80.35743129	5.31468790×10^{-5}
6	86.49786878	1.46048392×10^{-4}	81.62544614	1.47672391×10^{-4}
7	87.45924314	3.76735591×10^{-4}	82.91346987	3.81211624×10^{-4}
8	88.43130262	9.03079042×10^{-4}	84.22181820	9.14281793×10^{-4}
9	89.41416598	2.01171223×10^{-3}	85.55081186	2.03724318×10^{-3}
10	90.40795331	4.16445911×10^{-3}	86.90077662	4.21751104×10^{-3}
11	91.41278602	8.01134806×10^{-3}	88.27204340	8.11188441×10^{-3}
12	92.42878687	1.43221554×10^{-2}	89.66494833	1.44957601×10^{-2}
13	93.45607999	2.37940514×10^{-2}	91.07983286	2.40666376×10^{-2}
14	94.49479088	3.67355900×10^{-2}	92.51704382	3.71232862×10^{-2}
15	95.54504645	5.27066663×10^{-2}	93.97693351	5.32028979×10^{-2}
16	96.60697501	7.02757025×10^{-2}	95.45985981	7.08407692×10^{-2}
17	97.68070630	8.70777283×10^{-2}	96.96618621	8.76378050×10^{-2}
18	98.76637150	1.00270291×10^{-1}	98.49628196	1.00730505×10^{-1}
19	99.86410325	1.07300492×10^{-1}	100.05052214	1.07570403×10^{-1}
20	100.97403566	1.06707825×10^{-1}	101.62928774	1.06730297×10^{-1}
21	102.09630433	9.86181205×10^{-2}	103.23296577	9.83890053×10^{-2}
22	103.23104637	8.47000701×10^{-2}	104.86194932	8.42694280×10^{-2}
23	104.37840043	6.76048755×10^{-2}	106.51663772	6.70592599×10^{-2}
24	105.53850667	5.01464253×10^{-2}	108.19743658	4.95807645×10^{-2}
25	106.71150683	3.45676682×10^{-2}	109.90475792	3.40591546×10^{-2}
26	107.89754421	2.21446425×10^{-2}	111.63902024	2.17380724×10^{-2}
27	109.09676372	1.31836602×10^{-2}	113.40064868	1.28906548×10^{-2}
28	110.30931187	7.29410882×10^{-3}	115.19007505	7.10224855×10^{-3}
29	111.53533680	3.75039895×10^{-3}	117.00773801	3.63566437×10^{-3}
30	112.77498830	1.79205649×10^{-3}	118.85408312	1.72917413×10^{-3}
31	114.02841781	7.95783345×10^{-4}	120.72956296	7.64118335×10^{-4}
32	115.29577847	3.28402732×10^{-4}	122.63463729	3.13725124×10^{-4}
33	116.57722513	1.25946823×10^{-4}	124.56977309	1.19675301×10^{-4}
34	117.87291433	4.48885839×10^{-5}	126.53544471	4.24156376×10^{-5}
35	119.18300438	1.48679967×10^{-5}	128.53213402	1.39673279×10^{-5}
36	120.50765532	4.57653118×10^{-6}	130.56033045	4.27332645×10^{-6}
37	121.84702901	1.30914368×10^{-6}	132.62053117	1.21474105×10^{-6}
38	123.20128907	3.48020275×10^{-7}	134.71324122	3.20823125×10^{-7}
39	124.57060095	1.10949314×10^{-7}	136.83897357	1.01387277×10^{-7}

Table 2.6: a 40 by 40 point state space pair with lognormal prices

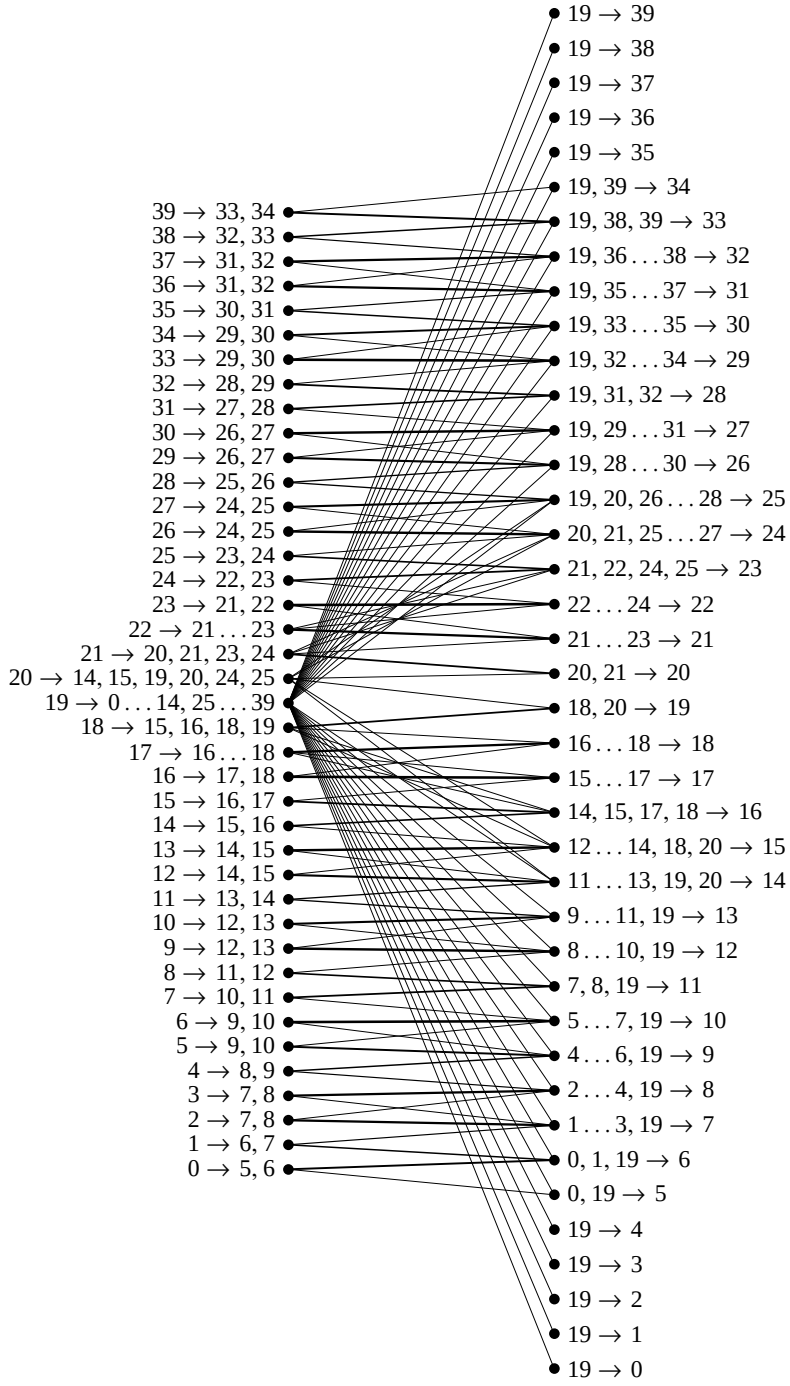

Figure 2.4: diagrammatic representation of willow tree transitions implied by Table 2.6, with heavier lines indicating higher probabilities

The data on which this diagram are based may be found in Appendix A in Table A.3. It is worthwhile to verify that these results satisfy the probability and martingale constraints given by Equations 2.4, 2.5, and 2.7 by which can be done by direct computation in a manner analogous to Tables 2.3 and 2.4. These results are to be found in the appendix as well in Tables A.4 and A.5.

The figure illustrates that most states evolve to their nearest drift adjusted future counterparts with high probability, but have a slight tendency to wander usually no further than their nearest neighbors. A notable exception is state number 19, which branches prolifically to many extremes. This result is understandable in view of the cost constraint in Equation 2.8. Costs are globally minimized by having most transitions jump no further than necessary, but to the extent that non-local jumps are necessary to meet the constraints, it is best to source most of them from one state. In this way, the probability of each individual long jump is diminished by virtue of it being one of many, and with low probability comes reduced cost. The source state should be close to the center so that many of the jumps from it are further shortened and those at the extremes are minimized.

2.5.2 A simple decomposition – the cut

A first attempt at decomposing the state spaces may proceed along the same lines as the intuitive description in Section 2.3.2 and depicted in Figure 2.3. In this case, one simply divides the state among low, medium, and high prices in three roughly equal parts, similarly to cutting a deck of cards.

Table 2.7 shows some of the implications of this decomposition. The earlier states are cut into three disjoint sets, and the later states are cut into three sets sharing some state prices in common, which are shown in bold. The unconditional probabilities differ from those in Table 2.6 because they are scaled up within each subset to sum to unity. The probabilities for the bold prices are not only scaled but adjusted to their weighted multiples per Section 2.3.2. The combined effects of these transformations are

- all three pairs of state spaces have identical drifts
- within each pair, both sets of probabilities are scaled by the same factor
- when each system is solved separately, the contribution to the probabilities of the shared states from each system will sum to the right value

sooner			later		
state	price	probability	state	price	probability
0	80.94752249	4.58132399×10^{-6}	0	74.30677854	4.58542471×10^{-6}
1	81.84720792	1.41793369×10^{-5}	1	75.47931600	1.42328198×10^{-5}
2	82.75689284	5.27810147×10^{-5}	2	76.67035574	5.30824175×10^{-5}
3	83.67668838	1.82574124×10^{-4}	3	77.88018971	1.83928012×10^{-4}
4	84.60670693	5.86870675×10^{-4}	4	79.10911448	5.92085654×10^{-4}
5	85.54706210	1.75302963×10^{-3}	5	80.35743129	1.77077210×10^{-3}
6	86.49786878	4.86610734×10^{-3}	6	81.62544614	4.92021647×10^{-3}
7	87.45924314	1.25522493×10^{-2}	7	82.91346987	1.27013838×10^{-2}
8	88.43130262	3.00892019×10^{-2}	8	84.22181820	3.04624603×10^{-2}
9	89.41416598	6.70271513×10^{-2}	9	85.55081186	6.78778031×10^{-2}
10	90.40795331	1.38753360×10^{-1}	10	86.90077662	1.40520968×10^{-1}
11	91.41278602	2.66925771×10^{-1}	**11**	**88.27204340**	8.51341779×10^{-3}
12	92.42878687	4.77192144×10^{-1}	**15**	**93.97693351**	7.32385064×10^{-1}
13	93.45607999	2.52416566×10^{-2}	**11**	**88.27204340**	8.33434108×10^{-3}
14	94.49479088	3.89705449×10^{-2}	12	89.66494833	1.53776671×10^{-2}
15	95.54504645	5.59132849×10^{-2}	13	91.07983286	2.55308267×10^{-2}
16	96.60697501	7.45512029×10^{-2}	14	92.51704382	3.93818282×10^{-2}
17	97.68070630	9.23754465×10^{-2}	**15**	**93.97693351**	3.31210229×10^{-2}
18	98.76637150	1.06370631×10^{-1}	16	95.45985981	7.51506477×10^{-2}
19	99.86410325	1.13828542×10^{-1}	17	96.96618621	9.29695976×10^{-2}
20	100.97403566	1.13199817×10^{-1}	18	98.49628196	1.06858844×10^{-1}
21	102.09630433	1.04617944×10^{-1}	19	100.05052214	1.14114874×10^{-1}
22	103.23104637	8.98531340×10^{-2}	20	101.62928774	1.13223657×10^{-1}
23	104.37840043	7.17178856×10^{-2}	21	103.23296577	1.04374890×10^{-1}
24	105.53850667	5.31972815×10^{-2}	22	104.86194932	8.93962921×10^{-2}
25	106.71150683	3.66707291×10^{-2}	23	106.51663772	7.11390754×10^{-2}
26	107.89754421	2.34918995×10^{-2}	**24**	**108.19743658**	3.08608543×10^{-2}
			25	109.90475792	3.61312780×10^{-2}
			26	111.63902024	2.30605941×10^{-2}
			27	113.40064868	1.36749087×10^{-2}
			28	**115.19007505**	7.29880120×10^{-3}
27	109.09676372	4.82273890×10^{-1}	**24**	**108.19743658**	7.49540235×10^{-1}
28	110.30931187	2.66827131×10^{-1}	**28**	**115.19007505**	8.12221201×10^{-3}
29	111.53533680	1.37194031×10^{-1}	29	117.00773801	1.32996904×10^{-1}
30	112.77498830	6.55555468×10^{-2}	30	118.85408312	6.32552355×10^{-2}
31	114.02841781	2.91106963×10^{-2}	31	120.72956296	2.79523528×10^{-2}
32	115.29577847	1.20133605×10^{-2}	32	122.63463729	1.14764362×10^{-2}
33	116.57722513	4.60728379×10^{-3}	33	124.56977309	4.37786411×10^{-3}
34	117.87291433	1.64207751×10^{-3}	34	126.53544471	1.55161421×10^{-3}
35	119.18300438	5.43888909×10^{-4}	35	128.53213402	5.10941382×10^{-4}
36	120.50765532	1.67414925×10^{-4}	36	130.56033045	1.56323338×10^{-4}
37	121.84702901	4.78900246×10^{-5}	37	132.62053117	4.44366649×10^{-5}
38	123.20128907	1.27309934×10^{-5}	38	134.71324122	1.17360895×10^{-5}
39	124.57060095	4.05865716×10^{-6}	39	136.83897357	3.70886656×10^{-6}

Table 2.7: a straightforward cut separating states mainly between low, medium, and high prices

The remaining question is that of how the shared states are chosen. It may be noted that contrary to the intuitive picture of Figure 2.3, they do not form contiguous blocks. The precise choice of shared states is not known in advance but is determined on the fly in the course of solving for the weights. For this example, the set l_{k+1} would contain state numbers 0 to 15, m_{k+1} would contain state numbers 11 to 28 and h_{k+1} would contain state numbers 24 to 39. The intersections are therefore $l_{k+1} \cap m_{k+1} = \{11 \ldots 15\}$, and $h_{k+1} \cap m_{k+1} = \{24 \ldots 28\}$. These sets are determined by iteratively searching for the smallest overlap that allows a feasible solution for the weights v_j and w_j. When the solution for the weights is found, it shows that w_{12}, w_{13} and w_{14} are all equal to 1, while v_{25}, v_{26} and v_{26} are all equal to 0. Only the points shown in bold have weights other than 0 or 1. In effect, the linear programming solver indicates that points 12, 13, 14, 25, 26, and 27 do not really belong in the intersection, so they are removed.

Having performed the decomposition shown in Table 2.7, one may attempt to obtain willow tree transition probabilities for each subproblem. In this case, the subproblems are feasible, and the transition probabilities are reported in Table A.6 in terms of the state numbers from the original data in Table 2.6. Similarly to the case of the undivided willow tree, these probabilities may be subject to verification as shown in Tables A.7 and A.8, which are readily seen to be correct within machine precision.

An intuitive overview of the lattice structure is afforded by the diagram in Figure 2.5. The diagram provides some insight into the reason a feasible solution requires some of the shared points to cross over toward the center of the future state space rather than sticking to the boundaries of the tails. As noted in Section 2.2.2 on page 24, in order for the martingale condition to be met (Equation 2.7) it is necessary for the rising and falling transitions from each initial point to balance one another by landing on points above and below whose unconditional probability masses are large enough not to preclude sufficiently probable transitions to them. Because the probabilities in the tails are increasing or decreasing monotonically, the only way to balance the rising and falling transitions from the most probable top ends of the tails is to allow a wider mix.

Although this decomposition strategy is simple and direct, it has the drawback of not being composable with itself, because it relies on a marginal distribution being at least somewhat thinner at the tails than in the center. It may sometimes be possible to cut a state space and then cut the center subspace again, but not the tails, because these become heavily skewed on the first cut, often with the most probable point at an extreme end. This shortcoming makes the strategy

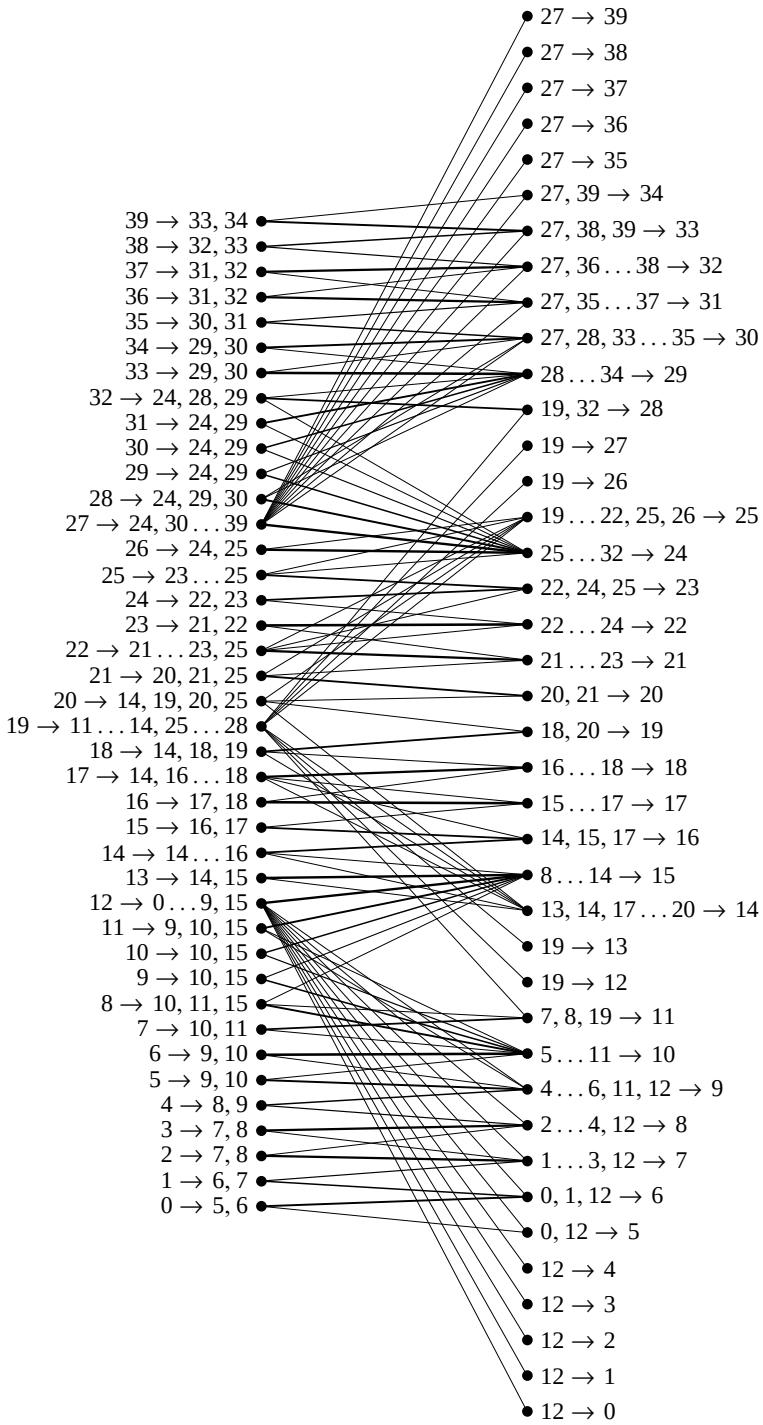

Figure 2.5: diagrammatic representation of cut willow tree transitions based on Table 2.7

ineffective by itself for hierarchical decompositions of large state spaces, so a more sophisticated idea is needed.

2.5.3 A fairer decomposition - the deal

Another way to make three piles of a deck of cards is to deal it. The analogous decomposition strategy for state spaces would include nearly a full range of prices in each subset, and maintain the general shape of the original distribution. In so doing, it may be a more promising candidate for use in a hierarchical series of decompositions.

A single iteration of the deal decomposition is shown in Table 2.8. Having demonstrated in the Section 2.5.2 on page 38 that it is able to make an appropriate choice of shared states automatically, the linear programming solver is given a freer hand in this strategy to select them throughout the state space. In this strategy, the earlier state space is dealt into three equal piles, and the later into five. The first, third, and fifth piles play the rôles of l_{k+1}, m_{k+1}, and h_{k+1} in the cut decomposition, and the remaining piles serve as the intersections.

Other than that, similar points to the discussion of the cut decomposition apply to the deal. The resulting transition probabilities are shown in Table A.9. It is prudent to verify that these transition probabilities are valid, which is done in Tables A.10 and A.11.

The diagram in Figure 2.6 shows that the dealing decomposition results in a pattern similar to three willow trees slightly displaced with one another and overlayed, as one might expect.

