Applied Artificial Intelligence, in press (2000), 19 pages Submitted 7/00; Accepted: 11/00

Efficient methods for computing investment strategies for multi-market commodity trading

Milos Hauskrecht
Computer and Information Sciences Department
Temple University
Wachman Hall 313
1805 North Broad Street
Philadelphia, PA 19122
milos@joda.cis.temple.edu

Luis Ortiz and **Ioannis Tsochantaridis** and **Eli Upfal**
Computer Science Department, Box 1910
Brown University
Providence, RI 02912
{*leo,it,eli*}*@cs.brown.edu*

Abstract

The focus of this work is the computation of efficient strategies for commodity trading in a multi-market environment. In today's "global economy" commodities are often bought in one location and then sold (right away, or after some storage period) in different markets. Thus, a trading decision in one location must be based on expectations about future price curves in all other relevant markets, and on current and future storage and transportation costs. Investors try to compute a strategy that maximizes expected return, usually with some limitations on assumed risk. With standard stochastic assumptions on commodity price fluctuations, computing an optimal strategy can be modeled as a Markov decision process (MDP). However, in general such a formulation does not lead to efficient algorithms. In this work we propose a model for representing the multi-market trading problem and show how to obtain efficient structured algorithms for computing optimal strategies for a number of commonly used trading objective functions (Expected NPV, Mean-Variance, and Value at Risk).

1. Introduction

Investment is the act of incurring immediate cost in the expectation of future reward. Investment options represent various tradeoffs between risk and expected profit. Investors try to maximize their expected return subject to the risk level that they are willing to assume. Modern economics theory models the uncertainty of future rewards as a stochastic process defining future price curves. The process is typically Markovian, thus investment decision can be modeled as a Markov decision process (MDP) [3, 16, 24] where a state of the underlying process needs only to include the current investment portfolio and current prices. While the MDP model gives a succinct formalization of the investment decision processes it does not necessarily imply efficient algorithms for computing optimal strategies. A challenging goal in this research area is to characterize special cases of the general investment paradigm that are interesting enough from the application point of view while simple enough to allow efficiently computable analytic solutions.

We focus in this paper on commodity trading. Past work has mainly dealt with single market trading problems (see [12, 15] and the references there), where commodity is bought, stored and eventually sold at the same location. Here we address a more realistic scenario in today's "global economy", that of a multi-site trading problem where a commodity can be bought in one location, stored at a second location and eventually sold at a third market. Prices at different locations may be different, and they may have different future price curves. Transportation costs also vary in time. While there can be large gaps in spot prices in different locations, future prices are more correlated - the future price of the commodity at site X cannot be larger than the price at site Y plus the cost of transportation between Y and X. Trading in a "global economy" is significantly more complex, since a local trading decision must be based on expectations about future price curves in all other relevant markets, as well as transportation and storage costs.

Modeling the multi-site commodity trading as a Markov decision process leads to a large state space, and a large action space. There has been extensive research in AI in recent years on solving MDPs with large state spaces exploiting, e.g., the structure of a specific problem through factoring and decompositions [20, 7, 11, 10, 22, 6] or various abstractions [9, 14]. However, all these works assume finite or at least discrete state space and the optimal solution or its close approximations still may not be efficiently computable. Nevertheless, we show in this work that under several commonly used utility functions the optimal trading strategy for the multi-site commodity problem still can be computed efficiently. In addition we show that the problem offers a specific structure which can be exploited to further reduce the computational complexity of a problem and speed-up the solution.

A standard assumption in mathematical economics is that commodity prices (e.g., oil and copper) are best modeled as a *mean reverting* stochastic process [12]. In our case, prices in all locations follow the mean reverting process but with different set of parameters for different sites. To solve the trading problem we first consider the *expected net present value (ENPV)* objective function, where the goal is to maximize expected gain with no consideration to risk. Under this objective function the optimization problem becomes myopic and can be computed by considering only current and next step prices. This allows us to design global optimal portfolio allocation algorithms that are polynomial in the number of sites in each trading step.

Building on the myopic property of the ENPV objective function we extend the result to two commonly used objective functions that combine ENPV maximization with limits on assumed risk at any one step. In the *Mean-Variance* function the goal is to maximize a weighted difference of the expected gain and the variance. The *Value at Risk* function maximizes expected gain subject to a (probabilistic) limit on the possibility of a large loss at any one step. Since both functions include a term that is linear in the variance of the process, the optimization problems in both cases lead to a constrained quadratic optimization problem. However, the computational complexities of the two problems are different. The mean-variance function has a particular structure that allows for polynomial time solution. The complexity of the optimization problem for the value at risk function varies, some special cases have polynomial time solutions.

A typical investor is risk-averse and tries to maximize the expected value of the investment while reducing the risk. Mean-Variance and Value at Risk are examples of objective functions with risk-aversion. We show that for a class of risk-averse functions a multi-site trading problem offers a special problem structure that allows us to decompose the optimization process to a sequence of problems of smaller complexity and thus to reduce the complexity of the original problem. Using this result, we develop and present efficient structure-based algorithms for both the Mean-Variance and Value at Risk models and show empirically significant running-time improvements for these algorithms compared to unstructured ones.

2. The Model

We consider investment problems with one type of commodity that is traded at n different sites. Once the commodidity is bought it can be either stored in each of the locations or transported between any two locations.

2.1 Price model

We assume that trading occurs at discrete time steps. To model commodity price fluctuations we adopt a discrete time version of the mean-reverting model [12]:

$$p^{(t+1)} = \mu - e^{-\eta}(\mu - p^{(t)}) + \epsilon^{(t)}, \tag{1}$$

where μ is the long term average price of the commodity i.e., a value to which the process reverts, η is the speed of reversion and $\epsilon^{(t)}$ is a sequence of independent random variables following normal distribution $N(0, \sigma_\epsilon)$.[1] Figure 1 shows an example of a disrete-time mean-reverting