2.5.4 Cutting a deal for nested decomposition

By dealing the state spaces first into three piles as in Section 2.5.3 and then cutting each pile three ways as in Section 2.5.2, one arrives at the decomposition into nine subproblems as shown in Table 2.9. The benefit of pursuing a hierarchical decomposition strategy may be appreciated even in an example as small as this one by noting that the largest remaining subproblem has only five by nine points, requiring the solution of a linear programming problem in 45 variables, compared to 1600 variables for the problem in its original 40 by 40 point form.

sooner			later		
state	price	probability	state	price	probability
0	80.94752249	4.12503165×10^{-7}	0	74.30677854	4.12872394×10^{-7}
3	83.67668838	1.64390041×10^{-5}	1	75.47931600	1.28152543×10^{-6}
6	86.49786878	4.38145105×10^{-4}	5	80.35743129	1.59440611×10^{-4}
9	89.41416598	6.03513572×10^{-3}	6	81.62544614	4.43017100×10^{-4}
12	92.42878687	4.29664591×10^{-2}	10	86.90077662	1.26525311×10^{-2}
15	95.54504645	1.58119973×10^{-1}	11	88.27204340	2.43356493×10^{-2}
18	98.76637150	3.00810825×10^{-1}	15	93.97693351	1.59608668×10^{-1}
21	102.09630433	2.95854313×10^{-1}	**16**	**95.45985981**	1.51005860×10^{-1}
24	105.53850667	1.50439251×10^{-1}	20	101.62928774	3.20190840×10^{-1}
27	109.09676372	3.95509743×10^{-2}	**21**	**103.23296577**	1.56676048×10^{-1}
30	112.77498830	5.37616860×10^{-3}	25	109.90475792	1.02177447×10^{-1}
33	116.57722513	3.77840406×10^{-4}	26	111.63902024	6.52142067×10^{-2}
36	120.50765532	1.37295913×10^{-5}	30	118.85408312	5.18752154×10^{-3}
39	124.57060095	3.32847886×10^{-7}	31	120.72956296	2.29235463×10^{-3}
			35	128.53213402	4.19019770×10^{-5}
			36	130.56033045	1.28199773×10^{-5}
1	81.84720792	1.27671013×10^{-6}	2	76.67035574	4.77955074×10^{-6}
4	84.60670693	5.28419447×10^{-5}	7	82.91346987	1.14363496×10^{-3}
7	87.45924314	1.13020686×10^{-3}	12	89.66494833	4.34872837×10^{-2}
10	90.40795331	1.24933783×10^{-2}	**16**	**95.45985981**	6.15164276×10^{-2}
13	93.45607999	7.13821595×10^{-2}	17	96.96618621	2.62913435×10^{-1}
16	96.60697501	2.10827123×10^{-1}	**18**	**98.49628196**	1.41782159×10^{-1}
19	99.86410325	3.21901500×10^{-1}	**21**	**103.23296577**	1.38490952×10^{-1}
22	103.23104637	2.54100230×10^{-1}	22	104.86194932	2.52808303×10^{-1}
25	106.71150683	1.03703012×10^{-1}	**23**	**106.51663772**	5.82362385×10^{-2}
28	110.30931187	2.18823281×10^{-2}	27	113.40064868	3.86719672×10^{-2}
31	114.02841781	2.38735021×10^{-3}	32	122.63463729	9.41175443×10^{-4}
34	117.87291433	1.34665762×10^{-4}	**33**	**124.56977309**	3.98598855×10^{-20}
37	121.84702901	3.92743132×10^{-6}	37	132.62053117	3.64422343×10^{-6}
2	82.75689284	4.75241243×10^{-6}	3	77.88018971	1.65609126×10^{-5}
5	85.54706210	1.57843116×10^{-4}	4	79.10911448	5.33115030×10^{-5}
8	88.43130262	2.70923736×10^{-3}	8	84.22181820	2.74284562×10^{-3}
11	91.41278602	2.40340463×10^{-2}	9	85.55081186	6.11173008×10^{-3}
14	94.49479088	1.10206779×10^{-1}	13	91.07983286	7.21999190×10^{-2}
17	97.68070630	2.61233208×10^{-1}	14	92.51704382	1.11369868×10^{-1}
20	100.97403566	3.20123502×10^{-1}	**18**	**98.49628196**	1.60409382×10^{-1}
23	104.37840043	2.02814644×10^{-1}	19	100.05052214	3.22711237×10^{-1}
26	107.89754421	6.64339334×10^{-2}	**23**	**106.51663772**	1.42941558×10^{-1}
29	111.53533680	1.12511978×10^{-2}	24	108.19743658	1.48742306×10^{-1}
32	115.29577847	9.85208283×10^{-4}	28	115.19007505	2.13067475×10^{-2}
35	119.18300438	4.46039939×10^{-5}	29	117.00773801	1.09069941×10^{-2}
38	123.20128907	1.04406092×10^{-6}	**33**	**124.56977309**	3.59025935×10^{-4}
			34	126.53544471	1.27246924×10^{-4}
			38	134.71324122	9.62469460×10^{-7}
			39	136.83897357	3.04161858×10^{-7}

Table 2.8: decomposition based on dealing the prices fairly among subsets

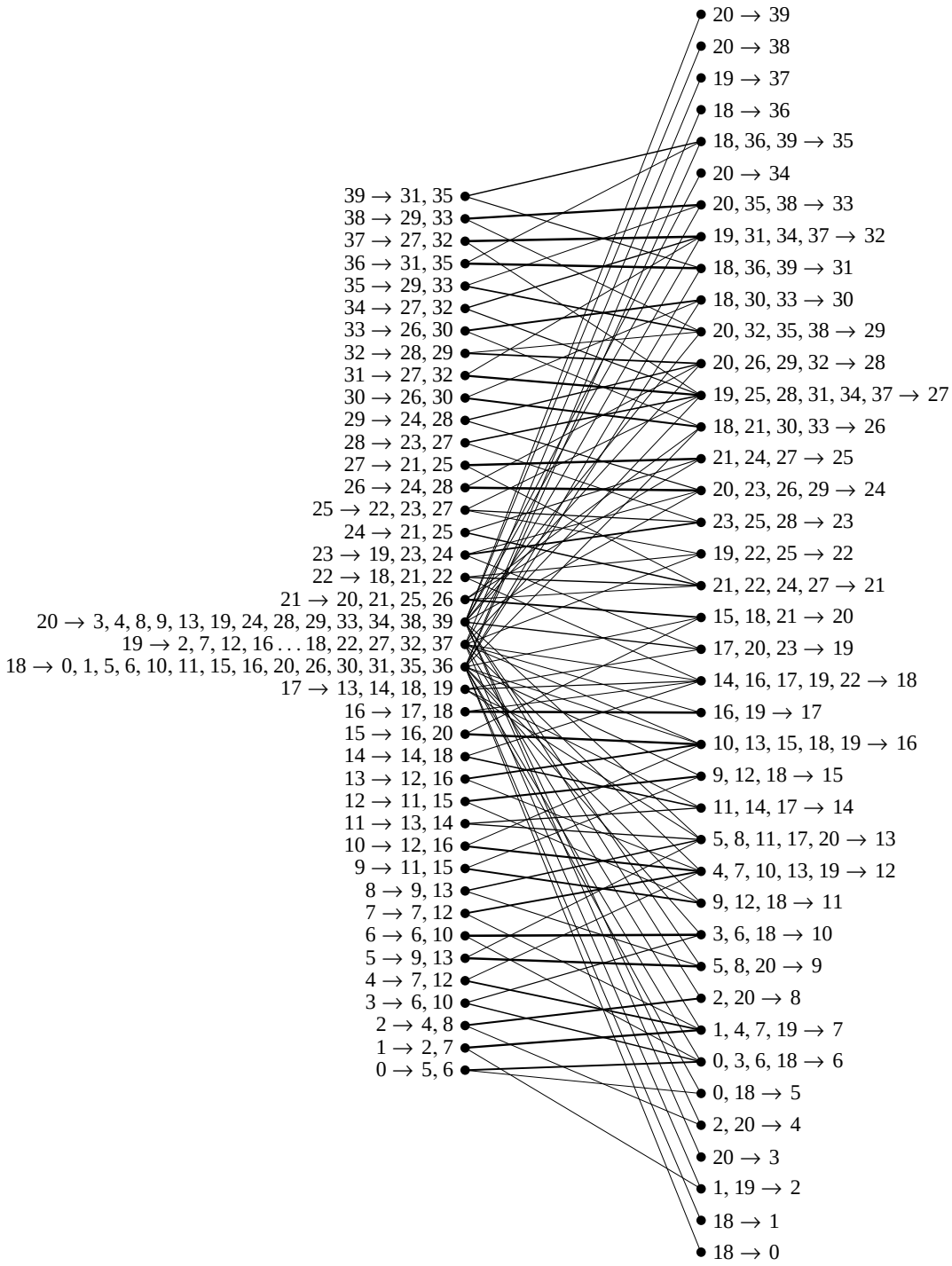

Figure 2.6: diagrammatic representation of dealt willow tree transitions implied by Table 2.8

The transition probabilities obtained for this decomposition are shown in Table A.12, and depicted in Figure 2.7. Verification of these results indicates similar accuracy to those obtained for the cut and deal decompositions individually in Tables A.7, A.8, A.10, and A.11, and is not shown.

Despite this success, it is usually impossible to compose the dealing decomposition with itself due to infeasibility, and as already noted in Section 2.5.2, the cut decomposition has the same disadvantage. This strategy is therefore limited to two phases, making it appropriate only for state spaces of moderate size. A properly hierarchical decomposition strategy will need to be more sophisticated.

2.5.5 A composable decomposition – the skim

In order to break the impasse associated with state spaces that resist nested decomposition by other means, something more like a brute force approach is required, which ensures that the separated subspaces are feasibly solvable for transition probabilities unless no such decomposition exists.

Although trying every possible decomposition is prohibitive, a workable heuristic is to search for the shortest tails that can be feasibly solved as willow trees, and "skim" them from the rest of the state space. Even without any simple decision procedure specifically for feasibility, it is not computationally expensive to verify feasibility in the tails by explicitly solving for the transition probabilities, owing to the small number of states involved. If this solution is obtained during the decomposition phase, the effort need not be wasted because the transition probabilities are needed for building the lattice.

Consequently, the subspaces associated with the tails are feasible by construction. Moreover, the remaining subspace, aside from being necessarily smaller than the original, does not have a significantly different distribution, because it loses only a fraction of probability mass from selected points, which is shared with tail points often having orders of magnitude lower probability. The majority of remaining states will therefore be almost as likely a candidate for further decomposition as the original state space.

In more precise terms, the decomposition algorithm starts with two points in each tail for both the earlier and the later state space, and attempts to find a feasible solution for the weights w_j and v_j as described in Section 2.3.2. If no feasible solution is found, the number of tail points in the later state space is incremented.

sooner			later		
state	price	probability	state	price	probability
0	80.94752249	$6.35585138 \times 10^{-5}$	0	74.30677854	$6.36154046 \times 10^{-5}$
3	83.67668838	$2.53292280 \times 10^{-3}$	1	75.47931600	$1.97457520 \times 10^{-4}$
6	86.49786878	$6.75094255 \times 10^{-2}$	5	80.35743129	$2.45666194 \times 10^{-2}$
9	89.41416598	$9.29894093 \times 10^{-1}$	6	81.62544614	$6.82601028 \times 10^{-2}$
			10	**86.90077662**	$4.55001463 \times 10^{-1}$
			15	**93.97693351**	$4.51910741 \times 10^{-1}$
12	92.42878687	$4.53141478 \times 10^{-2}$	**10**	**86.90077662**	$1.02294930 \times 10^{-2}$
15	95.54504645	$1.66759654 \times 10^{-1}$	11	88.27204340	$2.56653499 \times 10^{-2}$
18	98.76637150	$3.17247139 \times 10^{-1}$	**15**	**93.97693351**	$1.65236473 \times 10^{-1}$
21	102.09630433	$3.12019803 \times 10^{-1}$	16	95.45985981	$1.59256825 \times 10^{-1}$
24	105.53850667	$1.58659257 \times 10^{-1}$	20	101.62928774	$3.37686078 \times 10^{-1}$
			21	**103.23296577**	$1.48810297 \times 10^{-1}$
			25	109.90475792	$1.07760426 \times 10^{-1}$
			26	**111.63902024**	$4.53550577 \times 10^{-2}$
27	109.09676372	$8.72723016 \times 10^{-1}$	**21**	**103.23296577**	$3.43685312 \times 10^{-1}$
30	112.77498830	$1.18629343 \times 10^{-1}$	**26**	**111.63902024**	$4.90057919 \times 10^{-1}$
33	116.57722513	$8.33734251 \times 10^{-3}$	30	118.85408312	$1.14466698 \times 10^{-1}$
36	120.50765532	$3.02954113 \times 10^{-4}$	31	120.72956296	$5.05825883 \times 10^{-2}$
39	124.57060095	$7.34454755 \times 10^{-6}$	35	128.53213402	$9.24599720 \times 10^{-4}$
			36	130.56033045	$2.82882772 \times 10^{-4}$
1	81.84720792	$9.33424314 \times 10^{-5}$	2	76.67035574	$3.49441018 \times 10^{-4}$
4	84.60670693	$3.86336372 \times 10^{-3}$	7	82.91346987	$8.36130812 \times 10^{-2}$
7	87.45924314	$8.26313301 \times 10^{-2}$	**12**	**89.66494833**	$6.70725376 \times 10^{-1}$
10	90.40795331	$9.13411964 \times 10^{-1}$	**16**	**95.45985981**	$2.45312102 \times 10^{-1}$
			18	**98.49628196**	$1.15084973 \times 10^{-15}$
13	93.45607999	$7.42084611 \times 10^{-2}$	**12**	**89.66494833**	$3.56718998 \times 10^{-2}$
16	96.60697501	$2.19174602 \times 10^{-1}$	**16**	**95.45985981**	$6.04639498 \times 10^{-2}$
19	99.86410325	$3.34646851 \times 10^{-1}$	17	96.96618621	$2.73323216 \times 10^{-1}$
22	103.23104637	$2.64161061 \times 10^{-1}$	**18**	**98.49628196**	$1.47395874 \times 10^{-1}$
25	106.71150683	$1.07809024 \times 10^{-1}$	21	103.23296577	$1.43974356 \times 10^{-1}$
			22	104.86194932	$2.62817982 \times 10^{-1}$
			23	**106.51663772**	$5.10297336 \times 10^{-2}$
			27	**113.40064868**	$2.53229881 \times 10^{-2}$
28	110.30931187	$8.96512811 \times 10^{-1}$	**23**	**106.51663772**	$3.74873824 \times 10^{-1}$
31	114.02841781	$9.78090649 \times 10^{-2}$	**27**	**113.40064868**	$5.86417181 \times 10^{-1}$
34	117.87291433	$5.51721828 \times 10^{-3}$	32	122.63463729	$3.85596924 \times 10^{-2}$
37	121.84702901	$1.60905753 \times 10^{-4}$	33	124.56977309	$1.63304826 \times 10^{-18}$
			37	132.62053117	$1.49302806 \times 10^{-4}$
2	82.75689284	$1.76631003 \times 10^{-4}$	3	77.88018971	$6.15512783 \times 10^{-4}$
5	85.54706210	$5.86649165 \times 10^{-3}$	4	79.10911448	$1.98140721 \times 10^{-3}$
8	88.43130262	$1.00693136 \times 10^{-1}$	8	84.22181820	$1.01942241 \times 10^{-1}$
11	91.41278602	$8.93263741 \times 10^{-1}$	9	85.55081186	$2.27152216 \times 10^{-1}$
			13	**91.07983286**	$3.49561750 \times 10^{-1}$
			18	**98.49628196**	$3.18746873 \times 10^{-1}$
14	94.49479088	$1.14701702 \times 10^{-1}$	**13**	**91.07983286**	$6.53558120 \times 10^{-2}$
17	97.68070630	$2.71887934 \times 10^{-1}$	14	92.51704382	$1.15912229 \times 10^{-1}$
20	100.97403566	$3.33180143 \times 10^{-1}$	**18**	**98.49628196**	$1.58025926 \times 10^{-1}$
23	104.37840043	$2.11086695 \times 10^{-1}$	19	100.05052214	$3.35873422 \times 10^{-1}$
26	107.89754421	$6.91435253 \times 10^{-2}$	23	106.51663772	$1.48771610 \times 10^{-1}$
			24	**108.19743658**	$1.47559380 \times 10^{-1}$
			28	115.19007505	$2.21757701 \times 10^{-2}$
			29	**117.00773801**	$6.32585048 \times 10^{-3}$
			33	**124.56977309**	$4.14856217 \times 10^{-20}$
29	111.53533680	$9.16068084 \times 10^{-1}$	**24**	**108.19743658**	$5.67126150 \times 10^{-1}$
32	115.29577847	$8.02152693 \times 10^{-2}$	**29**	**117.00773801**	$3.93178577 \times 10^{-1}$
35	119.18300438	$3.63163957 \times 10^{-3}$	**33**	**124.56977309**	$2.92317498 \times 10^{-2}$
38	123.20128907	$8.50070275 \times 10^{-5}$	34	126.53544471	$1.03603943 \times 10^{-2}$
			38	134.71324122	$7.83638833 \times 10^{-5}$
			39	136.83897357	$2.47647384 \times 10^{-5}$

Table 2.9: decomposition first by dealing and then cutting

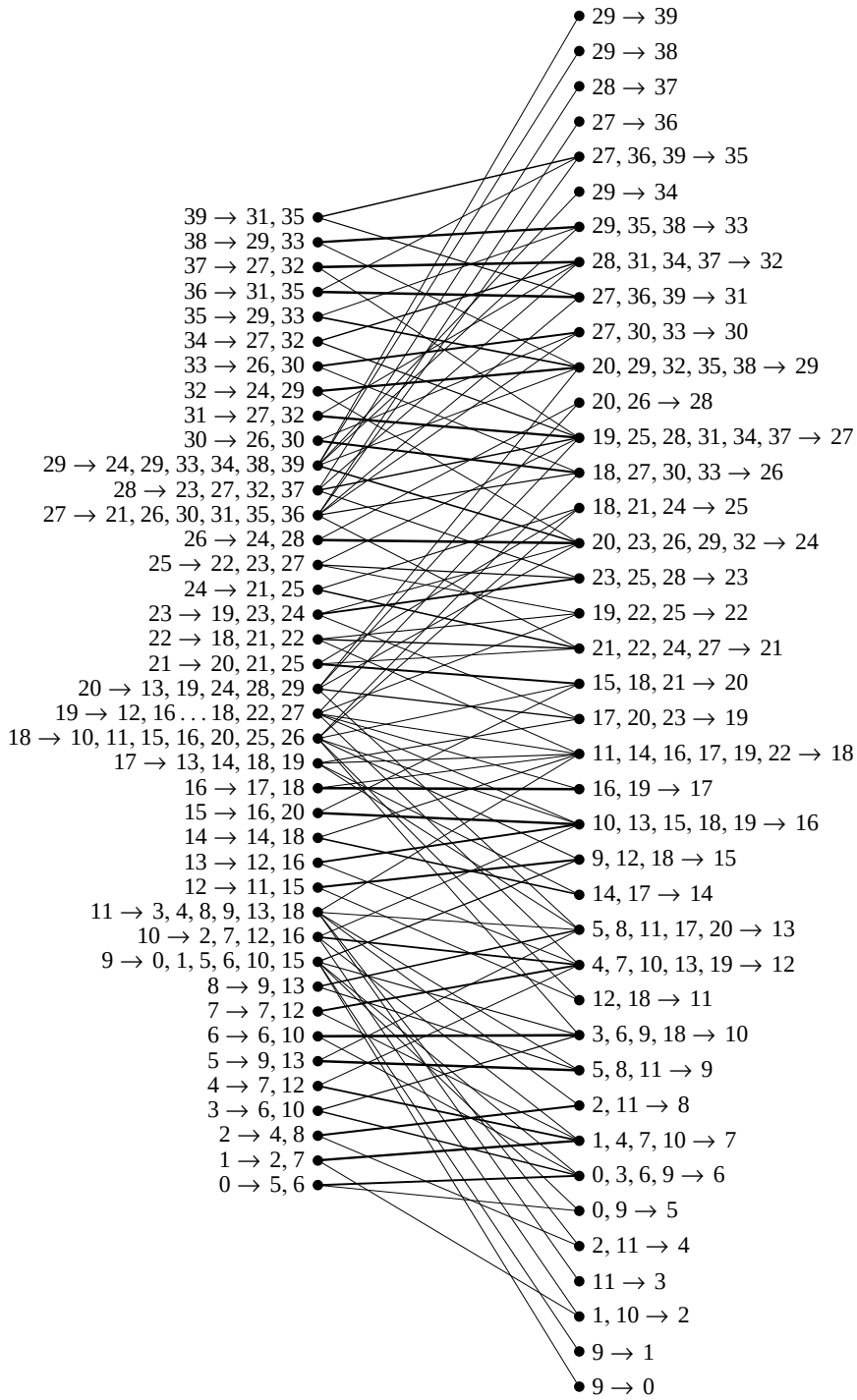

Figure 2.7: branching pattern resulting from the two stage decomposition in Table 2.9

sooner			later		
state	price	probability	state	price	probability
0	80.94752249	2.44198433×10^{-1}	0	74.30677854	2.44417014×10^{-1}
1	81.84720792	7.55801567×10^{-1}	**1**	**75.47931600**	8.28326316×10^{-2}
			6	**81.62544614**	2.91169786×10^{-14}
			9	**85.55081186**	6.72750355×10^{-1}
2	82.75689284	1.58413896×10^{-6}	**1**	**75.47931600**	3.80534943×10^{-7}
3	83.67668838	5.47967454×10^{-6}	2	76.67035574	1.59318509×10^{-6}
4	84.60670693	1.76139982×10^{-5}	3	77.88018971	5.52030935×10^{-6}
5	85.54706210	5.26144210×10^{-5}	4	79.10911448	1.77705176×10^{-5}
6	86.49786878	1.46048541×10^{-4}	5	80.35743129	5.31469333×10^{-5}
7	87.45924314	3.76735976×10^{-4}	**6**	**81.62544614**	1.47672542×10^{-4}
8	88.43130262	9.03079965×10^{-4}	7	82.91346987	3.81212013×10^{-4}
9	89.41416598	2.01171429×10^{-3}	8	84.22181820	9.14282728×10^{-4}
10	90.40795331	4.16446337×10^{-3}	**9**	**85.55081186**	2.03686646×10^{-3}
11	91.41278602	8.01135625×10^{-3}	10	86.90077662	4.21751535×10^{-3}
12	92.42878687	1.43221700×10^{-2}	11	88.27204340	8.11189270×10^{-3}
13	93.45607999	2.37940757×10^{-2}	12	89.66494833	1.44957750×10^{-2}
14	94.49479088	3.67356275×10^{-2}	13	91.07983286	2.40666622×10^{-2}
15	95.54504645	5.27067202×10^{-2}	14	92.51704382	3.71233241×10^{-2}
16	96.60697501	7.02757743×10^{-2}	15	93.97693351	5.32029523×10^{-2}
17	97.68070630	8.70778173×10^{-2}	16	95.45985981	7.08408416×10^{-2}
18	98.76637150	1.00270394×10^{-1}	17	96.96618621	8.76378945×10^{-2}
19	99.86410325	1.07300601×10^{-1}	18	98.49628196	1.00730608×10^{-1}
20	100.97403566	1.06707934×10^{-1}	19	100.05052214	1.07570513×10^{-1}
21	102.09630433	9.86182213×10^{-2}	20	101.62928774	1.06730406×10^{-1}
22	103.23104637	8.47001567×10^{-2}	21	103.23296577	9.83891058×10^{-2}
23	104.37840043	6.76049446×10^{-2}	22	104.86194932	8.42695141×10^{-2}
24	105.53850667	5.01464765×10^{-2}	23	106.51663772	6.70593285×10^{-2}
25	106.71150683	3.45677035×10^{-2}	24	108.19743658	4.95808152×10^{-2}
26	107.89754421	2.21446652×10^{-2}	25	109.90475792	3.40591894×10^{-2}
27	109.09676372	1.31836737×10^{-2}	26	111.63902024	2.17380947×10^{-2}
28	110.30931187	7.29411628×10^{-3}	27	113.40064868	1.28906679×10^{-2}
29	111.53533680	3.75040278×10^{-3}	28	115.19007505	7.10225581×10^{-3}
30	112.77498830	1.79205832×10^{-3}	29	117.00773801	3.63566809×10^{-3}
31	114.02841781	7.95784158×10^{-4}	**30**	**118.85408312**	1.72885365×10^{-3}
32	115.29577847	3.28403068×10^{-4}	31	120.72956296	7.64119116×10^{-4}
33	116.57722513	1.25946951×10^{-4}	32	122.63463729	3.13725445×10^{-4}
34	117.87291433	4.48886297×10^{-5}	33	124.56977309	1.19675424×10^{-4}
35	119.18300438	1.48680119×10^{-5}	34	126.53544471	4.24156810×10^{-5}
36	120.50765532	4.57653586×10^{-6}	35	128.53213402	1.39673422×10^{-5}
37	121.84702901	1.30914501×10^{-6}	36	130.56033045	4.27333082×10^{-6}
			37	132.62053117	1.21474229×10^{-6}
			38	**134.71324122**	2.85484013×10^{-7}
38	123.20128907	7.58264346×10^{-1}	**30**	**118.85408312**	7.02100784×10^{-1}
39	124.57060095	2.41735654×10^{-1}	**38**	**134.71324122**	7.69972662×10^{-2}
			39	136.83897357	2.20901950×10^{-1}

Table 2.10: The skimming decomposition separates out the shortest feasible tails.

This search proceeds until the each tail in the later state space covers about a third of the total. If no solution is found by then, the number of points in the tails of the earlier state space is incremented, and the tails in the later state space are reset to two points each for the search to continue. A pseudocode specification of the algorithm might be as follows.

$$
\begin{aligned}
l_k &\leftarrow \{0, 1\} \\
l_{k+1} &\leftarrow \{0, 1\} \\
h_k &\leftarrow \{N_k - 1, N_k\} \\
h_{k+1} &\leftarrow \{N_{k+1} - 1, N_{k+1}\} \\
m_k &\leftarrow \{2 \ldots N_k - 2\} \\
m_{k+1} &\leftarrow \{1 \ldots N_{k+1} - 1\} \\
&\text{while } (\neg C_1 \vee \neg C_2) \wedge (C_3 \vee C_4) \\
&\quad \text{if } C_4 \text{ then} \\
&\qquad l_{k+1} \leftarrow l_{k+1} \cup \{|l_{k+1}|\} \qquad \textit{// grow the later tails} \\
&\qquad h_{k+1} \leftarrow h_{k+1} \cup \{N_{k+1} - |h_{k+1}|\} \qquad \textit{// toward the center} \\
&\quad \text{else} \\
&\qquad l_{k+1} \leftarrow \{0, 1\} \qquad \textit{// reset the later tails} \\
&\qquad h_{k+1} \leftarrow \{N_{k+1} - 1, N_{k+1}\} \\
&\qquad l_k \leftarrow l_k \cup \{|l_{k+1}|\} \qquad \textit{// grow the earlier tails} \\
&\qquad h_k \leftarrow h_k \cup \{N_k - |h_k|\} \qquad \textit{// toward the center} \\
&\qquad m_k \leftarrow \{0 \ldots N_k\} - h_k - l_k
\end{aligned}
$$

The termination conditions depend on the existence of feasible solutions for the weights and the tail transition probabilities given the current choice of subsets, and on whether all reasonable choices have been tried.

$$
\begin{aligned}
C_1 &\Leftrightarrow \exists v_j, w_j \text{ satisfying Equations 2.15 through 2.18} \\
C_2 &\Leftrightarrow \exists p_{ij}^k \text{ satisfying Equations 2.21 through 2.29} \\
C_3 &\Leftrightarrow |l_k| < N_k/3 \\
C_4 &\Leftrightarrow |l_{k+1}| < N_{k+1}/3
\end{aligned}
$$

A more sophisticated searching algorithm using bisection is worthwhile for large state spaces, but the advantage is lost unless a heuristic is employed for bisecting the tail intervals toward the thin end rather than the middle, due to the far greater cost at the thick end.

Table 2.10 shows the application of this strategy to the same data set used in

previous examples. In this case, no more that two states are needed in the tails of the earlier state space, and it is necessary to search only eleven points deep in the later one, making a linear programming problem in 22 variables the largest that needs to be solved for the decomposition.[1] One may also note that the probability distributions in the central state spaces remain well behaved, showing no unusual peaks or troughs even on the shared states displayed in bold.

The crucial test of whether this strategy lends itself to hierarchical decomposition is to attempt to apply it again to the remaining states. A favorable result is found by inspection of Figure 2.8. The figure shows diagrammatic representations of branching patterns obtained by applying this strategy up to three nested iterations, after which the remaining problem is solved as a willow tree. The willow tree branching pattern similar to that of Figure 2.4 can be seen in the center of each diagram, but at the upper and lower tails a different pattern emerges. With each iteration, more tail points become disconnected from the central state. Several more iterations are in fact possible before the skimming decomposition itself becomes infeasible.

2.6 The crossover lattice decomposition strategy

As the Section 2.5.5 implies, a simple solution to hierarchical decomposition is easily obtained by carrying out the skimming decomposition as far as possible, and then reverting to the willow tree method. However, this choice is not ideal, because it will reduce the state spaces by only a moderate constant factor. In some cases the earlier state space retains a third of its original size, and the later a half, which is not much of an improvement asymptotically. Preferable strategies are investigated in this section.

2.6.1 A recursive algorithm

One strategy that appears to perform well incorporates a judicious combination of the three strategies (cut, deal, and skim) developed in Section 2.5. A recursive algorithm initially skims the state spaces as far as possible, and then attempts to deal them. The algorithm descends to each subproblem resulting from the deal,

[1] Experience has shown that better results are obtained by excluding the outermost point on each end from the sharable sets $h_{k+1} \cap m_{k+1}$ and $m_{k+1} \cap l_{k+1}$, so in this example only eleven rather than twelve points are involved.

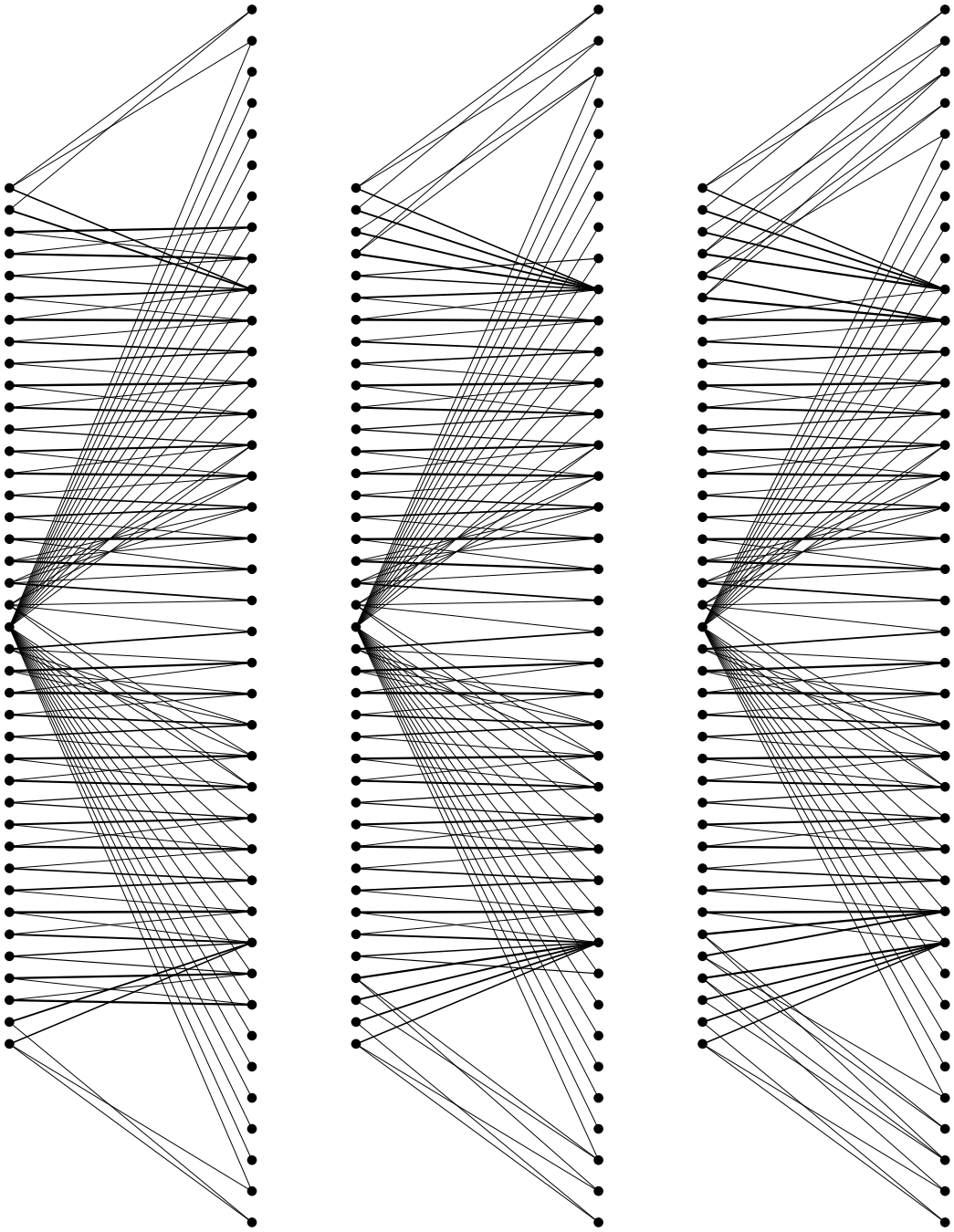

Figure 2.8: After separating the tails, the skimming decomposition may be applied repeatedly to the remaining central region, as shown here in three iterations.

because in some cases further skimming is made possible by dealing. If not all three subproblems can be solved, the algorithm attempts a cut, and then uses the willow tree method as a last resort if unsuccessful in the cut, which is how the recursion terminates.

Variations on this strategy may be more effective in particular circumstances and less in others. Globally dealing the whole state space before proceeding as above sometimes produces faster results, but wastes a constant factor of time when it is not applicable. Giving the deal priority over the skim in the recursive algorithm is seldom advantageous because the deal tends not to be composable with itself, although doing so does not diminish the algorithm's ability to find an effective decomposition. If this version is used, it is important for the algorithm to be able to backtrack from an inappropriate deal and proceed with further skimming, which can be accomplished by declining to solve for willow tree probabilities on subproblems whose size is an appreciable fraction of the original. The best strategy for a particular application is a matter of practical judgment.

2.6.2 Exercising the crossover lattice

The simpler recursive strategy described above performs as required when applied to the data in Table 2.6. The transition probabilities are listed in Table A.13, and verified in Tables A.14 and A.15. The branching pattern is depicted in Figure 2.9.

Successful results have been achieved in some larger examples, including 100, 256, 512, and 1024 point state spaces. A diagram of the branching pattern for a 100 point state space is shown in Figure 2.10. The figure is based on data derived from similar marginal distributions to those of Table 2.6, but using 100 points per time step rather than 40. The alternative strategy of giving priority to decomposition by dealing is also applicable to this data set and results in the distinctly different branching pattern shown in Figure 2.11.

2.6.3 Assessing the efficacy of decomposition

A measure of how well the crossover lattice succeeds at scalability would be to compare the original state space sizes in some larger examples with those of the willow trees embedded in the result. The latter are constructed when further decomposition is not possible, and ideally should be small relative to the originals.

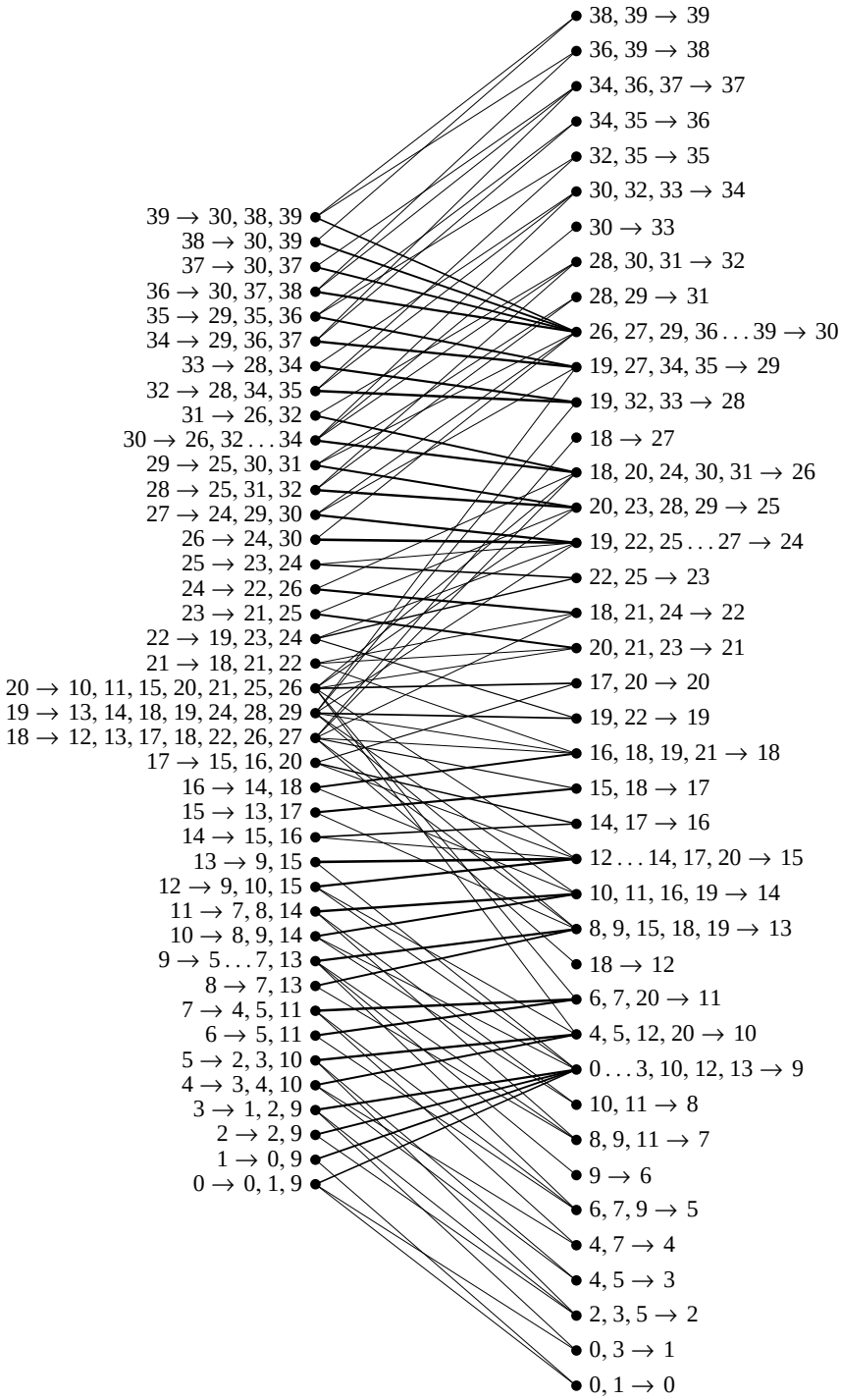

Figure 2.9: cutting deals and skimming a little leads to crossover lattice transitions

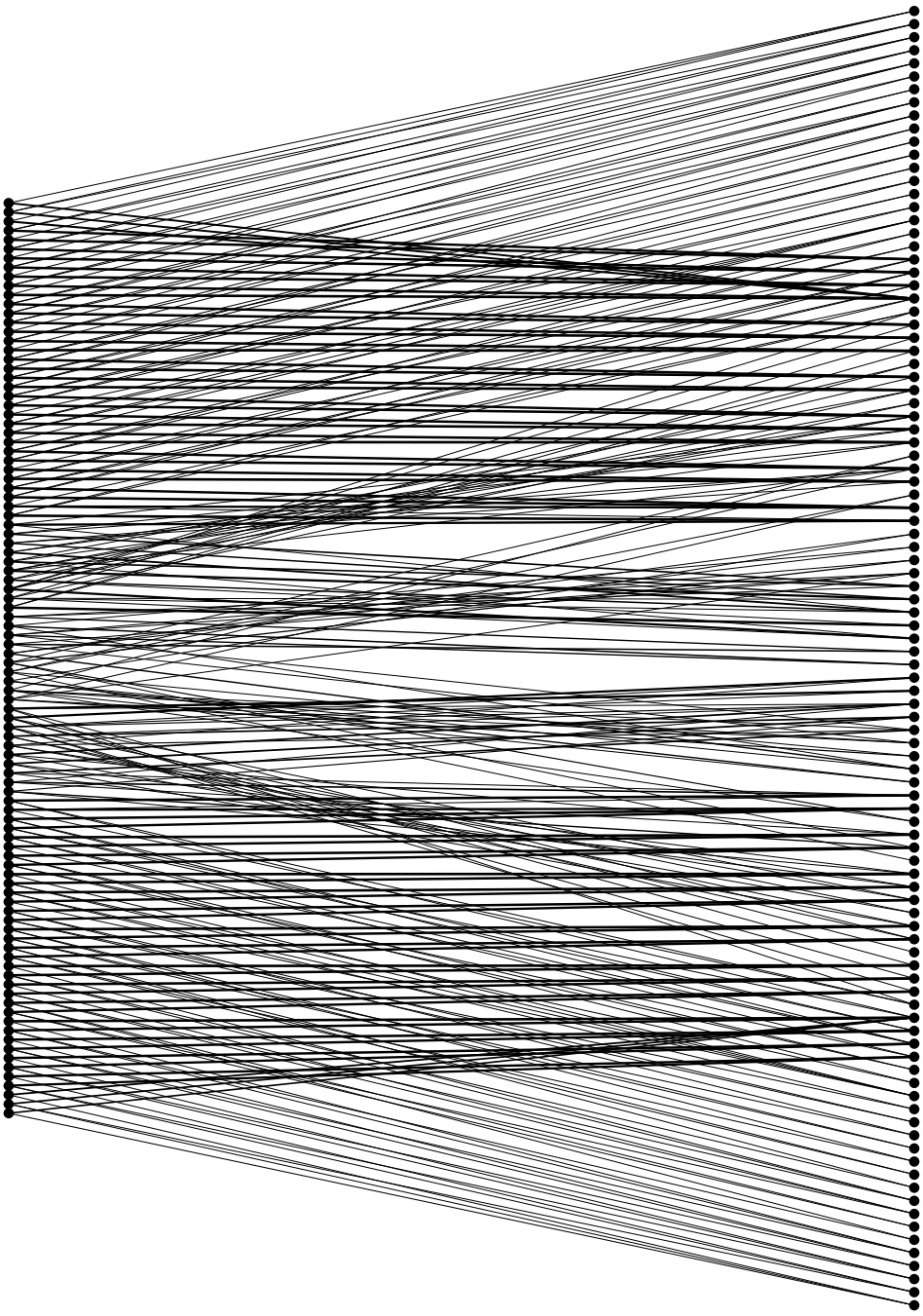

Figure 2.10: transitions in a 100 point crossover lattice, which has no feasible solution without decomposition

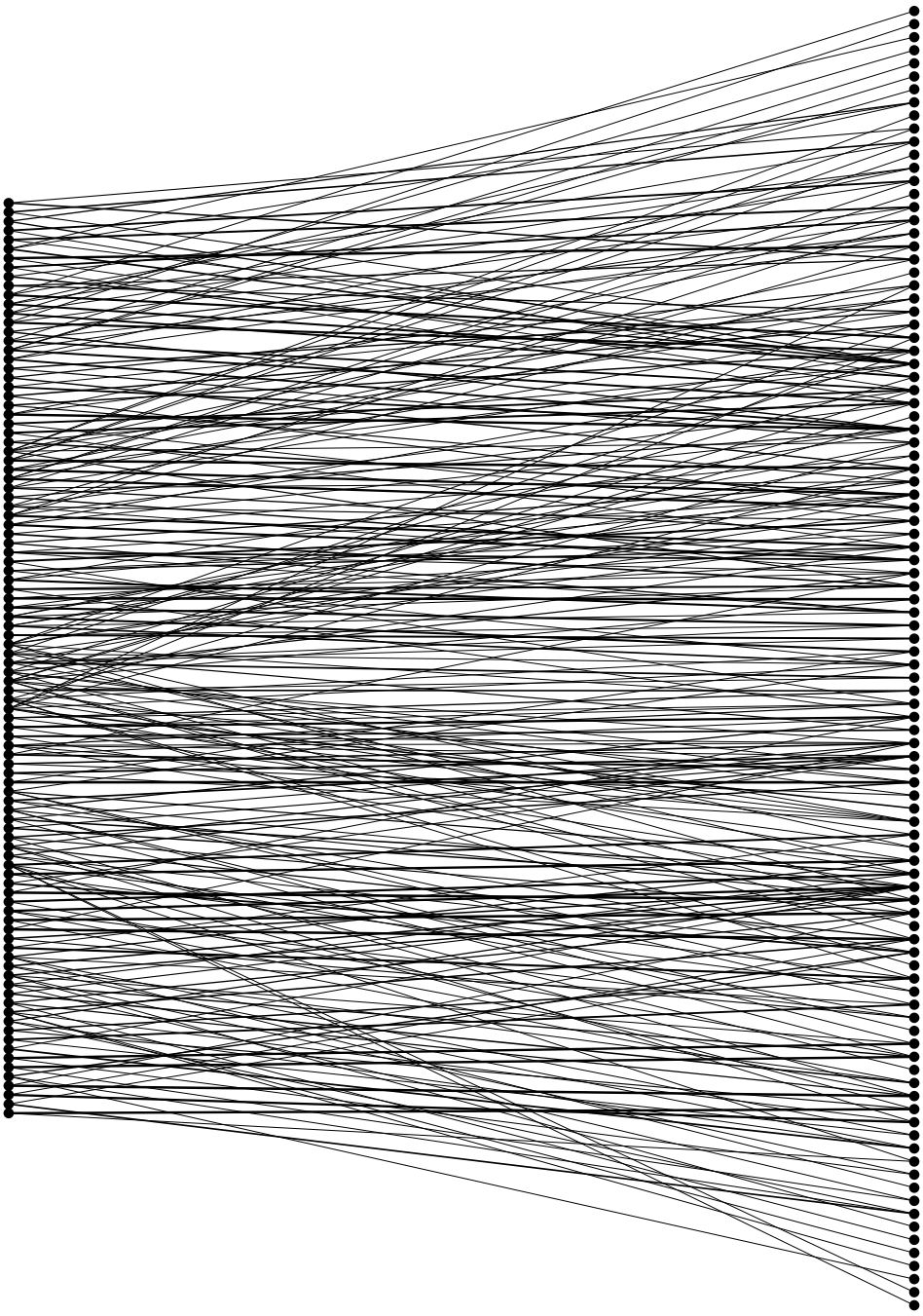

Figure 2.11: a branching pattern obtained for the same data as Figure 2.10 with an alternative strategy giving priority to decomposition by dealing

40 × 40 states			100 × 100 states			256 × 256 states			512 × 512 states		
N_k	N_{k+1}	count	N_k	N_{k+1}	count	N_k	N_{k+1}	count	N_k	N_{k+1}	count
2	3	13	2	3	19	2	3	42	2	3	87
2	4	13	2	4	25	2	4	34	2	4	59
2	5	6	2	5	9	2	5	14	2	5	37
2	6	1	2	6	10	2	6	17	2	6	18
5	8	1	2	7	2	2	7	14	2	7	8
5	9	1	2	8	3	2	8	19	2	8	4
			3	3	2	2	9	7	2	9	6
			3	5	1	2	10	7	2	10	18
			3	6	2	2	11	4	2	11	14
			3	7	1	2	12	2	2	12	20
			3	8	3	3	3	2	2	13	18
			3	10	1	3	5	3	2	14	8
			4	8	3	3	6	3	2	15	13
						3	7	2	2	16	4
						3	8	3	2	17	6
						3	9	1	2	18	2
						4	7	2	2	20	2
						5	11	1	3	3	1
						8	11	1	3	4	4
						9	14	1	3	5	5
						10	18	1	3	6	6
						21	7	1	3	7	7
						21	10	1	3	8	5
						21	11	1	3	10	1
						21	12	2	3	16	2
						21	13	2	4	3	2
						21	14	1	4	6	2
									4	7	1
									4	8	4
									4	14	2
									6	3	1
									6	5	1
									6	6	2
									6	11	1
									6	15	3
									7	16	1
									8	10	1
									8	12	1
									8	15	1
									8	18	1
									11	14	1
									23	29	1
									25	4	1
									25	5	2
									25	10	1
									25	12	2

Table 2.11: sizes and frequencies of state spaces remaining after decomposition

N_k	N_{k+1}	count	N_k	N_{k+1}	count	N_k	N_{k+1}	count	N_k	N_{k+1}	count
1	4	1	2	16	16	2	30	2	4	14	3
2	3	300	2	17	23	2	31	4	5	9	1
2	4	63	2	18	43	2	32	2	5	10	2
2	5	53	2	19	24	2	33	2	5	14	4
2	6	25	2	20	29	2	36	2	6	10	1
2	7	8	2	21	22	3	4	2	7	11	1
2	8	6	2	22	35	3	5	5	9	13	1
2	9	6	2	23	31	3	6	6	9	15	2
2	10	4	2	24	19	3	7	4	9	17	1
2	11	2	2	25	20	3	8	4	10	3	5
2	12	6	2	26	10	4	10	2	10	32	1
2	13	10	2	27	2	4	11	6	10	35	3
2	14	18	2	28	5	4	12	4	10	41	1
2	15	10	2	29	4	4	13	8			

Table 2.12: more decomposition statistics, this time for a 1024 × 1024 state system of transition probabilities

A rough idea about the effectiveness of the decomposition strategy can be gathered from the statistics in Tables 2.11 and 2.12, which are extracted from a profile log of an implementation of the first recursive algorithm described in Section 2.6.1. The data for the 40 by 40 point state space are listed in Table 2.6. The remaining state spaces are constructed from similar distributions with varying numbers of points.

In each case, the tables show how many sets of willow tree transitions the algorithm needs to evaluate in order to construct the crossover lattice, and the cardinalities of the state spaces relevant to each set. These statistics include not only those willow tree transitions that are actually incorporated into the resulting lattice, but also those whose solution is attempted and discarded due to infeasibility during the searching phase of the skimming decomposition (Section 2.5.5, page 46).

In all cases other than one of the 40 point state spaces, the size of the largest state space left standing after decomposition is less than ten percent of the corresponding original, and it appears to grow sublinearly. Because the number of variables in the linear programming system needing to be solved for willow tree probabilities is equal to the product of the state space sizes, these reductions correspond to a reduction in the number of variables to less than one percent of those that would be required for solving the original problem without decomposition. An even greater reduction in computation time will accrue insofar as the time required for

solving a linear programming problem may increase more than linearly with the number of variables.

Although a head to head comparison of computation times would be of interest, this experiment is not possible because the data sets larger than 40×40 points used in Tables 2.11 and 2.12 are *not solvable* as willow trees despite being solvable as crossover lattices. This situation may be a consequence of the range of probabilities in larger data sets causing numerical instability by spanning many orders of magnitude, whereas `glpk` is limited to IEEE standard double precision arithmetic. In what may be regarded as another strength of the method, the crossover lattice evades this hazard by decomposing the state spaces into subsets with similar probabilities and separately rescaling them.

2.7 Summary

The main points of interest in this chapter are the development of the implied willow tree model and its generalization to the crossover lattice. The former demonstrates how a lattice of transition probabilities suitable for derivatives valuation can be obtained consistently with an arbitrary given sequence of discrete marginal distributions, and the latter extends this technique to considerably larger state spaces than would be feasible otherwise.

The implied willow tree requires the solution of a linear programming problem that effectively restricts it to small state spaces, whereas the crossover lattice works essentially by reducing a given state space to a collection of more manageably sized subsets. To be effective at reducing the state spaces, the method depends on a combination of strategies and heuristics, namely the cut, deal, and skimming decompositions, with some scope for variation in the implementation. A common feature of these strategies is their separation of states into partially overlapping subsets meeting conditions that are shown to ensure consistency in the solutions. They differ in the specific choice of subsets and their intersections.

Some empirical results show the crossover lattice to be fairly robust and scalable, and various examples demonstrate the branching patterns achieved through different decomposition strategies.

Chapter 3

Verification of Convergence

But we will not boast of things without our measure . . .

— 2 Corinthians 10:13

The crossover lattice model developed in the previous chapter has state transition probabilities that are always consistent with the marginal probabilities, and are arranged so that the discounted expected asset price in any state equals its current price in that state. These conditions, while necessary for an arbitrage-free lattice model, need not imply correct results when pricing derivative securities. It is worthwhile to confirm that correct prices are forthcoming by testing the model on problems having a known solution. In this chapter, selected vanilla options in a Black-Scholes economy are priced using crossover lattices.

3.1 Discrete approximation of continuous distributions

Unlike a discrete lattice model, a Black-Scholes economy postulates prices of any possible non-negative real value, distributed according to a continuous probability density function. Lattice models are nevertheless suitable for valuation problems in this context provided that a moment matching discrete approximation to the continuous probability density function is employed.

In conventional binomial or trinomial trees, moments are matched by construc-

tion, but the discretizations of space and time are decoupled in a crossover lattice, so this issue needs explicit consideration.

A general method of moment matching for discrete approximations of continuous probability density functions is developed in this section. In contrast to conventional trees, this method is not confined to Black-Scholes valuations, and hence potentially useful for any application of crossover lattices to situations where the marginal distributions are given in continuous form.

3.1.1 Overview of discretization

A continuous probability density function f describing a distribution of prices S expresses the probability q that a price lies in an interval $[a, b]$ as

$$q = \int_{S=a}^{b} f(S)\, dS \tag{3.1}$$

An approximation is sought for f as a finite set of prices S_i and their corresponding probabilities q_i such that

$$q_i = \int_{S=a_i}^{a_{i+1}} f(S)\, dS \tag{3.2}$$

$$\sum_i q_i = 1 \tag{3.3}$$

where the interval boundaries a_i satisfy $a_i < S_i < a_{i+1}$ for all i. The goal of a good approximation is to have a mean and variance that are as close as possible to those of the continuous distribution.

$$\sum_i q_i S_i \simeq \int_{S=0}^{\infty} S f(S)\, dS = \overline{S} \tag{3.4}$$

$$\sum_i q_i S_i^2 - \left(\sum_i q_i S_i \right)^2 \simeq \int_{S=0}^{\infty} \left(S - \overline{S}\right)^2 f(S)\, dS \tag{3.5}$$

Bearing in mind that the purpose of this exercise is to construct a lattice capable of accurate valuations, one should select the prices S_i over a range of at least several standard deviations about the mean, which is easily done. The difficulty is in choosing the precise limits of integration a_i. The seemingly natural choice

point i	price S_i	probability density interval from	to	probability integral q_i
0	58.885383605957	0.000000000000	62.454283432598	1.51178481×10^{-6}
1	66.239485594261	62.454283432598	70.254099649187	2.40504865×10^{-4}
2	74.512029694046	70.254099649187	79.028022518976	1.02112471×10^{-2}
3	83.817718681188	79.028022518976	88.897706674006	1.18806699×10^{-1}
4	94.285580378977	88.897706674006	100.000000000000	3.90629305×10^{-1}
5	106.060756690528	100.000000000000	112.488841097675	3.70696190×10^{-1}
6	119.306516060600	112.488841097675	126.537393714980	1.01425583×10^{-1}
7	134.206517270581	126.537393714980	142.340447745183	7.82310546×10^{-3}
8	150.967355955227	142.340447745183	160.117120081798	1.64926632×10^{-4}
9	169.821429149837	160.117120081798	∞	9.25880862×10^{-7}

Table 3.1: a naive discretization of a lognormal distribution having mean 100 and standard deviation 10, with interval boundaries at geometric means between points

of positioning the interval boundaries midway between the adjacent price points can lead to an inaccurate discretization if the quality is judged by how well the moments match.

3.1.2 A motivating example

Table 3.1 shows a ten point discretization of a lognormal probability distribution with mean 100 and standard deviation 10. (These parameters would correspond to values of 4.600195020562 and 0.099751345120 respectively for θ and λ in Equation 3.12). Points are chosen in a geometric progression about the mean, and the probability assigned to each point is the integral of the probability density function over the interval shown. The interval boundaries between each pair of successive points are their geometric mean, and the integration is performed using a reliable standard numerical integration software library typically capable of twelve digit precision for smooth functions [PdDKUK83]. Despite these provisions, the mean of the discrete distribution turns out to be

$$\sum_{i=0}^{9} q_i S_i = 100.05772 \tag{3.6}$$

point i	price S_i	probability density interval from	to	probability integral q_i
0	58.885383605957	0.000000000000	60.542598940395	3.16506641×10^{-7}
1	66.239485594261	60.542598940395	68.632898983348	9.79277551×10^{-5}
2	74.512029694046	68.632898983348	77.826810559293	6.78661578×10^{-3}
3	83.817718681188	77.826810559293	88.251189558253	1.07589767×10^{-1}
4	94.285580378977	88.251189558253	100.045623376376	4.07236390×10^{-1}
5	106.060756690528	100.045623376376	113.365665704620	3.82765242×10^{-1}
6	119.306516060600	113.365665704620	128.386199364091	9.02117809×10^{-2}
7	134.206517270581	128.386199364091	145.304851391330	5.23838416×10^{-3}
8	150.967355955227	145.304851391330	164.345469918009	7.33308795×10^{-5}
9	169.821429149837	164.345469918009	∞	2.44900996×10^{-7}

Table 3.2: a better discretization of a lognormal distribution having mean 100 and standard deviation 10, with interval boundaries optimized to match the mean and variance

and the standard deviation is

$$\sqrt{\sum_{i=0}^{9} q_i S_i^2 - \left(\sum_{i=0}^{9} q_i S_i\right)^2} = 10.57275 \tag{3.7}$$

both of which differ significantly from corresponding moments of the continuous distribution, 100 and 10. These differences are a matter of concern insofar as accurate derivative prices can hardly be expected of a lattice based on an inaccurate approximation of the underlying marginal distributions. While it may be argued that this error diminishes as the number of price points increases, the number of price points in a crossover lattice is a freely chosen parameter that is not compelled to increase. It would be advantageous to obtain a well matched approximation to the continuous distribution for any given number of points.

One such approximation is shown in Table 3.2, which has the same price points as Table 3.1 and is based on the same distribution, but differs in the choice of interval boundaries. This simple adjustment brings the discrete mean within 1.065×10^{-9} of the correct value, and the discrete standard deviation within 5.875×10^{-11} .

3.1.3 The discretization method

The method of discretization amounts to a systematic search for the interval boundaries a_i to optimize Equations 3.4 and 3.5. Taking each boundary a_i as an independent parameter whose optimal value is sought would be prohibitive, but an acceptable solution can be found by restricting them to vary *en masse* with respect to a pair of independent parameters κ and ω. This approach reduces the problem to a bivariate non-linear optimization.

In order to satisfy Equation 3.3, a_0 and a_N must be fixed at 0 and ∞. For any choice of κ and ω, the remaining boundaries a_i are given by

$$a_i = S_{i-1} + (S_i - S_{i-1}) \left(\frac{g_i - S_{i-1}}{S_i - S_{i-1}} \right)^{(g_i/\overline{S})^{\omega}} \tag{3.8}$$

where

$$g_i = S_{i-1} + (S_i - S_{i-1}) \left(\frac{\sqrt{S_i S_{i-1}} - S_{i-1}}{S_i - S_{i-1}} \right)^{\kappa} \tag{3.9}$$

A value of $\kappa = 1$ therefore identifies g_i with the geometric mean of S_{i-1} and S_i, higher values tilt it downward, and lower values upward. A value of 0 for ω identifies a_i with g_i, higher values draw it inward toward the distribution mean $\overline{S}$, and lower values push it outward.

The intuition underlying this parameterization is that pushing the interval boundaries away from the mean will cause more probability mass to accrue to the central intervals, which will diminish the overall variance of the discrete approximation. Tilting all of the interval boundaries upward will leave more probability mass in the lower intervals, thereby reducing the mean, and conversely tilting them downward will increase it. Hence, κ can be tuned to adjust the mean and ω the variance.

In practice both parameters have an effect on the mean and variance, but the effect of κ on the variance and the effect of ω on the mean are small. This condition permits an adequate substitute for more sophisticated non-linear optimization techniques to be made by solving for each of them individually using bisection. Accuracies on the scale reported for the example in the previous section are typically achieved by refining the solution for each parameter after the other is found for five or six further iterations.

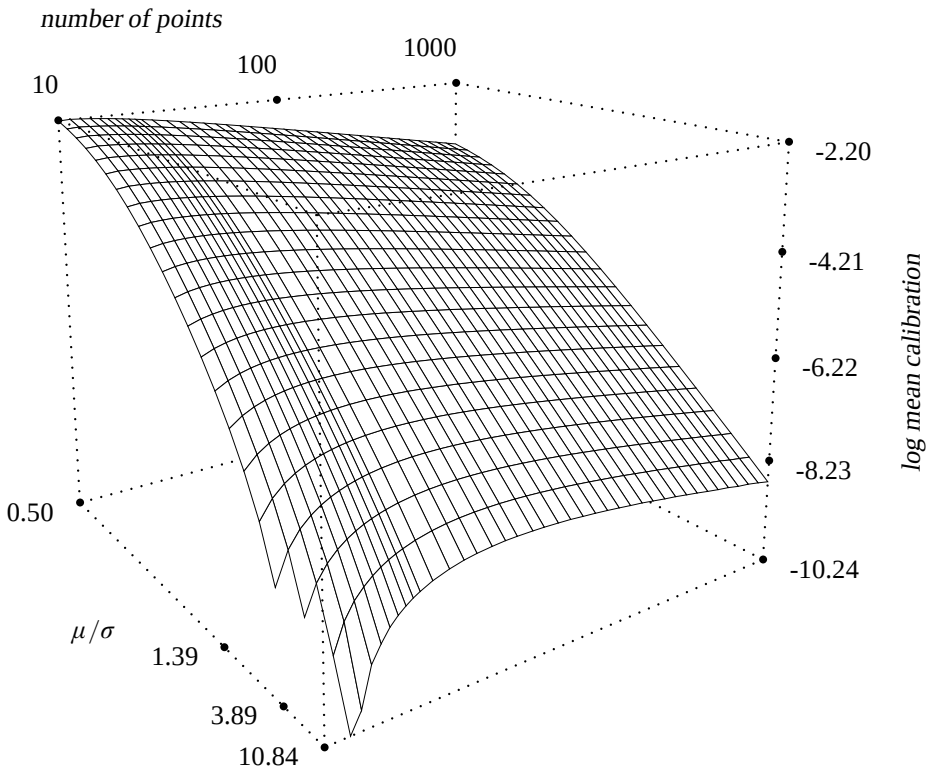

Figure 3.1: plot of $\kappa - 1$ for mean calibration parameter κ over a range of lognormal distribution specifications with varying discrete state space sizes

3.1.4 The calibration space

As one might expect, this optimization problem is time consuming, with the bulk of the time spent on numerical integrations. Because these apply to separate intervals of the probability density function with no data dependence between them, there would seem to be a potential for enhanced performance through a distributed concurrent implementation, but such issues are beyond the scope of this dissertation.

In a more conventional vein, the calibration parameters κ and ω are determined by the number of points and shape of the distribution being approximated. If similar forms of risk neutral distributions were to recur frequently in the course of a firm's derivatives valuation activities, it could be worthwhile to compute cali-

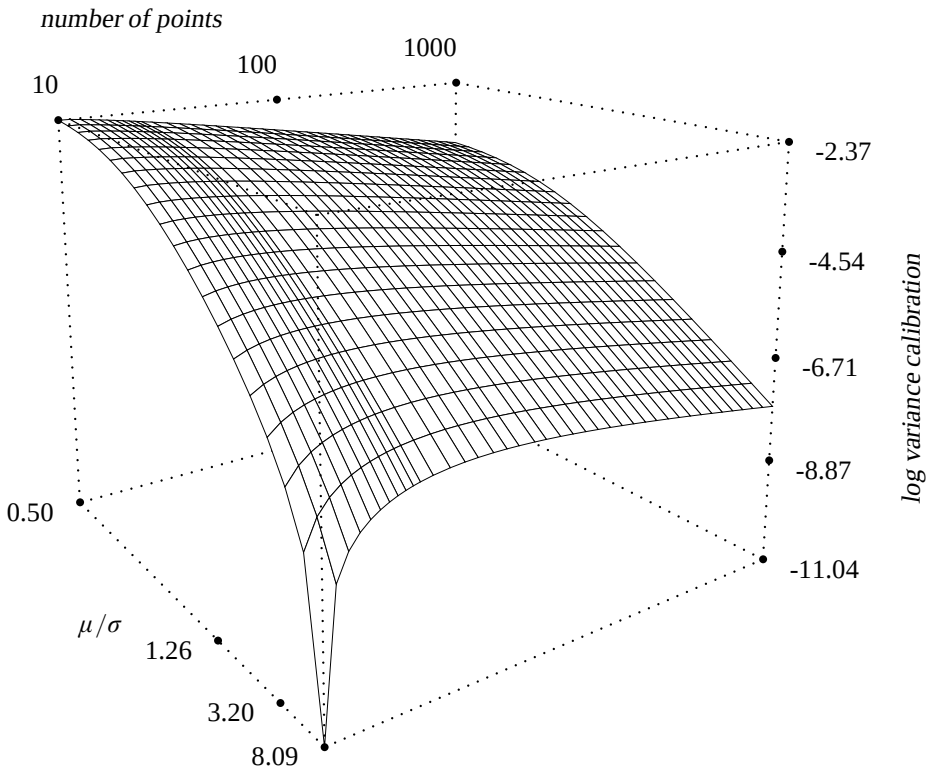

Figure 3.2: plot of variance calibration parameter ω over a range of lognormal distribution specifications with varying discrete state space sizes

bration parameters in advance for a range of distributions, and obtain individual ones as needed by interpolation. Some examples for lognormal distributions are shown in Figures 3.1 and 3.2.

3.2 Marginal distributions under Black-Scholes assumptions

In a Black-Scholes economy, the marginal distributions of asset prices S^k at time step k postulated in Section 2.1 take a known form. For a constant drift parameter μ, the expected price $E\left[S^k\right]$ increases exponentially with time

$$E\left[S^k\right] = S^0 e^{\mu k \Delta t} \tag{3.10}$$

where Δt is the amount of elapsed time per step. A constant volatility parameter σ determines the variance $V\left[S^k\right]$ at time step k.

$$V\left[S^k\right] = \left(S^0\right)^2 e^{2\mu k \Delta t}\left(e^{\sigma^2 k \Delta t} - 1\right) \tag{3.11}$$

The marginal distribution at time step k is given by a lognormal probability density function f of the form

$$f(S) = \frac{1}{\sqrt{2\pi}\lambda S} \exp\left(-\frac{1}{2}\left(\frac{\ln S - \theta}{\lambda}\right)^2\right) \tag{3.12}$$

for $S \geq 0$, where

$$\theta = \ln\left(E\left[S^k\right]^2\right) - \frac{1}{2}\ln\left(E\left[S^k\right]^2 + V\left[S^k\right]\right) \tag{3.13}$$

$$\lambda^2 = \ln\left(1 + V\left[S^k\right] / E\left[S^k\right]^2\right) \tag{3.14}$$

When the following parameters are fixed

$$\text{the initial asset price} \quad S^0 = 100 \tag{3.15}$$

$$\text{the annual volatility} \quad \sigma = 0.2 \tag{3.16}$$

$$\text{the annual drift} \quad \mu = 0.1 \tag{3.17}$$

$$\text{the expiration in years} \quad T = 1 \tag{3.18}$$

$$\text{the number of years per time step} \quad \Delta t = 1/24 \tag{3.19}$$

it is possible to compute the means and variances of the marginal distributions in a Black-Scholes economy based on Equations 3.10 and 3.11 as shown in Table 3.3.

k	$E\left[S^k\right]$	$V\left[S^k\right]$	θ	λ
0	100.000000000000	0.000000000000	4.605170185988	0.000000000000
1	100.417535929112	16.820148767210	4.608503519321	0.040824829046
2	100.836815220745	33.950096404948	4.611836852655	0.057735026919
3	101.257845154063	51.394134290879	4.615170185988	0.070710678119
4	101.680633038626	69.156606788142	4.618503519321	0.081649658093
5	102.105186214511	87.241911860327	4.621836852655	0.091287092918
6	102.531512052443	105.654501693355	4.625170185988	0.100000000000
7	102.959617953923	124.398883324282	4.628503519321	0.108012344973
8	103.389511351357	143.479619277122	4.631836852655	0.115470053838
9	103.821199708183	162.901328205788	4.635170185988	0.122474487139
10	104.254690518999	182.668685544186	4.638503519321	0.129099444874
11	104.689991309700	202.786424163583	4.641836852655	0.135400640077
12	105.127109637602	223.259335037280	4.645170185988	0.141421356237
13	105.566053091576	244.092267912733	4.648503519321	0.147196014439
14	106.006829292179	265.290131991129	4.651836852655	0.152752523165
15	106.449445891786	286.857896614568	4.655170185988	0.158113883008
16	106.893910574725	308.800591960876	4.658503519321	0.163299316186
17	107.340231057407	331.123309746175	4.661836852655	0.168325082306
18	107.788415088463	353.831203935271	4.665170185988	0.173205080757
19	108.238470448878	376.929491459941	4.668503519321	0.177951304201
20	108.690404952123	400.423452945238	4.671836852655	0.182574185835
21	109.144226444295	424.318433443852	4.675170185988	0.187082869339
22	109.599942804251	448.619843178652	4.678503519321	0.191485421551
23	110.057561943745	473.333158293491	4.681836852655	0.195789002075
24	110.517091807565	498.463921612348	4.685170185988	0.200000000000

Table 3.3: marginal distribution statistics for 24 time steps corresponding to Black-Scholes parameters $S^0 = 100$, $\sigma = 0.2$, $\mu = 0.1$ and $T = 1$

The time to expiration T differs from the number of time steps $\mathcal{T}$, which is used in Chapter 2 but is expressed as $T/\Delta t$ hereafter to avoid confusion.

These statistics determine a family of probability density functions f according to Equations 3.13, 3.14, and 3.12. These probability density functions in turn can be discretized using the method described in Section 3.1, and the resulting discrete sets of state prices will determine the transition probabilities in a crossover lattice as described in Section 2.6.

3.2.1 Black-Scholes prices

To the extent that the crossover lattice successfully models the Black-Scholes economy, the prices it estimates for European call and put options should converge to the analytical solutions

$$c = S^0 N(d_1) - X e^{-rT} N(d_2) \tag{3.20}$$

$$p = X e^{-rT} N(-d_2) - S^0 N(-d_1) \tag{3.21}$$

where X is the strike price of the option, N is the cumulative standard normal density function, r is the risk free interest rate assumed to equal the drift μ, and

$$d_1 = \frac{\ln(S^0/X) + (\sigma^2/2)T}{\sigma\sqrt{T}} \tag{3.22}$$

$$d_2 = d_1 - \sigma\sqrt{T} \tag{3.23}$$

3.2.2 Crossover lattice prices

The price of a European option with any given payoff function f may be estimated from a crossover lattice by a standard backward induction argument, which asserts the price of any derivative to be equal to its discounted expected payoff under the risk neutral measure. In the case of a call with strike X, the payoff function is

$$f_c(S) = \max(S - X, 0) \tag{3.24}$$

and in the case of a put, the payoff function is

$$f_p(S) = \max(X - S, 0) \tag{3.25}$$

The price of the derivative at time k in state i is

$$v_i^k = \begin{cases} f\left(S_i^k\right) & \text{if } k\Delta t = T \\ d_k \sum_{j=0}^{N_{k+1}} p_{ij}^k v_j^{k+1} & \text{otherwise} \end{cases} \tag{3.26}$$

from which the current price is given by v_0^0. The discount factor d_k is defined in Equation 2.6. It may be noted in passing that the lattice is readily amenable to pricing American options as well.

$$v_i^k = \begin{cases} f\left(S_i^k\right) & \text{if } k\Delta t = T \\ \max\left(f\left(S_i^k\right), d_k \sum_{j=0}^{N_{k+1}} p_{ij}^k v_j^{k+1}\right) & \text{otherwise} \end{cases} \tag{3.27}$$

3.3 Results

Some experimental results demonstrate the effect of varying the number of time steps and the number of states in a crossover lattice used to value European calls and puts at a selection of strike prices.

3.3.1 Varying the number of time steps

Tables 3.4 and 3.5 are based on crossover lattices with a constant state space size of 256 states per time step, except at time 0, which necessarily has only a single state. Lattices with fewer than 24 time steps are obtained by using a subset of the marginal distributions listed in Table 3.3 to compute the transition probabilities.

Black-Scholes prices in the tables are computed from Equations 3.20 and 3.21, and crossover lattice prices are computed from Equations 3.24, 3.25, and 3.26.

A clear pattern of improving accuracy is noticeable as the number of time steps increases. At 24 times steps, all options are priced correctly to the penny, but better results are obtained deep in the money for both calls and puts, with up to eight significant digits.

3.3.2 Varying the number of states

Comparable results are obtained by holding the number of time steps fixed at 24 and varying the number of states from 23 to 40, as shown in Tables 3.6 and 3.7. Overall accuracies are lower in this experiment than in the previous one because of the smaller number of states used, although even the worst results are off by no more than a few percentage points. A trend toward improved accuracy with increasing state space sizes is evident but not without some exceptions.

3.3.3 General observations and conclusions

Based on these results, there is reasonable cause for confidence that correct derivative prices can be estimated from crossover lattices when used in conjunction with an accurate moment matching discretization such as that of Section 3.1. It

	time steps				
strike	6	8	12	24	Black-Scholes
50.00	54.1890978	54.3795973	54.5693104	54.7582402	54.7582404
52.50	51.8986449	52.0987018	52.2979386	52.4963598	52.4963604
55.00	49.6083603	49.8180193	50.0268306	50.2348001	50.2347995
57.50	47.3184734	47.5378195	47.7563010	47.9739213	47.9739201
60.00	45.0294473	45.2586323	45.4869431	45.7143839	45.7143803
62.50	42.7420995	42.9813709	43.2197554	43.4572812	43.4572870
65.00	40.4578739	40.7075945	40.9564168	41.2043605	41.2043760
67.50	38.1789490	38.4396133	38.6993328	38.9582039	38.9582023
70.00	35.9085909	36.1807880	36.4519774	36.7223605	36.7223151
72.50	33.6513088	33.9356159	34.2191327	34.5013449	34.5013894
75.00	31.4129020	31.7103049	32.0063849	32.3013681	32.3012872
77.50	29.2008443	29.5117534	29.8212320	30.1287772	30.1290284
80.00	27.0238663	27.3486811	27.6712496	27.9927167	27.9926628
82.50	24.8916111	25.2299174	25.5667438	25.9012616	25.9010456
85.00	22.8141543	23.1668532	23.5167636	23.8638227	23.8635336
87.50	20.8039414	21.1692332	21.5311991	21.8899746	21.8896255
90.00	18.8708102	19.2471609	19.6198096	19.9889155	19.9885771
92.50	17.0248440	17.4102341	17.7915452	18.1689606	18.1690248
95.00	15.2750769	15.6667871	16.0540562	16.4379516	16.4386438
97.50	13.6288237	14.0245724	14.4167112	14.8043830	14.8038666
100.00	12.0933451	12.4903645	12.8823959	13.2696834	13.2696766
102.50	10.6722856	11.0651449	11.4527810	11.8396452	11.8394838
105.00	9.3652737	9.7511905	10.1357342	10.5151860	10.5150844
107.50	8.1738023	8.5515231	8.9237816	9.2972441	9.2966948
110.00	7.0968949	7.4596697	7.8232691	8.1816541	8.1830521
112.50	6.1282925	6.4773658	6.8245428	7.1720750	7.1715645
115.00	5.2659086	5.5967916	5.9267170	6.2592024	6.2584970
117.50	4.5024545	4.8115710	5.1260143	5.4391016	5.4391783
120.00	3.8293733	4.1205759	4.4138221	4.7069846	4.7082143
122.50	3.2435912	3.5127761	3.7844120	4.0589665	4.0596971
125.00	2.7349504	2.9814412	3.2315915	3.4871805	3.4874015
127.50	2.2955293	2.5198678	2.7488963	2.9849318	2.9849602
130.00	1.9183496	2.1212925	2.3298795	2.5459801	2.5460172
132.50	1.5965737	1.7790543	1.9678603	2.1641460	2.1643538
135.00	1.3236416	1.4867321	1.6564846	1.8334388	1.8339899
137.50	1.0933778	1.2382542	1.3900871	1.5484414	1.5492606
140.00	0.9000695	1.0279785	1.1629739	1.3047476	1.3048694
142.50	0.7385153	0.8507447	0.9700420	1.0962178	1.0959224
145.00	0.6040485	0.7019013	0.8066733	0.9182829	0.9179450
147.50	0.4925378	0.5773095	0.6687418	0.7668475	0.7668850
150.00	0.4003703	0.4733301	0.5526026	0.6387711	0.6391043

Table 3.4: European call prices from crossover lattices with 256 states and varying numbers of time steps compared to the Black-Scholes price, for $S^0 = 100$, $\sigma = 0.2$, $\mu = 0.1$ and $T = 1$

	time steps				
strike	6	8	12	24	Black-Scholes
50.00	0.0000414	0.0000591	0.0000824	0.0001111	0.0001113
52.50	0.0001357	0.0001866	0.0002492	0.0003242	0.0003248
55.00	0.0003983	0.0005272	0.0006797	0.0008581	0.0008575
57.50	0.0010585	0.0013505	0.0016888	0.0020729	0.0020716
60.00	0.0025796	0.0031864	0.0038695	0.0046290	0.0046254
62.50	0.0057790	0.0069481	0.0082204	0.0096199	0.0096256
65.00	0.0121006	0.0141948	0.0164204	0.0187927	0.0188082
67.50	0.0237228	0.0272367	0.0308749	0.0347296	0.0347280
70.00	0.0439119	0.0494345	0.0550581	0.0609798	0.0609344
72.50	0.0771770	0.0852855	0.0937520	0.1020577	0.1021022
75.00	0.1293174	0.1409976	0.1525428	0.1641744	0.1640935
77.50	0.2078069	0.2234692	0.2389286	0.2536771	0.2539283
80.00	0.3213760	0.3414199	0.3604847	0.3797101	0.3796562
82.50	0.4796680	0.5036793	0.5275176	0.5503486	0.5501326
85.00	0.6927584	0.7216382	0.7490759	0.7750033	0.7747142
87.50	0.9730926	1.0050413	1.0350501	1.0632487	1.0628995
90.00	1.3305086	1.3639921	1.3951992	1.4242831	1.4239447
92.50	1.7750896	1.8080884	1.8384733	1.8664217	1.8664860
95.00	2.3158697	2.3456644	2.3725230	2.3975063	2.3981985
97.50	2.9601637	2.9844728	3.0067165	3.0260313	3.0255149
100.00	3.7152322	3.7312881	3.7439399	3.7534252	3.7534184
102.50	4.5847200	4.5870915	4.5858635	4.5854805	4.5853192
105.00	5.5682552	5.5541602	5.5403553	5.5231149	5.5230132
107.50	6.6673310	6.6355159	6.5999413	6.5672665	6.5667172
110.00	7.8809708	7.8246856	7.7709674	7.7137700	7.7151681
112.50	9.2029155	9.1234048	9.0437798	8.9662846	8.9657741
115.00	10.6310788	10.5238536	10.4174925	10.3155055	10.3148001
117.50	12.1581719	12.0196561	11.8883284	11.7574982	11.7575750
120.00	13.7756378	13.6096841	13.4476748	13.2874747	13.2887044
122.50	15.4804030	15.2829075	15.0898033	14.9015502	14.9022808
125.00	17.2623094	17.0325957	16.8085215	16.5918578	16.5920787
127.50	19.1134354	18.8520453	18.5973648	18.3517026	18.3517310
130.00	21.0268029	20.7344931	20.4498866	20.1748444	20.1748815
132.50	22.9955742	22.6732780	22.3594060	22.0551038	22.0553117
135.00	25.0131893	24.6619789	24.3195689	23.9864902	23.9870414
137.50	27.0734727	26.6945241	26.3247100	25.9635864	25.9644056
140.00	29.1707116	28.7652714	28.3691354	27.9819861	27.9821079
142.50	31.2997045	30.8690607	30.4477421	30.0355499	30.0352545
145.00	33.4557849	33.0012404	32.5559120	32.1197085	32.1193706
147.50	35.6348213	35.1576717	34.6895191	34.2303667	34.2304042
150.00	37.8332011	37.3347154	36.8449185	36.3643838	36.3647170

Table 3.5: European put prices from crossover lattices with 256 states and varying numbers of time steps compared to the Black-Scholes price, for $S^0 = 100$, $\sigma = 0.2$, $\mu = 0.1$ and $T = 1$

	strike								
states	80.00	85.00	90.00	95.00	100.00	105.00	110.00	115.00	120.00
23	27.9852	23.8790	19.9865	16.4808	13.2049	10.6126	8.0203	6.3349	4.7542
24	28.0134	23.8234	20.0282	16.3812	13.3431	10.3586	8.2791	6.1997	4.7314
25	27.9739	23.8963	19.9017	16.4975	13.1492	10.5998	8.0504	6.3323	4.7102
26	28.0016	23.8416	20.0318	16.3395	13.3389	10.3741	8.2637	6.1838	4.7551
27	28.0039	23.8761	19.9636	16.4895	13.1748	10.5866	8.0734	6.3276	4.6704
28	27.9637	23.8786	20.0068	16.4083	13.3254	10.4237	8.2516	6.1715	4.7648
29	28.0060	23.8268	20.0043	16.4652	13.2297	10.5735	8.0913	6.3218	4.6344
30	27.9948	23.8858	19.9619	16.4488	13.3062	10.4595	8.2421	6.1616	4.7652
31	27.9817	23.8539	20.0179	16.4305	13.2652	10.5609	8.1056	6.3154	4.6019
32	28.0048	23.8680	19.9510	16.4679	13.2838	10.4852	8.2343	6.1537	4.7595
33	27.9866	23.8769	20.0104	16.3892	13.2866	10.5489	8.1170	6.3088	4.6448
34	27.9940	23.8301	19.9884	16.4713	13.2596	10.5037	8.2279	6.1802	4.7497
35	28.0011	23.8790	19.9869	16.4007	13.2977	10.5375	8.1264	6.3021	4.6785
36	27.9792	23.8602	20.0063	16.4630	13.2348	10.5167	8.2227	6.2032	4.7373
37	27.9991	23.8634	19.9514	16.4330	13.3013	10.5269	8.1341	6.2957	4.7019
38	27.9964	23.8750	20.0080	16.4461	13.2098	10.5256	8.2182	6.2210	4.7233
39	27.9814	23.8407	19.9771	16.4516	13.2992	10.5168	8.1405	6.2894	4.7174
40	28.0002	23.8745	19.9969	16.4231	13.2352	10.5315	8.2145	6.2348	4.7083
	27.9927	23.8635	19.9886	16.4386	13.2697	10.5151	8.1831	6.2585	4.7082

Table 3.6: European call prices from crossover lattices with 24 time steps and with varying state space sizes compared to the Black-Scholes price, for $S^0 = 100$, $\sigma = 0.2$, $\mu = 0.1$ and $T = 1$

	strike								
states	80.00	85.00	90.00	95.00	100.00	105.00	110.00	115.00	120.00
23	0.3722	0.7902	1.4218	2.4403	3.6886	5.6205	7.5524	10.3912	13.3346
24	0.4003	0.7345	1.4636	2.3408	3.8268	5.3665	7.8112	10.2560	13.3119
25	0.3609	0.8075	1.3370	2.4571	3.6330	5.6077	7.5825	10.3886	13.2907
26	0.3886	0.7528	1.4672	2.2991	3.8227	5.3820	7.7958	10.2401	13.3356
27	0.3909	0.7873	1.3989	2.4490	3.6586	5.5945	7.6055	10.3839	13.2509
28	0.3507	0.7898	1.4422	2.3679	3.8091	5.4316	7.7838	10.2278	13.3453
29	0.3930	0.7380	1.4396	2.4248	3.7134	5.5814	7.6234	10.3781	13.2149
30	0.3818	0.7970	1.3972	2.4083	3.7900	5.4674	7.7742	10.2180	13.3457
31	0.3686	0.7651	1.4533	2.3900	3.7489	5.5688	7.6377	10.3717	13.1824
32	0.3918	0.7792	1.3864	2.4275	3.7675	5.4932	7.7664	10.2100	13.3400
33	0.3736	0.7881	1.4458	2.3488	3.7704	5.5568	7.6492	10.3651	13.2252
34	0.3810	0.7413	1.4238	2.4308	3.7434	5.5116	7.7601	10.2365	13.3302
35	0.3881	0.7902	1.4223	2.3602	3.7815	5.5455	7.6585	10.3584	13.2590
36	0.3662	0.7714	1.4416	2.4225	3.7185	5.5246	7.7548	10.2595	13.3178
37	0.3861	0.7746	1.3867	2.3926	3.7850	5.5348	7.6662	10.3520	13.2824
38	0.3834	0.7862	1.4434	2.4057	3.6935	5.5336	7.7503	10.2774	13.3038
39	0.3684	0.7519	1.4125	2.4111	3.7830	5.5248	7.6727	10.3457	13.2979
40	0.3872	0.7856	1.4323	2.3826	3.7189	5.5394	7.7466	10.2911	13.2888
	0.3797	0.7747	1.4239	2.3982	3.7534	5.5230	7.7152	10.3148	13.2887

Table 3.7: European put prices from crossover lattices with 24 time steps and varying state space sizes compared to the Black-Scholes price, for $S^0 = 100$, $\sigma = 0.2$, $\mu = 0.1$ and $T = 1$

is also clear that state spaces and time steps can be varied independently of one another, with effects that are broadly in accord with intuitive expectations.

Beyond this minimal attainment of credibility, many variations on the basic construction and their effects on convergence remain to be considered, including

- different branching patterns due to different decomposition strategies
- non-constant time step sizes, particularly with a view to terminal correction
- non-constant state space sizes
- accelerated convergence by Levin u-transforms or Richardson extrapolation

Although extrapolation is potentially rewarding, it should be noted that the crossover lattice frustrates obvious attempts by non-monotonic convergence, which is not easily removed by differencing or averaging.

Chapter 4

Multifactor Models

> *Man knoweth not the price thereof; neither is it found in the land of the living.*
>
> — Job 28:13

The crucial advantage of a lattice model allowing state space sizes to be specified independently of the time step is in pricing derivatives contingent on multiple sources of risk, which occur frequently in fixed income applications, convertible bonds, and structured products. Approaches to this problem based on generalizations of conventional binomial or trinomial trees to their multidimensional analogues can lead to state explosion or wasted computational effort on states of negligible probability, which would be avoided if states could be hand picked.

This chapter describes how the crossover lattice may generalized to multifactor valuation problems and gives some numerical examples.

4.1 Inputs for a multifactor crossover lattice

Rather than a single asset price S_i^k in state i at time k, a multifactor model requires a set of prices S_{hi}^k for $0 \leq h \leq n$, where $n \in \mathbb{N}$ specifies the number of underlying assets or factors. Marginal probabilities q_i^k and state transition probabilities p_{ij}^k carry over from the single factor case, and the probability constraints in

Equations 2.4 and 2.5 are unaffected. However, the martingale condition (Equation 2.7) becomes

$$\forall k < \mathcal{T},\ \forall i \leq N_k,\ \forall h \leq n,\ d_h^k \sum_{j=0}^{N_{k+1}} p_{ij}^k S_{hj}^{k+1} = S_{hi}^k \tag{4.1}$$

which is to say that the transition probabilities are required to ensure simultaneously that every asset in every state has a price equal to its discounted expected value at the subsequent time step. The discount rate d_h^k is often independent of the asset h, but may vary in applications such as FX derivatives, so its definition generalized accordingly.

$$d_h^k = \frac{\sum_{i=0}^{N_k} q_i^k S_{hi}^k}{\sum_{j=0}^{N_{k+1}} q_j^{k+1} S_{hj}^{k+1}} \tag{4.2}$$

If the cost criterion stipulated in Equation 2.8 is an expedient feature of the single factor model, the following is more so.

$$C = \sum_{k=0}^{\mathcal{T}-1} \sum_{i=0}^{N_k} \sum_{j=0}^{N_{k+1}} p_{ij}^k \sum_{h=0}^{n} \left| \ln \frac{d_h^k S_{hj}^{k+1}}{S_{hi}^k} \right|^{\alpha} \tag{4.3}$$

That is to say, state transitions are to be preferred that cause the least total displacement of the underlying asset prices.

4.2 Simple cases

These ingredients already suffice for multifactor lattices that are small enough to be solvable for transition probabilities without decomposition, and that involve only uncorrelated assets. An example is shown in Table 4.1, which consists of three equities with the given discrete marginal distributions for two consecutive time steps. These happen to have means of approximately 20, 50, and 100, with similar drifts of 0.1 per time step, and volatilities of 0.45, 0.25, and 0.15 respectively.

Because there are three assets, each state in the multifactor model entails a triple of prices. With ten states for each asset, the full joint distribution has 1000 states, one for each possible triple. However, this size is disproportionately costly for

	earlier		later	
equity	price	probability	price	probability
Acme	6.31	4.703×10^{-4}	3.91	4.758×10^{-4}
	8.24	6.893×10^{-3}	5.75	7.297×10^{-3}
	10.77	5.077×10^{-2}	8.44	5.465×10^{-2}
	14.08	1.830×10^{-1}	12.41	1.952×10^{-1}
	18.39	3.242×10^{-1}	18.23	3.344×10^{-1}
	24.03	2.834×10^{-1}	26.79	2.756×10^{-1}
	31.41	1.223×10^{-1}	39.38	1.095×10^{-1}
	41.04	2.607×10^{-2}	57.86	2.092×10^{-2}
	53.63	2.736×10^{-3}	85.03	1.921×10^{-3}
	70.08	1.447×10^{-4}	124.96	8.623×10^{-5}
Blofeld	27.27	4.646×10^{-4}	21.64	4.675×10^{-4}
	31.55	6.484×10^{-3}	26.65	6.693×10^{-3}
	36.50	4.686×10^{-2}	32.83	4.885×10^{-2}
	42.23	1.703×10^{-1}	40.43	1.768×10^{-1}
	48.87	3.124×10^{-1}	49.79	3.186×10^{-1}
	56.54	2.901×10^{-1}	61.32	2.869×10^{-1}
	65.42	1.366×10^{-1}	75.53	1.292×10^{-1}
	75.69	3.261×10^{-2}	93.02	2.908×10^{-2}
	87.58	3.936×10^{-3}	114.56	3.267×10^{-3}
	101.33	2.474×10^{-4}	141.09	1.878×10^{-4}
Cogswell	71.17	4.617×10^{-4}	63.45	4.635×10^{-4}
	77.62	6.287×10^{-3}	71.78	6.408×10^{-3}
	84.65	4.500×10^{-2}	81.20	4.615×10^{-2}
	92.31	1.641×10^{-1}	91.85	1.679×10^{-1}
	100.67	3.061×10^{-1}	103.91	3.100×10^{-1}
	109.78	2.929×10^{-1}	117.55	2.912×10^{-1}
	119.72	1.439×10^{-1}	132.97	1.394×10^{-1}
	130.56	3.630×10^{-2}	150.42	3.397×10^{-2}
	142.39	4.694×10^{-3}	170.17	4.210×10^{-3}
	155.28	3.211×10^{-4}	192.50	2.732×10^{-4}

Table 4.1: price distributions on two consecutive time steps for three uncorrelated equities underlying a multifactor derivative

	earlier prices S_{hi}^k				later prices S_{hj}^{k+1}			
state	Acme	Blofeld	Cogswell	q_i^k	Acme	Blofeld	Cogswell	q_j^{k+1}
0	14.08	42.23	100.67	0.015288	12.41	40.43	103.91	0.016869
1	14.08	42.23	109.78	0.014627	12.41	40.43	117.55	0.015846
2	14.08	48.87	92.31	0.015036	12.41	49.79	91.85	0.016467
3	14.08	48.87	100.67	0.028046	12.41	49.79	103.91	0.030401
4	14.08	48.87	109.78	0.026833	12.41	49.79	117.55	0.028558
5	14.08	48.87	119.72	0.013184	12.41	49.79	132.97	0.013668
6	14.08	56.54	92.31	0.013966	12.41	61.32	91.85	0.014826
7	14.08	56.54	100.67	0.026051	12.41	61.32	103.91	0.027370
8	14.08	56.54	109.78	0.024924	12.41	61.32	117.55	0.025712
9	18.39	42.23	92.31	0.014523	18.23	40.43	91.85	0.015650
10	18.39	42.23	100.67	0.027090	18.23	40.43	103.91	0.028893
11	18.39	42.23	109.78	0.025919	18.23	40.43	117.55	0.027142
12	18.39	42.23	119.72	0.012735	18.23	40.43	132.97	0.012990
13	18.39	48.87	92.31	0.026642	18.23	49.79	91.85	0.028205
14	18.39	48.87	100.67	0.049695	18.23	49.79	103.91	0.052071
15	18.39	48.87	109.78	0.047547	18.23	49.79	117.55	0.048915
16	18.39	48.87	119.72	0.023362	18.23	49.79	132.97	0.023410
17	18.39	56.54	92.31	0.024747	18.23	61.32	91.85	0.025394
18	18.39	56.54	100.67	0.046160	18.23	61.32	103.91	0.046881
19	18.39	56.54	109.78	0.044165	18.23	61.32	117.55	0.044039
20	18.39	56.54	119.72	0.021700	18.23	61.32	132.97	0.021077
21	18.39	65.42	100.67	0.021738	18.23	75.53	103.91	0.021112
22	18.39	65.42	109.78	0.020798	18.23	75.53	117.55	0.019832
23	24.03	42.23	92.31	0.012694	26.79	40.43	91.85	0.012899
24	24.03	42.23	100.67	0.023678	26.79	40.43	103.91	0.023815
25	24.03	42.23	109.78	0.022654	26.79	40.43	117.55	0.022371
26	24.03	48.87	92.31	0.023287	26.79	49.79	91.85	0.023248
27	24.03	48.87	100.67	0.043436	26.79	49.79	103.91	0.042919
28	24.03	48.87	109.78	0.041558	26.79	49.79	117.55	0.040318
29	24.03	48.87	119.72	0.020420	26.79	49.79	132.97	0.019296
30	24.03	56.54	92.31	0.021630	26.79	61.32	91.85	0.020930
31	24.03	56.54	100.67	0.040347	26.79	61.32	103.91	0.038641
32	24.03	56.54	109.78	0.038602	26.79	61.32	117.55	0.036299
33	24.03	56.54	119.72	0.018967	26.79	61.32	132.97	0.017372
34	24.03	65.42	100.67	0.019000	26.79	75.53	103.91	0.017401
35	24.03	65.42	109.78	0.018178	26.79	75.53	117.55	0.016346
36	31.41	48.87	100.67	0.018751	39.38	49.79	103.91	0.017044
37	31.41	48.87	109.78	0.017940	39.38	49.79	117.55	0.016011
38	31.41	56.54	100.67	0.017417	39.38	61.32	103.91	0.015345
39	31.41	56.54	109.78	0.016664	39.38	61.32	117.55	0.014415

Table 4.2: joint distributions with ranges curtailed to the most probable 40 points

such a small example, and perhaps not even solvable. Moreover, the majority of these states have insignificantly small probabilities. One should therefore take the opportunity to dispense with some of them.

An extreme example of being selective about the state space is shown in Table 4.2. In a manner analogous to range curtailment for binomial trees [AWDN04], all states below a certain threshold probability are excluded, where state probabilities are calculated as the product of the individual price probabilities given in Table 4.1. This choice of discarded states has the dubious consequence of leaving all surviving probabilities within less than an order of magnitude of one another, and is perhaps not the best way of sampling the distribution. In practice, some attention should be paid to the derivative being valued to ensure that it has no appreciable payoff in the states being neglected, but the model mandates no preferences in this regard. The probabilities for the states remaining in the table are scaled up to unity.

Minimizing Equation 4.3 subject to the constraints in Equations 2.4, 2.5 and 4.1 for the data in Table 4.2 leads directly to the transition probabilities shown in Table 4.3. It is straightforward to verify that these data meet the constraints by routine calculation.

4.3 Decomposable cases

Multifactor models with larger state spaces require a means of decomposing them comparable to the methods described in Sections 2.3.2 and 2.5. One aspect of this problem is that of choosing the subsets l_k, m_k, and h_k on the earlier state space, and l_{k+1}, m_{k+1}, and h_{k+1} on the later one. The other is solving for the weights v_j and w_j that will allow the transition probabilities to be solvable in terms of the subsets.

4.3.1 Selection of subsets

Choosing the subsets can proceed essentially along the same lines as the cut, deal, and skim decompositions described in Section 2.5 in connection with single factor models, although the cut is not essential.

The only complication is that the cut and skim decompositions are based on the

i	j	p_{ij}^k	i	j	p_{ij}^k	i	j	p_{ij}^k	i	j	p_{ij}^k	i	j	p_{ij}^k
0	1	0.2263	8	18	0.1834	16	25	0.1034	24	15	0.1680	32	27	0.0521
0	3	0.1515	9	9	0.0858	17	14	0.0948	24	18	0.0436	32	28	0.1461
0	4	0.2406	9	10	0.4188	17	17	0.4778	24	24	0.5879	32	31	0.1429
0	9	0.3816	9	13	0.3921	17	18	0.3239	24	25	0.0737	32	32	0.4240
1	1	0.2263	9	23	0.1034	17	26	0.1034	24	26	0.1268	32	33	0.0233
1	4	0.2039	10	10	0.0877	18	2	0.1668	25	15	0.0877	33	22	0.2116
1	5	0.1882	10	11	0.4169	18	3	0.0452	25	16	0.0259	33	29	0.4588
1	10	0.3816	10	13	0.3081	18	9	0.0614	25	18	0.0980	33	32	0.2611
2	3	0.0097	10	14	0.0840	18	10	0.1482	25	25	0.7287	33	33	0.0684
2	4	0.1919	10	23	0.0169	18	13	0.0551	25	26	0.0455	34	22	0.2116
2	6	0.4168	10	24	0.0865	18	15	0.0259	25	27	0.0142	34	30	0.0928
2	9	0.3816	11	11	0.4188	18	18	0.0995	26	15	0.0453	34	31	0.4154
3	4	0.3661	11	12	0.0857	18	21	0.0312	26	18	0.1663	34	34	0.2802
3	6	0.2058	11	14	0.1623	18	22	0.1376	26	25	0.0732	35	22	0.2116
3	7	0.1373	11	15	0.2298	18	35	0.2165	26	26	0.7152	35	32	0.4096
3	10	0.2291	11	24	0.1034	18	38	0.0126	27	10	0.0366	35	33	0.0986
3	23	0.0618	12	12	0.5045	19	15	0.3354	27	14	0.5479	35	34	0.2802
4	4	0.0735	12	15	0.2779	19	18	0.1206	27	15	0.0075	36	27	0.2038
4	5	0.2869	12	16	0.1141	19	19	0.4327	27	27	0.1493	36	28	0.2241
4	7	0.2580	12	24	0.1034	19	22	0.0080	27	37	0.1221	36	30	0.1069
4	11	0.1861	13	13	0.2870	19	34	0.0482	27	38	0.1366	36	36	0.4651
4	13	0.0450	13	14	0.4188	19	35	0.0552	28	15	0.2116	37	28	0.1968
4	14	0.1506	13	17	0.1909	20	16	0.0948	28	27	0.2897	37	29	0.2312
5	5	0.2440	13	23	0.1034	20	19	0.3191	28	28	0.3918	37	31	0.1069
5	8	0.3744	14	0	0.3394	20	20	0.4827	28	33	0.1069	37	36	0.3233
5	12	0.3294	14	2	0.1764	20	27	0.0569	29	16	0.1047	37	37	0.1418
5	14	0.0522	14	3	0.1985	20	28	0.0465	29	20	0.1069	38	31	0.4723
6	6	0.1996	14	35	0.0787	21	18	0.1838	29	28	0.4727	38	32	0.0038
6	7	0.4188	14	39	0.2068	21	19	0.2210	29	29	0.3157	38	34	0.0588
6	13	0.1982	15	1	0.1909	21	21	0.4918	30	18	0.2037	38	36	0.1448
6	17	0.1834	15	3	0.3362	21	27	0.0000	30	19	0.0079	38	37	0.1258
7	7	0.4126	15	4	0.1424	21	30	0.1034	30	27	0.1982	38	38	0.1945
7	8	0.2058	15	16	0.0125	22	20	0.4048	30	30	0.5902	39	32	0.4481
7	14	0.1366	15	33	0.1540	22	21	0.4319	31	18	0.0911	39	33	0.0868
7	15	0.0616	15	37	0.0561	22	22	0.0600	31	19	0.1205	39	37	0.1982
7	17	0.1559	15	38	0.0209	22	30	0.1034	31	27	0.3152	39	38	0.2669
7	18	0.0274	15	39	0.0870	23	15	0.0651	31	31	0.3739			
8	8	0.6184	16	15	0.0097	23	17	0.1465	31	33	0.0043			
8	15	0.1853	16	16	0.6960	23	23	0.5083	31	34	0.0950			
8	16	0.0129	16	18	0.1909	23	24	0.2801	32	19	0.2116			

Table 4.3: state transition probabilities for the joint distributions in Table 4.2

assumption of a two tailed distribution, which does not apply to the multifactor case. A simple way of resolving this issue is to sort the states in order of probability and make an arbitrary selection of states from the low end to constitute one tail or the other. However, an inappropriate selection may render the martingale condition infeasible when transition probabilities are sought. A better approach is to divide the space among nearby states by employing some form of cluster analysis, which is a well studied topic with robust implementations that are freely available [vD00].

4.3.2 Solving for weights

Similarly to the single factor case, when the selection of subsets has been made, the necessary weights v_j and w_j can be computed by solving a linear programming problem. The probability constraints given by Equations 2.15 through 2.16 are unchanged for the multifactor case, but the martingale constraints given by Equations 2.17 through 2.18 generalize to the following.

$$\sum_{i \in h_k} q_i^k \sum_{h=0}^{n} S_{hi}^k = \sum_{j \in h_{k+1} \cap m_{k+1}} v_j q_j^{k+1} \sum_{h=0}^{n} d_h^k S_{hj}^{k+1} + \sum_{j \in h_{k+1} - m_{k+1}} q_j^{k+1} \sum_{h=0}^{n} d_h^k S_{hj}^{k+1} \tag{4.4}$$

$$\begin{aligned} \sum_{i \in m_k} q_i^k \sum_{h=0}^{n} S_{hi}^k = & \sum_{j \in h_{k+1} \cap m_{k+1}} (1 - v_j) q_j^{k+1} \sum_{h=0}^{n} d_h^k S_{hj}^{k+1} \\ & + \sum_{j \in m_{k+1} - h_{k+1} - l_{k+1}} q_j^{k+1} \sum_{h=0}^{n} d_h^k S_{hj}^{k+1} \\ & + \sum_{j \in m_{k+1} \cap l_{k+1}} w_j q_j^{k+1} \sum_{h=0}^{n} d_h^k S_{hj}^{k+1} \end{aligned} \tag{4.5}$$

$$\begin{aligned} \sum_{i \in l_k} q_i^k \sum_{h=0}^{n} S_{hi}^k = & \sum_{j \in m_{k+1} \cap l_{k+1}} (1 - w_j) q_j^{k+1} \sum_{h=0}^{n} d_h^k S_{hj}^{k+1} \\ & + \sum_{j \in l_{k+1} - m_{k+1}} q_j^{k+1} \sum_{h=0}^{n} d_h^k S_{hj}^{k+1} \end{aligned} \tag{4.6}$$

The cost criterion in Equation 2.19 carries over to multifactor models without modification, as do the subproblems in Section 2.3.3 except that the martingale conditions in Equations 2.24, 2.29 and 2.35 generalize consistently with Equation 4.1.

$$\forall i \in h_k,\ \forall h \leq n,\ d_h^k \sum_{j \in h_{k+1}} p_{ij}^k S_{hj}^{k+1} = S_{hi}^k \tag{4.7}$$

$$\forall i \in l_k,\ \forall h \leq n,\ d_h^k \sum_{j \in l_{k+1}} p_{ij}^k S_{hj}^{k+1} = S_{hi}^k \tag{4.8}$$

$$\forall i \in m_k,\ \forall h \leq n,\ d_h^k \sum_{j \in m_{k+1}} p_{ij}^k S_{hj}^{k+1} = S_{hi}^k \tag{4.9}$$

4.4 Summary

A sketch of crossover lattice specifications generalized to multifactor valuation problems is provided in this chapter, mainly by drawing on the constructions in Chapter 2 and making a note of where they differ. A numerical example establishes that there is no inherent barrier to the application of this model in a multifactor setting where asset prices are uncorrelated and decomposition of the state space is not required. The example also demonstrates that tunable space efficiency is readily attainable by truncating or sampling the state spaces according to chosen criteria (for example, by excluding states of low probability).

A theoretical description of multifactor state space decomposition is given, but this subject awaits the settlement of a minor question by an implementation, namely that of generalizing the skim decomposition (Section 2.5.5) to higher dimensions. The answer may entail cluster analysis or some simpler heuristic solution.

Chapter 5

Concluding Remarks

Lattice models of the type considered in this dissertation would seem to be a relatively unexplored subject that deserves further investigation. The present treatment at best constitutes a tour of some of the main implementation issues accompanied by some informal excursions around the theory. It is nevertheless hoped to be of use to researchers and practitioners faced with problems in derivatives valuation that may be out of the ordinary. To this end, some concluding remarks are offered on the possible merits and probable deficiencies of this endeavor.

5.1 Decomposition strategies

Although the decomposition strategies described in Section 2.5 do a superb job of allowing large linear programming problems and numerical instability to be avoided, they introduce performance bottlenecks of their own, as Table 5.1 shows. These figures pertain to a research prototype implementation in which the `glpk` linear programming solver is driven by code written in a (slow) interpreted programming language, and include the time required for compilation. Times may vary considerably depending on the decomposition strategy selected. These figures refer to a single time step and should be multiplied by the total number of time steps in the lattice in order to estimate the time required to build it. Once built, the lattice is usable for any number of derivative valuations (in seconds or less) until changes in the marginal distributions due to market conditions necessitate a new version.

states	time
40	10 - 15 seconds
100	30 - 45 seconds
256	2 - 8 minutes
512	1 hour
1024	8.5 hours

Table 5.1: approximate times to compute the transition probabilities for a single time step of a crossover lattice on a 1.1 Ghz Pentium processor

states	time
10	0:53.49
20	1:22.54
40	2:20.20
100	5:13.00
200	10:05.35

Table 5.2: representative example times in minutes and seconds to optimize the moment matching discretization of a lognormal density function needed for a single time step in the lattice on a 1.1 Ghz Pentium processor

Profile statistics show that the main hot spot in the code for building the lattice is the search phase of the skimming decomposition (Section 2.5.2, page 46). Although the linear programming problems solved or attempted in this phase are small, quadratically many of them need to be done in the worst case. This case occurs in the course of any decomposition when further skimming is infeasible, which is established only by testing all potentially worthy candidates unsuccessfully. As noted, a bisecting search strategy provides some improvement, but care in finding the minimal sized solution is essential to the overall effectiveness of the decomposition. A better approach would be a more direct decision procedure for infeasibility based on a firm theoretical foundation.

5.2 Distribution calibration parameters

In cases where the marginal distributions are given in continuous form, the other main performance bottleneck associated with crossover lattices is in calibrating discrete versions of the marginals as described in Section 3.1. Some representative computation times are shown in Table 5.2. These figures are based on numerical integration performed by `quadpack` via the GNU Scientific Library (`gsl`)

and driven by a research prototype implementation in an interpreted language. The times shown here suffice for results accurate to at least six decimal places, and could be shortened if lower accuracies are acceptable. A fixed number of iterations is used for all state space sizes regardless of convergence rates.

The idea proposed in Section 3.1.4 of calculating the κ and ω parameters in advance and interpolating leaves open the question of what form of interpolation function should be chosen. A straightforward cubic spline is far too unstable for the accuracies required of these parameters, but many alternatives are available for families of interpolating polynomials or other basis functions. When none of these turns out to be a magic bullet, a re-examination of the particular forms of Equations 3.8 and 3.9 may be necessary. These are an entirely *ad hoc* choice which could be replaced by any number of alternatives with similar pushing or tilting effects to those noted. A choice based on better numerical properties suitable for interpolation would be a worthwhile advance.

5.3 Convergence properties

The crossover lattice allows greater flexibility in the choice of time intervals and state space sizes than other lattice models. The effects of these choices on convergence call for a proper investigation as noted in Section 3.3.3. In particular, intuition suggests that fewer states should be needed at the earlier time steps, which implies the notion of an optimal growth pattern.

Appendix A

Supplementary data

Readers wishing to replicate the results from Chapter 2 as a way of testing their own implementations of crossover lattices may find it helpful to use the exact binary representations for the numbers in Table 2.6, which are shown in Tables A.1 and A.2. This format should be readable by any standard C compiler.

The remainder of this appendix contains the data and results mentioned in Section 2.5.

	sooner	
state	price	probability
0	0x1.43ca4355e67b9p+6	0x1.2748044da59a6p-23
1	0x1.47638a78f9ef1p+6	0x1.c8f3cb32257e2p-22
2	0x1.4b070eea68682p+6	0x1.a93d16c975a1cp-20
3	0x1.4eb4edcc94942p+6	0x1.6fbbfed933447p-18
4	0x1.526d4494c66dep+6	0x1.2783781202b56p-16
5	0x1.5630310c17178p+6	0x1.b95c7e4ad1e1ep-15
6	0x1.59fdd1505f594p+6	0x1.3249222cb849fp-13
7	0x1.5dd643d528c1p+6	0x1.8b0930a6fabadp-12
8	0x1.61b9a764a1758p+6	0x1.d9793798b2438p-11
9	0x1.65a81b2092bb9p+6	0x1.07addc82d1af6p-9
10	0x1.69a1be835a38bp+6	0x1.10ec07b167268p-8
11	0x1.6da6b160e5fe6p+6	0x1.06840ef700f51p-7
12	0x1.71b713e7b35dap+6	0x1.d54ef26df6208p-7
13	0x1.75d306a1d094fp+6	0x1.85d77c19c3c4dp-6
14	0x1.79faaa75e1556p+6	0x1.2cf01dae49b2ep-5
15	0x1.7e2e20a826339p+6	0x1.afc5e403b0ef8p-5
16	0x1.826d8adb86ff8p+6	0x1.1fd96a3e8b914p-4
17	0x1.86b90b12a0186p+6	0x1.64ab9db5199b6p-4
18	0x1.8b10c3b0d2ba9p+6	0x1.9ab5056562d91p-4
19	0x1.8f74d77b5854bp+6	0x1.b780b86fae705p-4
20	0x1.93e5699a58efap+6	0x1.b5134391f0213p-4
21	0x1.98629d9a04a94p+6	0x1.93f098236c6a2p-4
22	0x1.9cec976bb055dp+6	0x1.5aee75f2623fcp-4
23	0x1.a1837b66f5456p+6	0x1.14e8d993e3b08p-4
24	0x1.a6276e4ad441ap+6	0x1.9accad1219bc8p-5
25	0x1.aad8953edbc3dp+6	0x1.1b2da78e76881p-5
26	0x1.af9715d4516fep+6	0x1.6ad15cdef1ffbp-6
27	0x1.b46316075ee1dp+6	0x1.b0008ecb5914bp-7
28	0x1.b93cbc4041ccfp+6	0x1.de06d6d9b5052p-8
29	0x1.be242f547f834p+6	0x1.eb9281ad04a19p-9
30	0x1.c31996881be08p+6	0x1.d5c6e01bc5e6ep-10
31	0x1.c81d198ed3b12p+6	0x1.a1383b84408a5p-11
32	0x1.cd2ee08d5a92dp+6	0x1.585aefebd6998p-12
33	0x1.d24f141a9c5dbp+6	0x1.08212f804f75ep-13
34	0x1.d77ddd4102248p+6	0x1.788d7ff49014dp-15
35	0x1.dcbb657fbacafp+6	0x1.f2e31e6a689fep-17
36	0x1.e207d6cc07499p+6	0x1.332034ff3ac49p-18
37	0x1.e7635b928a9cbp+6	0x1.5f6bab1763f23p-20
38	0x1.ecce1eb89d758p+6	0x1.75af15b81f6acp-22
39	0x1.f2484b9da5ab5p+6	0x1.dc860f7p-24

Table A.1: test data from Table 2.6 in machine readable form

	later	
state	price	probability
0	0x1.293a24271475bp+6	0x1.278badbf683cfp-23
1	0x1.2dead1d08705cp+6	0x1.caad0753fdee4p-22
2	0x1.32ae71bc538f4p+6	0x1.abaabbd0f24fp-20
3	0x1.378550739447bp+6	0x1.72761874833cp-18
4	0x1.3c6fbbb48fabap+6	0x1.2a23b6c2fbf3p-16
5	0x1.416e0277996e2p+6	0x1.bdd40db60a129p-15
6	0x1.468074f4071d2p+6	0x1.35b102d6a8b91p-13
7	0x1.4ba764a538cabp+6	0x1.8fbab74d6b462p-12
8	0x1.50e3244fb60c1p+6	0x1.df58d336695e5p-11
9	0x1.563408065fa0bp+6	0x1.0b0689af76cc9p-9
10	0x1.5b9a652fb613dp+6	0x1.146617fa55488p-8
11	0x1.6116928b35afp+6	0x1.09cf6b1f238a9p-7
12	0x1.66a8e836c812p+6	0x1.daff3fde1de34p-7
13	0x1.6c51bfb44bc8ep+6	0x1.8a4ecb5297aa6p-6
14	0x1.721173ef3237ap+6	0x1.301d2c81bb60bp-5
15	0x1.77e8614234379p+6	0x1.b3d69050f1f1fp-5
16	0x1.7dd6e57d1dbfdp+6	0x1.2229ee2c13952p-4
17	0x1.83dd5feab0f83p+6	0x1.66f6e62350dd8p-4
18	0x1.89fc3156a1137p+6	0x1.9c97971878c0ap-4
19	0x1.9033bc13a551dp+6	0x1.b89bbe2d9fd35p-4
20	0x1.96846401a48cfp+6	0x1.b52ad40281957p-4
21	0x1.9cee8e93f9b0dp+6	0x1.9300597e28d27p-4
22	0x1.a372a2d7d186p+6	0x1.592ae656047cfp-4
23	0x1.aa11097aa233ap+6	0x1.12acbb0595597p-4
24	0x1.b0ca2cd0bce0ap+6	0x1.962a66414f45ep-5
25	0x1.b79e78dbf9deep+6	0x1.1703396309059p-5
26	0x1.be8e5b527fc99p+6	0x1.6428158fbe6c9p-6
27	0x1.c59a43a5a605cp+6	0x1.a666a64ac0877p-7
28	0x1.ccc2a308f3124p+6	0x1.d173f53e1da81p-8
29	0x1.d407ec7937175p+6	0x1.dc88a722323ecp-9
30	0x1.db6a94c3c329bp+6	0x1.c54ae951aab8cp-10
31	0x1.e2eb128dbdb25p+6	0x1.909e3a13868ecp-11
32	0x1.ea89de5b94739p+6	0x1.48f6f25a9cb79p-12
33	0x1.f24772988ca0ap+6	0x1.f5f4604e02101p-14
34	0x1.fa244b9e7183p+6	0x1.63cee35ed0afep-15
35	0x1.011073deb113dp+7	0x1.d4aa6ef89197ep-17
36	0x1.051ee3a1df496p+7	0x1.1ec73080d2629p-18
37	0x1.093db6431a7bep+7	0x1.46145e92e1f75p-20
38	0x1.0d6d2df3f7bc4p+7	0x1.587b306c8e5a8p-22
39	0x1.11ad8df171773p+7	0x1.b3747d8p-24

Table A.2: more test data from Table 2.6 in machine readable form

i	j	p_{ij}	i	j	p_{ij}	i	j	p_{ij}
0	5	0.2680873838259	18	19	0.6695055524885	21	24	0.0622969320396
0	6	0.7319126161741	19	0	0.0000012826051	22	21	0.8552976178646
1	6	0.5625055153790	19	1	0.0000039811114	22	22	0.0278287070343
1	7	0.4374944846210	19	2	0.0000148478673	22	23	0.1168736751011
2	7	0.8555715453373	19	3	0.0000514471426	23	21	0.0293014429213
2	8	0.1444284546627	19	4	0.0001656143336	23	22	0.9706985570787
3	7	0.1496158013081	19	5	0.0004949652681	24	22	0.3248163634194
3	8	0.8503841986919	19	6	0.0013730819314	24	23	0.6751836365806
4	8	0.4445781440798	19	7	0.0035307410524	25	23	0.6189740480475
4	9	0.5554218559202	19	8	0.0084022204320	25	24	0.3810259519525
5	9	0.7381857888511	19	9	0.0184911008475	26	24	0.9117807302953
5	10	0.2618142111489	19	10	0.0367120888886	26	25	0.0882192697047
6	9	0.0309253692568	19	11	0.0680117302293	27	24	0.2064497192178
6	10	0.9690746307432	19	12	0.1067018278061	27	25	0.7935502807822
7	10	0.3264328314138	19	13	0.1543945812507	28	25	0.5011510361439
7	11	0.6735671685862	19	14	0.1617483750498	28	26	0.4988489638561
8	11	0.6205830919554	19	25	0.0870772814954	29	26	0.7944988539046
8	12	0.3794169080446	19	26	0.1394426817415	29	27	0.2055011460954
9	12	0.9133823842138	19	27	0.0948781488452	30	26	0.0878643218300
9	13	0.0866176157862	19	28	0.0595431498036	30	27	0.9121356781700
10	12	0.2080691746039	19	29	0.0316456240760	31	27	0.3831102757400
10	13	0.7919308253961	19	30	0.0156944538770	31	28	0.6168897242600
11	13	0.5027630537230	19	31	0.0070220859636	32	28	0.6770002290851
11	14	0.4972369462770	19	32	0.0029056563071	32	29	0.3229997709149
12	14	0.7961034678369	19	33	0.0011127566452	33	29	0.9695404096815
12	15	0.2038965321631	19	34	0.0003950236612	33	30	0.0304595903185
13	14	0.0894867701287	19	35	0.0001301702137	34	29	0.2648513644149
13	15	0.9105132298713	19	36	0.0000398257863	34	30	0.7351486355851
14	15	0.3847252724859	19	37	0.0000113209272	35	30	0.5592844552643
14	16	0.6152747275141	19	38	0.0000029899502	35	31	0.4407155447357
15	16	0.6786078085016	19	39	0.0000009448911	36	31	0.8523652788524
15	17	0.3213921914984	20	14	0.0211127666053	36	32	0.1476347211476
16	17	0.9711406058348	20	15	0.0739634534365	37	31	0.1463738990045
16	18	0.0288593941652	20	19	0.3789683303432	37	32	0.8536261009955
17	16	0.1200955451612	20	20	0.3760325549697	38	32	0.4413511311297
17	17	0.0281438861745	20	24	0.0689092687308	38	33	0.5586488688703
17	18	0.8517605686642	20	25	0.0810136259146	39	33	0.7349735968708
18	15	0.0657450215077	21	20	0.6753797475168	39	34	0.2650264031292
18	16	0.0200805887219	21	21	0.2430011496641			
18	18	0.2446688372818	21	23	0.0193221707795			

Table A.3: transition probabilities for an implied willow tree based on Table 2.6

j	q_j^{k+1}	$\sum_{i=0}^{N_k} p_{ij}^k q_i$
0	$1.3762415383442527\times10^{-7}$	$1.3762415383442527\times10^{-7}$
1	$4.2717521364598702\times10^{-7}$	$4.2717521366112631\times10^{-7}$
2	$1.5931834591468337\times10^{-6}$	$1.5931834591468337\times10^{-6}$
3	$5.5203037061636400\times10^{-6}$	$5.5203037061636400\times10^{-6}$
4	$1.7770499436352966\times10^{-5}$	$1.7770499436352966\times10^{-5}$
5	$5.3146878983336490\times10^{-5}$	$5.3146878983336504\times10^{-5}$
6	$1.4767239072907256\times10^{-4}$	$1.4767239072907258\times10^{-4}$
7	$3.8121162364285731\times10^{-4}$	$3.8121162364285731\times10^{-4}$
8	$9.1428179330382538\times10^{-4}$	$9.1428179330382559\times10^{-4}$
9	$2.0372431824689824\times10^{-3}$	$2.0372431824689824\times10^{-3}$
10	$4.2175110377985561\times10^{-3}$	$4.2175110377985561\times10^{-3}$
11	$8.1118844084609610\times10^{-3}$	$8.1118844084609610\times10^{-3}$
12	$1.4495760140774562\times10^{-2}$	$1.4495760140774564\times10^{-2}$
13	$2.4066637570622128\times10^{-2}$	$2.4066637570622128\times10^{-2}$
14	$3.7123286177898583\times10^{-2}$	$3.7123286177898583\times10^{-2}$
15	$5.3202897899210060\times10^{-2}$	$5.3202897899210094\times10^{-2}$
16	$7.0840769157461098\times10^{-2}$	$7.0840769157461167\times10^{-2}$
17	$8.7637804962518273\times10^{-2}$	$8.7637804962518301\times10^{-2}$
18	$1.0073050519672236\times10^{-1}$	$1.0073050519672233\times10^{-1}$
19	$1.0757040298812089\times10^{-1}$	$1.0757040298812084\times10^{-1}$
20	$1.0673029724430215\times10^{-1}$	$1.0673029724430205\times10^{-1}$
21	$9.8389005261698345\times10^{-2}$	$9.8389005261698387\times10^{-2}$
22	$8.4269428014777534\times10^{-2}$	$8.4269428014777603\times10^{-2}$
23	$6.7059259942956792\times10^{-2}$	$6.7059259942956806\times10^{-2}$
24	$4.9580764500640070\times10^{-2}$	$4.9580764500640029\times10^{-2}$
25	$3.4059154590324332\times10^{-2}$	$3.4059154590324332\times10^{-2}$
26	$2.1738072448875188\times10^{-2}$	$2.1738072448875191\times10^{-2}$
27	$1.2890654752031090\times10^{-2}$	$1.2890654752031090\times10^{-2}$
28	$7.1022485482133977\times10^{-3}$	$7.1022485482133985\times10^{-3}$
29	$3.6356643696291348\times10^{-3}$	$3.6356643696291353\times10^{-3}$
30	$1.7291741282103616\times10^{-3}$	$1.7291741282103616\times10^{-3}$
31	$7.6411833510128596\times10^{-4}$	$7.6411833510128607\times10^{-4}$
32	$3.1372512390886277\times10^{-4}$	$3.1372512390886282\times10^{-4}$
33	$1.1967530119261564\times10^{-4}$	$1.1967530119261564\times10^{-4}$
34	$4.2415637600009743\times10^{-5}$	$4.2415637600009749\times10^{-5}$
35	$1.3967327945907335\times10^{-5}$	$1.3967327945907337\times10^{-5}$
36	$4.2733264543153883\times10^{-6}$	$4.2733264543153883\times10^{-6}$
37	$1.2147410514990309\times10^{-6}$	$1.2147410514990311\times10^{-6}$
38	$3.2082312520809555\times10^{-7}$	$3.2082312520809555\times10^{-7}$
39	$1.0138727724040564\times10^{-7}$	$1.0138727724040564\times10^{-7}$

Table A.4: verification of willow tree transition probabilities from Table A.3, with root mean squared error 5.793×10^{-16}

i	$\sum_{j=0}^{N_{k+1}} p_{ij}^k$	S_i^k	$d_k \sum_{j=0}^{N_{k+1}} p_{ij}^k S_j^{k+1}$
0	1.00000000000000000	80.9475224897313552	80.9475223667286770
1	1.00000000000000000	81.8472079184100636	81.8472079184100920
2	1.00000000000000000	82.7568928362107101	82.7568928362107101
3	1.00000000000000000	83.6766883817372502	83.6766883817372502
4	1.00000000000000000	84.6067069288360187	84.6067069288360329
5	1.00000000000000000	85.5470621003240694	85.5470621003240694
6	1.00000000000000000	86.4978687818714320	86.4978687818714320
7	1.00000000000000000	87.4592431360363207	87.4592431360363065
8	1.00000000000000000	88.4313026164576286	88.4313026164576428
9	1.00000000000000000	89.4141659822042953	89.4141659822043096
10	1.00000000000000000	90.4079533122841639	90.4079533122841781
11	1.00000000000000000	91.4127860203148828	91.4127860203148828
12	1.00000000000000000	92.4287868693567418	92.4287868693567276
13	1.00000000000000000	93.4560799869116039	93.4560799869116039
14	1.00000000000000000	94.4947908800871517	94.4947908800871517
15	1.00000000000000000	95.5450464509311956	95.5450464509312098
16	1.00000000000000000	96.6069750119348782	96.6069750119348640
17	1.00000000000000000	97.6807063017095913	97.6807063017095771
18	1.00000000000000000	98.7663715008371952	98.7663715008372094
19	0.99999999999999989	99.8641032478964661	99.8641032478964519
20	1.00000000000000022	100.9740356556684731	100.9740356556685441
21	0.99999999999999922	102.0963043275208406	102.0963043275207838
22	1.00000000000000000	103.2310463739754169	103.2310463739754312
23	1.00000000000000000	104.3784004294587646	104.3784004294587646
24	1.00000000000000000	105.5385066692400926	105.5385066692400926
25	1.00000000000000000	106.7115068265565725	106.7115068265566009
26	1.00000000000000000	107.8975442099290092	107.8975442099290092
27	0.99999999999999978	109.0967637206708361	109.0967637206708076
28	1.00000000000000133	110.3093118705903493	110.3093118705904914
29	1.00000000000000000	111.5353367998912404	111.5353367998912404
30	1.00000000000000000	112.7749882952704183	112.7749882952704326
31	1.00000000000000000	114.0284178082186202	114.0284178082186202
32	1.00000000000000000	115.2957784735232991	115.2957784735232991
33	1.00000000000000000	116.5772251279772007	116.5772251279772149
34	1.00000000000000000	117.8729143292958952	117.8729143292958952
35	1.00000000000000000	119.1830043752440673	119.1830043752440957
36	1.00000000000000000	120.5076553229759071	120.5076553229759071
37	1.00000000000000000	121.8470290085889900	121.8470290085889900
38	1.00000000000000000	123.2012890668969476	123.2012890668969618
39	1.00000000000000000	124.5706009514204453	124.5706009514204453

Table A.5: verification of willow tree transition probabilities from Table A.3, with RMS errors of 3.945×10^{-17} and 3.418×10^{-10}

i	j	p_{ij}	i	j	p_{ij}	i	j	p_{ij}
0	5	0.2680872864165	14	15	0.2601493155300	26	25	0.0882192697047
0	6	0.7319127135846	14	16	0.6770751107399	27	24	0.8999731515475
1	6	0.5625055162098	15	16	0.6786078085016	27	30	0.0054465286546
1	7	0.4374944837908	15	17	0.3213921914984	27	31	0.0571520551334
2	7	0.8555715453372	16	17	0.9711406058348	27	32	0.0236488459870
2	8	0.1444284546628	16	18	0.0288593941652	27	33	0.0090566150095
3	7	0.1496158013077	17	14	0.0014975534624	27	34	0.0032150580581
3	8	0.8503841986922	17	16	0.1171466044919	27	35	0.0010594423466
4	8	0.4445781440796	17	17	0.0281438861745	27	36	0.0003241380903
4	9	0.5554218559204	17	18	0.8532119558712	27	37	0.0000921398936
5	9	0.7381857888509	18	14	0.0655010082077	27	38	0.0000243349054
5	10	0.2618142111491	18	18	0.2434084091020	27	39	0.0000076903739
6	9	0.0309253692566	18	19	0.6910905826903	28	24	0.7543377471624
6	10	0.9690746307434	19	11	0.0732183769633	28	29	0.0246321757276
7	10	0.3264328314137	19	12	0.1350950019164	28	30	0.2210300771100
7	11	0.6735671685863	19	13	0.2242919596593	29	24	0.5682780717027
8	10	0.7299557375679	19	14	0.1261479815399	29	29	0.4317219282973
8	11	0.0019486986739	19	25	0.0543990284946	30	24	0.4269858020080
8	15	0.2680955637583	19	26	0.2025906134360	30	29	0.5730141979920
9	10	0.5920489505152	19	27	0.1201360267773	31	24	0.2841231495231
9	15	0.4079510494848	19	28	0.0641210112131	31	29	0.7158768504769
10	10	0.4510208805469	20	14	0.1167110892796	32	24	0.0001860883505
10	15	0.5489791194531	20	19	0.3586854947098	32	28	0.6760982497311
11	9	0.2369225033346	20	20	0.3558962615891	32	29	0.3237156619184
11	10	0.0263036044602	20	25	0.1687071544215	33	29	0.9695404096815
11	15	0.7367738922052	21	20	0.6971678332459	33	30	0.0304595903185
12	0	0.0000096091790	21	21	0.2417301365369	34	29	0.2648513644149
12	1	0.0000298261821	21	25	0.0611020302172	34	30	0.7351486355851
12	2	0.0001112390852	22	21	0.8567774858920	35	30	0.5592844552644
12	3	0.0003854380552	22	22	0.0278287070343	35	31	0.4407155447356
12	4	0.0012407699118	22	23	0.1139595600990	36	31	0.8523652788524
12	5	0.0037082419074	22	25	0.0014342469747	36	32	0.1476347211476
12	6	0.0102870247396	23	21	0.0293014429213	37	31	0.1463738990045
12	7	0.0264520417370	23	22	0.9706985570787	37	32	0.8536261009955
12	8	0.0629487924070	24	22	0.3248163634194	38	32	0.4413511311297
12	9	0.0060071667845	24	23	0.6751836365806	38	33	0.5586488688703
12	15	0.8888198500113	25	23	0.6812386444839	39	33	0.7349735968708
13	14	0.0894867701287	25	24	0.2574640117800	39	34	0.2650264031292
13	15	0.9105132298713	25	25	0.0612973437361			
14	14	0.0627755737302	26	24	0.9117807302953			

Table A.6: transition probabilities based the cut decomposition in Table 2.7 of data in Table 2.6

j	q_j^{k+1}	$\sum_{i=0}^{N_k} p_{ij}^k q_i$
0	$1.3762415383442527\times10^{-7}$	$1.3762415383442620\times10^{-7}$
1	$4.2717521364598702\times10^{-7}$	$4.2717521364644495\times10^{-7}$
2	$1.5931834591468337\times10^{-6}$	$1.5931834591472438\times10^{-6}$
3	$5.5203037061636400\times10^{-6}$	$5.5203037061636070\times10^{-6}$
4	$1.7770499436352966\times10^{-5}$	$1.7770499436352454\times10^{-5}$
5	$5.3146878983336490\times10^{-5}$	$5.3146878983336416\times10^{-5}$
6	$1.4767239072907256\times10^{-4}$	$1.4767239072907261\times10^{-4}$
7	$3.8121162364285731\times10^{-4}$	$3.8121162364285638\times10^{-4}$
8	$9.1428179330382538\times10^{-4}$	$9.1428179330382679\times10^{-4}$
9	$2.0372431824689824\times10^{-3}$	$2.0372431824689863\times10^{-3}$
10	$4.2175110377985561\times10^{-3}$	$4.2175110377985639\times10^{-3}$
11	$8.1118844084609610\times10^{-3}$	$8.1118844084609645\times10^{-3}$
12	$1.4495760140774562\times10^{-2}$	$1.4495760140774260\times10^{-2}$
13	$2.4066637570622128\times10^{-2}$	$2.4066637570622211\times10^{-2}$
14	$3.7123286177898583\times10^{-2}$	$3.7123286177897535\times10^{-2}$
15	$5.3202897899210060\times10^{-2}$	$5.3202897899209553\times10^{-2}$
16	$7.0840769157461098\times10^{-2}$	$7.0840769157461819\times10^{-2}$
17	$8.7637804962518273\times10^{-2}$	$8.7637804962516927\times10^{-2}$
18	$1.0073050519672236\times10^{-1}$	$1.0073050519672226\times10^{-1}$
19	$1.0757040298812089\times10^{-1}$	$1.0757040298812050\times10^{-1}$
20	$1.0673029724430215\times10^{-1}$	$1.0673029724430316\times10^{-1}$
21	$9.8389005261698345\times10^{-2}$	$9.8389005261697970\times10^{-2}$
22	$8.4269428014777534\times10^{-2}$	$8.4269428014777215\times10^{-2}$
23	$6.7059259942956792\times10^{-2}$	$6.7059259942957278\times10^{-2}$
24	$4.9580764500640070\times10^{-2}$	$4.9580764500639869\times10^{-2}$
25	$3.4059154590324332\times10^{-2}$	$3.4059154590324048\times10^{-2}$
26	$2.1738072448875188\times10^{-2}$	$2.1738072448874990\times10^{-2}$
27	$1.2890654752031090\times10^{-2}$	$1.2890654752031092\times10^{-2}$
28	$7.1022485482133977\times10^{-3}$	$7.1022485482134029\times10^{-3}$
29	$3.6356643696291348\times10^{-3}$	$3.6356643696291340\times10^{-3}$
30	$1.7291741282103616\times10^{-3}$	$1.7291741282103614\times10^{-3}$
31	$7.6411833510128596\times10^{-4}$	$7.6411833510128585\times10^{-4}$
32	$3.1372512390886277\times10^{-4}$	$3.1372512390886282\times10^{-4}$
33	$1.1967530119261564\times10^{-4}$	$1.1967530119261562\times10^{-4}$
34	$4.2415637600009743\times10^{-5}$	$4.2415637600009736\times10^{-5}$
35	$1.3967327945907335\times10^{-5}$	$1.3967327945907335\times10^{-5}$
36	$4.2733264543153883\times10^{-6}$	$4.2733264543153883\times10^{-6}$
37	$1.2147410514990309\times10^{-6}$	$1.2147410514990309\times10^{-6}$
38	$3.2082312520809555\times10^{-7}$	$3.2082312520809555\times10^{-7}$
39	$1.0138727724040564\times10^{-7}$	$1.0138727724040564\times10^{-7}$

Table A.7: verification of cut willow tree transition probabilities from Table A.6, with RMS error 3.304×10^{-16}

i	$\sum_{j=0}^{N_{k+1}} p_{ij}^k$	S_i^k	$d_k \sum_{j=0}^{N_{k+1}} p_{ij}^k S_j^{k+1}$
0	1.00000000000110112	80.9475224897313552	80.9475224898211110
1	1.00000000000065947	81.8472079184100636	81.8472079173988050
2	1.00000000000000822	82.7568928362107101	82.7568928362114775
3	0.99999999999994305	83.6766883817372502	83.6766883817329159
4	0.99999999999998801	84.6067069288360187	84.6067069288352229
5	1.00000000000000089	85.5470621003240694	85.5470621003242968
6	1.00000000000000000	86.4978687818714320	86.4978687818715883
7	0.99999999999999889	87.4592431360363207	87.4592431360363776
8	1.00000000000000000	88.4313026164576286	88.4313026164577849
9	1.00000000000000000	89.4141659822042953	89.4141659822044659
10	0.99999999999999944	90.4079533122841639	90.4079533122842776
11	1.00000000000000000	91.4127860203148828	91.4127860203150533
12	1.00000000000000000	92.4287868693567418	92.4287868693569123
13	1.00000000000000600	93.4560799869116039	93.4560799869121723
14	1.00000000000000000	94.4947908800871517	94.4947908800871943
15	0.99999999999998324	95.5450464509311956	95.5450464509295898
16	0.99999999999999978	96.6069750119348782	96.6069750119348356
17	0.99999999999999023	97.6807063017095913	97.6807063017086250
18	0.99999999999999800	98.7663715008371952	98.7663715008370104
19	0.99999999999999978	99.8641032478964661	99.8641032478964235
20	1.00000000000000555	100.9740356556684731	100.9740356556690415
21	0.99999999999998757	102.0963043275208406	102.0963043275196611
22	1.00000000000000000	103.2310463739754169	103.2310463739754027
23	1.00000000000000622	104.3784004294587646	104.3784004294593473
24	0.99999999999998934	105.5385066692400926	105.5385066692388989
25	0.99999999999999523	106.7115068265565725	106.7115068265560751
26	0.99999999999999356	107.8975442099290092	107.8975442099282560
27	1.00000000000000044	109.0967637206708361	109.0967637206709355
28	1.00000000000000000	110.3093118705903493	110.3093118705903350
29	1.00000000000000000	111.5353367998912404	111.5353367998912262
30	1.00000000000000000	112.7749882952704183	112.7749882952704041
31	1.00000000000000000	114.0284178082186202	114.0284178082186060
32	1.00000000000000000	115.2957784735232991	115.2957784735232991
33	0.99999999999999989	116.5772251279772007	116.5772251279771865
34	1.00000000000000000	117.8729143292958952	117.8729143292958668
35	0.99999999999999989	119.1830043752440673	119.1830043752440531
36	1.00000000000000000	120.5076553229759071	120.5076553229758929
37	1.00000000000000000	121.8470290085889900	121.8470290085889758
38	1.00000000000000000	123.2012890668969476	123.2012890668969476
39	1.00000000000000000	124.5706009514204453	124.5706009514204169

Table A.8: verification of cut willow tree transition probabilities from Table A.6, with RMS errors 4.193×10^{-14} and 2.850×10^{-12}

i	j	p_{ij}	i	j	p_{ij}	i	j	p_{ij}
0	5	0.2680873075186	39	31	0.4410605205036	5	9	0.9360754899378
0	6	0.7319126924814	39	35	0.5589394794964	5	13	0.0639245100622
3	6	0.5449342011613	1	2	0.1160511302865	8	9	0.4122425618589
3	10	0.4550657988387	1	7	0.8839488697135	8	13	0.5877574381411
6	6	0.0079138470630	4	7	0.6968811390915	11	13	0.5027630537230
6	10	0.9920861529370	4	12	0.3031188609084	11	14	0.4972369462770
9	11	0.7343579237396	7	7	0.2726116121936	14	14	0.6032443082551
9	15	0.2656420762604	7	12	0.7273883878064	14	18	0.3967556917449
12	11	0.2037240376505	10	12	0.8066423140499	17	13	0.0354349738587
12	15	0.7962759623495	10	16	0.1933576859501	17	14	0.1260851953122
15	16	0.9215289411175	13	12	0.2784455141926	17	18	0.4466668527775
15	20	0.0784710588825	13	16	0.7215544858074	17	19	0.3918129780517
18	0	0.0000013725317	16	17	0.9711406058348	20	3	0.0000517328858
18	1	0.0000042602371	16	18	0.0288593941652	20	4	0.0001632838575
18	5	0.0005296685184	19	2	0.0000143875909	20	8	0.0085564905209
18	6	0.0014304325467	19	7	0.0024776968848	20	9	0.0151413877136
18	10	0.0405915329795	19	12	0.0394388883840	20	13	0.1538694656061
18	11	0.0370678116124	19	16	0.0235929919777	20	19	0.5173722986021
18	15	0.4115288174453	19	17	0.1807094855695	20	24	0.2295011436055
18	16	0.0175981980604	19	18	0.3828563609837	20	28	0.0408866725941
18	20	0.3134883201643	19	22	0.3542544324432	20	29	0.0329866938512
18	26	0.1576834124112	19	27	0.0148439755218	20	33	0.0010693808400
18	30	0.0123204215336	19	32	0.0018004597173	20	34	0.0003974932273
18	31	0.0075761008142	19	37	0.0000113209272	20	38	0.0000030065567
18	35	0.0001370330735	22	18	0.0490191557865	20	39	0.0000009501391
18	36	0.0000426180716	22	21	0.5450249002450	23	19	0.2698702761425
21	20	0.7215791963416	22	22	0.4059559439685	23	23	0.7047891379412
21	21	0.2233199428927	25	22	0.3434786578815	23	24	0.0253405859162
21	25	0.0094408008162	25	23	0.4809292135863	26	24	0.9784603991923
21	26	0.0456600599496	25	27	0.1755921285322	26	28	0.0215396008077
24	21	0.5883861450087	28	23	0.3821545047447	29	24	0.4560564888908
24	25	0.4116138549913	28	27	0.6178454952553	29	28	0.5439435111092
27	21	0.0528307840250	31	27	0.8804548294311	32	28	0.6770002290851
27	25	0.9471692159750	31	32	0.1195451705689	32	29	0.3229997709149
30	26	0.7772931191114	34	27	0.4623745352492	35	29	0.6465371819579
30	30	0.2227068808886	34	32	0.5376254647510	35	33	0.3534628180421
33	26	0.2481068038333	37	27	0.0301985610164	38	29	0.1129423976755
33	30	0.7518931961667	37	32	0.9698014389836	38	33	0.8870576023245
36	31	0.9639535334002	2	4	0.2189420050252			
36	35	0.0360464665998	2	8	0.7810579949748			

Table A.9: transition probabilities based the deal decomposition in Table 2.8 of data in Table 2.6

j	q_j^{k+1}	$\sum_{i=0}^{N_k} p_{ij}^k q_i$
0	$1.3762415383442527\times10^{-7}$	$1.3762415383442530\times10^{-7}$
1	$4.2717521364598702\times10^{-7}$	$4.2717521364544720\times10^{-7}$
2	$1.5931834591468337\times10^{-6}$	$1.5931834591476500\times10^{-6}$
3	$5.5203037061636400\times10^{-6}$	$5.5203037061636417\times10^{-6}$
4	$1.7770499436352966\times10^{-5}$	$1.7770499436370340\times10^{-5}$
5	$5.3146878983336490\times10^{-5}$	$5.3146878983336477\times10^{-5}$
6	$1.4767239072907256\times10^{-4}$	$1.4767239072907256\times10^{-4}$
7	$3.8121162364285731\times10^{-4}$	$3.8121162364289834\times10^{-4}$
8	$9.1428179330382538\times10^{-4}$	$9.1428179330382559\times10^{-4}$
9	$2.0372431824689824\times10^{-3}$	$2.0372431824689833\times10^{-3}$
10	$4.2175110377985561\times10^{-3}$	$4.2175110377985648\times10^{-3}$
11	$8.1118844084609610\times10^{-3}$	$8.1118844084609593\times10^{-3}$
12	$1.4495760140774562\times10^{-2}$	$1.4495760140774562\times10^{-2}$
13	$2.4066637570622128\times10^{-2}$	$2.4066637570622138\times10^{-2}$
14	$3.7123286177898583\times10^{-2}$	$3.7123286177898604\times10^{-2}$
15	$5.3202897899210060\times10^{-2}$	$5.3202897899210060\times10^{-2}$
16	$7.0840769157461098\times10^{-2}$	$7.0840769157461084\times10^{-2}$
17	$8.7637804962518273\times10^{-2}$	$8.7637804962518273\times10^{-2}$
18	$1.0073050519672236\times10^{-1}$	$1.0073050519672239\times10^{-1}$
19	$1.0757040298812089\times10^{-1}$	$1.0757040298812093\times10^{-1}$
20	$1.0673029724430215\times10^{-1}$	$1.0673029724430215\times10^{-1}$
21	$9.8389005261698345\times10^{-2}$	$9.8389005261698331\times10^{-2}$
22	$8.4269428014777534\times10^{-2}$	$8.4269428014777548\times10^{-2}$
23	$6.7059259942956792\times10^{-2}$	$6.7059259942956820\times10^{-2}$
24	$4.9580764500640070\times10^{-2}$	$4.9580764500640091\times10^{-2}$
25	$3.4059154590324332\times10^{-2}$	$3.4059154590324332\times10^{-2}$
26	$2.1738072448875188\times10^{-2}$	$2.1738072448875139\times10^{-2}$
27	$1.2890654752031090\times10^{-2}$	$1.2890654752031090\times10^{-2}$
28	$7.1022485482133977\times10^{-3}$	$7.1022485482134020\times10^{-3}$
29	$3.6356643696291348\times10^{-3}$	$3.6356643696291366\times10^{-3}$
30	$1.7291741282103616\times10^{-3}$	$1.7291741282103612\times10^{-3}$
31	$7.6411833510128596\times10^{-4}$	$7.6411833510128585\times10^{-4}$
32	$3.1372512390886277\times10^{-4}$	$3.1372512390886271\times10^{-4}$
33	$1.1967530119261564\times10^{-4}$	$1.1967530119261566\times10^{-4}$
34	$4.2415637600009743\times10^{-5}$	$4.2415637600009749\times10^{-5}$
35	$1.3967327945907335\times10^{-5}$	$1.3967327945907333\times10^{-5}$
36	$4.2733264543153883\times10^{-6}$	$4.2733264543153883\times10^{-6}$
37	$1.2147410514990309\times10^{-6}$	$1.2147410514982152\times10^{-6}$
38	$3.2082312520809555\times10^{-7}$	$3.2082312520809560\times10^{-7}$
39	$1.0138727724040564\times10^{-7}$	$1.0138727724040568\times10^{-7}$

Table A.10: verification of dealt willow tree transition probabilities from Table A.9, with RMS error 2.484×10^{-16}

i	$\sum_{j=0}^{N_{k+1}} p_{ij}^k$	S_i^k	$d_k \sum_{j=0}^{N_{k+1}} p_{ij}^k S_j^{k+1}$
0	1.00000000000000000	80.9475224897313552	80.9475224630850931
1	1.00000000000000000	81.8472079184100636	81.8472079184100636
2	1.00000000000000000	82.7568928362107101	82.7568928378686195
3	1.00000000000000888	83.6766883817372502	83.6766883817372360
4	0.99999999999999778	84.6067069288360187	84.6067069288319971
5	1.00000000000000000	85.5470621003240694	85.5470621003240836
6	1.00000000000000022	86.4978687818714320	86.4978687818714320
7	1.00000000000000000	87.4592431360363207	87.4592431360363207
8	0.99999999999999911	88.4313026164576286	88.4313026164576712
9	0.99999999999999978	89.4141659822042953	89.4141659822050627
10	1.00000000000000000	90.4079533122841639	90.4079533122841639
11	1.00000000000000000	91.4127860203148828	91.4127860203149112
12	1.00000000000000000	92.4287868693567418	92.4287868693565429
13	1.00000000000000000	93.4560799869116039	93.4560799869116039
14	0.99999999999999978	94.4947908800871517	94.4947908800871943
15	0.99999999999995215	95.5450464509311956	95.5450464509311956
16	1.00000000000000000	96.6069750119348782	96.6069750119348640
17	1.00000000000000000	97.6807063017095913	97.6807063017096198
18	1.00000000000000000	98.7663715008371952	98.7663715008372094
19	1.00000000000000000	99.8641032478964661	99.8641032478964945
20	1.00000000000000000	100.9740356556684731	100.9740356556684873
21	1.00000000000000000	102.0963043275208406	102.0963043275208406
22	1.00000000000000000	103.2310463739754169	103.2310463739754312
23	1.00000000000000000	104.3784004294587646	104.3784004294587646
24	1.00000000000000000	105.5385066692400926	105.5385066692399931
25	1.00000000000019784	106.7115068265565725	106.7115068265565725
26	1.00000000000000044	107.8975442099290092	107.8975442099290518
27	1.00000000000000000	109.0967637206708361	109.0967637206707934
28	1.00000000000000000	110.3093118705903493	110.3093118705903493
29	1.00000000000000000	111.5353367998912404	111.5353367998912688
30	1.00000000000000000	112.7749882952704183	112.7749882952704183
31	1.00000000000000000	114.0284178082186202	114.0284178082185491
32	1.00000000000000000	115.2957784735232991	115.2957784735233275
33	1.00000000000000000	116.5772251279772007	116.5772251279771865
34	0.99999999999999989	117.8729143292958952	117.8729143293039812
35	1.00000000000000000	119.1830043752440673	119.1830043752440957
36	1.00000000000000000	120.5076553229759071	120.5076553229758929
37	1.00000000000000000	121.8470290085889900	121.8470289970749576
38	1.00000000000000000	123.2012890668969476	123.2012890668970044
39	1.00000000000000000	124.5706009514204453	124.5706009514204453

Table A.11: verification of dealt willow tree transition probabilities from Table A.9, with RMS errors 4.101×10^{-15} and 3.910×10^{-11}

i	j	p_{ij}	i	j	p_{ij}	i	j	p_{ij}
0	5	0.2680872864179	39	31	0.4410605205036	5	9	0.9360754899375
0	6	0.7319127135821	39	35	0.5589394794964	5	13	0.0639245100625
3	6	0.5449342011607	1	2	0.1160511302863	8	9	0.4122425618588
3	10	0.4550657988393	1	7	0.8839488697137	8	13	0.5877574381412
6	6	0.0079138470634	4	7	0.6968811390915	11	3	0.0006890605254
6	10	0.9920861529366	4	12	0.3031188609085	11	4	0.0021748730791
9	0	0.0000684114514	7	7	0.2726116121933	11	8	0.1139688954881
9	1	0.0002123440949	7	12	0.7273883878067	11	9	0.2016769877384
9	5	0.0264004044841	10	2	0.0003707073444	11	13	0.3246561845349
9	6	0.0712974181093	10	7	0.0638397657463	11	18	0.3568339986341
9	10	0.4160405507956	10	12	0.6672227552608	14	14	0.6032443082551
9	15	0.4859808710646	10	16	0.2685667716484	14	18	0.3967556917449
12	11	0.2037240376505	13	12	0.2784455141926	17	13	0.0027052169418
12	15	0.7962759623495	13	16	0.7215544858074	17	14	0.1718321196268
15	16	0.9215289411175	16	17	0.9711406058348	17	18	0.4138373173807
15	20	0.0784710588825	16	18	0.0288593941651	17	19	0.4116253460507
18	10	0.0322445554819	19	12	0.0448499265114	20	13	0.1939500221725
18	11	0.0518011568427	19	16	0.0206740383260	20	19	0.5012046370135
18	15	0.4071081835466	19	17	0.1807094855695	20	24	0.2237711915639
18	16	0.0175981980604	19	18	0.3828563609837	20	28	0.0620878722748
18	20	0.2867552518299	19	22	0.3522818824432	20	29	0.0189862769753
18	25	0.0615282198795	19	27	0.0186283061662	23	19	0.2698702761425
18	26	0.1429644343590	22	18	0.0490191557865	23	23	0.7047891379412
21	20	0.7487601296143	22	21	0.5450249002450	23	24	0.0253405859162
21	21	0.1777367596226	22	22	0.4059559439685	26	24	0.9784603991923
21	25	0.0735031107631	25	22	0.3496015928669	26	28	0.0215396008077
24	21	0.5883861450087	25	23	0.4733345276754	29	24	0.6068569349164
24	25	0.4116138549913	25	27	0.1770638794577	29	29	0.3512944102466
27	21	0.3938080070442	28	23	0.4181466443005	29	33	0.0304264438745
27	26	0.4534996655472	28	27	0.5552010313621	29	34	0.0113096334986
27	30	0.0937048008321	28	32	0.0264857870564	29	38	0.0000855437327
27	31	0.0576211630373	28	37	0.0001665372811	29	39	0.0000270337312
27	35	0.0010422254489	31	27	0.8804548294311	32	24	0.1396726603411
27	36	0.0003241380903	31	32	0.1195451705689	32	29	0.8603273396589
30	26	0.7772931191114	34	27	0.4623745352475	35	29	0.6465371819579
30	30	0.2227068808886	34	32	0.5376254647525	35	33	0.3534628180421
33	26	0.2481068038333	37	27	0.0301985597643	38	29	0.1129423976756
33	30	0.7518931961667	37	32	0.9698014402357	38	33	0.8870576023244
36	31	0.9639535334002	2	4	0.2189420053507			
36	35	0.0360464665998	2	8	0.7810579946493			

Table A.12: transition probabilities obtained for the decomposition shown in Table 2.9

i	j	p_{ij}	i	j	p_{ij}	i	j	p_{ij}
0	0	0.0755095076461	14	17	0.6120833300766	26	22	0.4856028595979
0	1	0.3392021417267	14	18	0.0465206654784	26	26	0.5143971404021
0	9	0.5852883506271	15	11	0.0709833526290	27	22	0.0732484963663
1	0	0.2989907936017	15	12	0.2068260533370	27	25	0.9170054942419
1	9	0.7010092063983	15	18	0.7221905940340	27	26	0.0097460093919
2	2	0.2757042239581	16	9	0.0180446369870	28	25	0.9202922498293
2	9	0.7242957760419	16	10	0.0591573746209	28	31	0.0784930429140
3	1	0.0694448073520	16	11	0.0553960231917	28	32	0.0012147072567
3	2	0.0929378108318	16	18	0.8674019652004	29	25	0.7764665098569
3	9	0.8376173818162	17	15	0.4626980522235	29	30	0.1724505145066
4	3	0.0250842277356	17	20	0.5373019477765	29	31	0.0510829756365
4	4	0.2200469943325	18	15	0.1195347256408	30	26	0.8690759065582
4	10	0.7548687779320	18	16	0.5450328694994	30	32	0.0543957203648
5	2	0.0123000941037	18	23	0.2239637647064	30	33	0.0667809869305
5	3	0.0965225066806	18	24	0.1114686401534	30	34	0.0097473861465
5	10	0.8911773992157	19	17	0.3138987731315	31	26	0.7393955093766
6	5	0.1785324301459	19	19	0.5630740368955	31	32	0.2606044906234
6	11	0.8214675698541	19	24	0.0079774403748	32	28	0.9541069326646
7	4	0.0368815593133	19	25	0.1150497495983	32	34	0.0126225403270
7	5	0.0138583404979	20	14	0.1260322946804	32	35	0.0332705270084
7	11	0.9492601001887	20	15	0.0086818454642	33	28	0.8348311311089
8	7	0.2791080659337	20	20	0.5617513467915	33	34	0.1651688688911
8	13	0.7208919340663	20	22	0.3035345130640	34	29	0.9015183063797
9	5	0.0108621770784	21	16	0.1641702805859	34	36	0.0874847899932
9	6	0.0734063193569	21	17	0.1955712216303	34	37	0.0109969036271
9	7	0.0590042655803	21	22	0.1879851449832	35	29	0.7721647259103
9	13	0.8567272379844	21	23	0.4522733528007	35	35	0.2045464524677
10	8	0.0548780804956	22	19	0.5566970825386	35	36	0.0232888216220
10	9	0.1832236948902	22	24	0.4433029174614	36	30	0.8528166570667
10	14	0.7618982246142	23	21	0.8806972560298	36	37	0.0848034070667
11	7	0.0013050707465	23	28	0.0503590400583	36	38	0.0623799358666
11	8	0.0855966143874	23	29	0.0530097263170	37	30	0.7456351121866
11	14	0.9130983148661	23	30	0.0159339775949	37	37	0.2543648878134
12	13	0.0093584909293	24	21	0.7414174850386	38	30	0.7296833355532
12	14	0.9207141961805	24	27	0.1931888952619	38	39	0.2703166644468
12	17	0.0699273128902	24	28	0.0653936196995	39	30	0.6155812098775
13	13	0.5300214448612	25	21	0.0483151021388	39	38	0.3185184520534
13	17	0.4699785551388	25	22	0.6254976061003	39	39	0.0659003380691
14	12	0.0978519295966	25	26	0.2335307020067			
14	13	0.2435440748484	25	27	0.0926565897542			

Table A.13: transition probabilities for the crossover lattice based on Table 2.6

j	q_j^{k+1}	$\sum_{i=0}^{N_k} p_{ij}^k q_i$
0	$1.3762415383442527\times10^{-7}$	$1.3762415382921875\times10^{-7}$
1	$4.2717521364598702\times10^{-7}$	$4.2717521365351005\times10^{-7}$
2	$1.5931834591468337\times10^{-6}$	$1.5931834591690433\times10^{-6}$
3	$5.5203037061636400\times10^{-6}$	$5.5203037061565317\times10^{-6}$
4	$1.7770499436352966\times10^{-5}$	$1.7770499436350357\times10^{-5}$
5	$5.3146878983336490\times10^{-5}$	$5.3146878983341695\times10^{-5}$
6	$1.4767239072907256\times10^{-4}$	$1.4767239072907215\times10^{-4}$
7	$3.8121162364285731\times10^{-4}$	$3.8121162364285600\times10^{-4}$
8	$9.1428179330382538\times10^{-4}$	$9.1428179330383394\times10^{-4}$
9	$2.0372431824689824\times10^{-3}$	$2.0372431824691134\times10^{-3}$
10	$4.2175110377985561\times10^{-3}$	$4.2175110377984781\times10^{-3}$
11	$8.1118844084609610\times10^{-3}$	$8.1118844084610512\times10^{-3}$
12	$1.4495760140774562\times10^{-2}$	$1.4495760140774555\times10^{-2}$
13	$2.4066637570622128\times10^{-2}$	$2.4066637570622128\times10^{-2}$
14	$3.7123286177898583\times10^{-2}$	$3.7123286177898653\times10^{-2}$
15	$5.3202897899210060\times10^{-2}$	$5.3202897899210053\times10^{-2}$
16	$7.0840769157461098\times10^{-2}$	$7.0840769157461070\times10^{-2}$
17	$8.7637804962518273\times10^{-2}$	$8.7637804962518245\times10^{-2}$
18	$1.0073050519672236\times10^{-1}$	$1.0073050519672236\times10^{-1}$
19	$1.0757040298812089\times10^{-1}$	$1.0757040298812086\times10^{-1}$
20	$1.0673029724430215\times10^{-1}$	$1.0673029724430218\times10^{-1}$
21	$9.8389005261698345\times10^{-2}$	$9.8389005261698345\times10^{-2}$
22	$8.4269428014777534\times10^{-2}$	$8.4269428014777575\times10^{-2}$
23	$6.7059259942956792\times10^{-2}$	$6.7059259942956778\times10^{-2}$
24	$4.9580764500640070\times10^{-2}$	$4.9580764500640057\times10^{-2}$
25	$3.4059154590324332\times10^{-2}$	$3.4059154590324305\times10^{-2}$
26	$2.1738072448875188\times10^{-2}$	$2.1738072448875184\times10^{-2}$
27	$1.2890654752031090\times10^{-2}$	$1.2890654752031081\times10^{-2}$
28	$7.1022485482133977\times10^{-3}$	$7.1022485482133881\times10^{-3}$
29	$3.6356643696291348\times10^{-3}$	$3.6356643696291322\times10^{-3}$
30	$1.7291741282103616\times10^{-3}$	$1.7291741282103588\times10^{-3}$
31	$7.6411833510128596\times10^{-4}$	$7.6411833510128596\times10^{-4}$
32	$3.1372512390886277\times10^{-4}$	$3.1372512390886271\times10^{-4}$
33	$1.1967530119261564\times10^{-4}$	$1.1967530119261566\times10^{-4}$
34	$4.2415637600009743\times10^{-5}$	$4.2415637600009770\times10^{-5}$
35	$1.3967327945907335\times10^{-5}$	$1.3967327945907342\times10^{-5}$
36	$4.2733264543153883\times10^{-6}$	$4.2733264543153875\times10^{-6}$
37	$1.2147410514990309\times10^{-6}$	$1.2147410514990300\times10^{-6}$
38	$3.2082312520809555\times10^{-7}$	$3.2082312520809566\times10^{-7}$
39	$1.0138727724040564\times10^{-7}$	$1.0138727724040564\times10^{-7}$

Table A.14: crossover lattice transition probability verification based on Table A.13, with RMS error 6.432×10^{-16}

i	$\sum_{j=0}^{N_{k+1}} p_{ij}^k$	S_i^k	$d_k \sum_{j=0}^{N_{k+1}} p_{ij}^k S_j^{k+1}$
0	1.00000000000000000	80.9475224897313552	80.9475224866115894
1	1.00000000000000000	81.8472079184100636	81.8472079152553817
2	1.00000000000000000	82.7568928362107101	82.7568928382306268
3	1.00000000000000000	83.6766883817372502	83.6766883837796485
4	1.00000000000000000	84.6067069288360187	84.6067069287235256
5	1.00000000000000000	85.5470621003240694	85.5470621002102973
6	1.00000000000000000	86.4978687818714320	86.4978687818859271
7	0.99999999999999989	87.4592431360363207	87.4592431360509863
8	1.00000000000000000	88.4313026164576286	88.4313026164569891
9	1.00000000000000000	89.4141659822042953	89.4141659822036559
10	1.00000000000000000	90.4079533122841639	90.4079533122848460
11	1.00000000000000000	91.4127860203148828	91.4127860203155365
12	1.00000000000000000	92.4287868693567418	92.4287868693567276
13	1.00000000000000044	93.4560799869116039	93.4560799869114902
14	1.00000000000000000	94.4947908800871517	94.4947908800871375
15	0.99999999999999989	95.5450464509311956	95.5450464509311530
16	1.00000000000000022	96.6069750119348782	96.6069750119348782
17	1.00000000000000000	97.6807063017095913	97.6807063017096056
18	1.00000000000000000	98.7663715008371952	98.7663715008371526
19	1.00000000000000000	99.8641032478964661	99.8641032478964092
20	1.00000000000000000	100.9740356556684731	100.9740356556684873
21	1.00000000000000000	102.0963043275208406	102.0963043275207269
22	1.00000000000000000	103.2310463739754169	103.2310463739753601
23	0.99999999999999989	104.3784004294587646	104.3784004294586794
24	0.99999999999999967	105.5385066692400926	105.5385066692400642
25	1.00000000000000000	106.7115068265565725	106.7115068265565867
26	1.00000000000000089	107.8975442099290092	107.8975442099291229
27	1.00000000000000000	109.0967637206708361	109.0967637206708787
28	0.99999999999999989	110.3093118705903493	110.3093118705902924
29	1.00000000000000022	111.5353367998912404	111.5353367998911978
30	1.00000000000000000	112.7749882952704183	112.7749882952703757
31	1.00000000000000000	114.0284178082186202	114.0284178082185633
32	1.00000000000000000	115.2957784735232991	115.2957784735232707
33	1.00000000000000000	116.5772251279772007	116.5772251279771723
34	1.00000000000000000	117.8729143292958952	117.8729143292958526
35	0.99999999999999989	119.1830043752440673	119.1830043752440389
36	1.00000000000000000	120.5076553229759071	120.5076553229758645
37	1.00000000000000000	121.8470290085889900	121.8470290085889616
38	1.00000000000000000	123.2012890668969476	123.2012890668969476
39	1.00000000000000000	124.5706009514204453	124.5706009514204027

Table A.15: crossover lattice transition probability verification based on Table A.13, with RMS errors 1.415×10^{-17} and 1.463×10^{-11}

Bibliography

[AW01] Jonathan Alford and Nick Webber. Very high order lattice methods for one factor models. working paper, January 2001.

[AWDN04] Ari D. Andricopoulos, Martin Widdicks, Peter W. Duck, and David Newton. Curtailing the range for lattice and grid methods. *The Journal of Derivatives*, pages 55–61, Summer 2004.

[BG04] Mark Broadie and Paul Glasserman. A stochastic mesh method for pricing high dimensional American options. *Journal of Computational Finance*, 7(4):35–72, Summer 2004.

[CL95] H. Y. Cho and K. W. Lee. An extension of the three-jump process model for contingent claim valuation. *Journal of Derivatives*, 3:102–108, 1995.

[CRR79] J. Cox, S. Ross, and M. Rubinstein. Option pricing: a simplified approach. *Journal of Financial Economics*, 7:229–263, 1979.

[Cur01] Michael Curran. Willow power: optimizing derivative pricing trees. *Algo Research Quarterly*, 4(4), December 2001.

[DK94] E. Derman and I. Kani. Riding on a smile. *Risk Magazine*, 7(2), 1994.

[DKC96] E. Derman, I. Kani, and N. Chriss. Implied trinomial trees of the volatility smile. *Journal of Derivatives*, 3(4):7–22, 1996.

[Jäc00] Peter Jäckel. Non-recombining trees for the pricing of interest rate derivatives in the BGM/J framework. working paper, October 2000.

[JR82] R. Jarrow and A. Rudd. Approximate option valuation for arbitrary stochastic processes. *Journal of Financial Economics*, 10:347–369, 1982.

[LR96] Dietmar P. J. Leisen and Matthias Reimer. Binomial models for option valuation – examining and improving convergence. *Applied Mathematical Finance*, 3:319–346, 1996.

[MW02] L. A. McCarthy and N. J. Webber. Pricing three-factor models using icosahedral lattices. *The Journal of Computational Finance*, 5(1), 2002.

[PdDKUK83] R. Piessens, E. de Doncker-Kapenga, C.W. Uberhuber, and D.K. Kahaner. *QUADPACK: A subroutine package for automatic integration*. Springer Verlag, 1983.

[Rub94] Mark Rubinstein. Return to OZ. *Risk Magazine*, 7(11), 1994.

[Sch97] W. M. Schmidt. On a general class of one-factor models for the term structure of interest rates. *Finance and Stochastics*, 1:3–24, 1997.

[SW03] Mark Staley and Cameron Wicentowich. Spline trees: binomial lattices for multi-factor processes. working paper, December 2003.

[Tia93] Y. Tian. A modified lattice approach to option pricing. *J. Futures Markets*, 13:563–577, 1993.

[TL01] Yi Tang and Jeffrey Lange. A nonexploding bushy tree technique and its application to the multifactor interest rate market model. *The Journal of Computational Finance*, 4(2), Spring 2001.

[vD00] Stijn van Dongen. *Graph Clustering by Flow Simulation*. PhD thesis, University of Utrecht, May 2000.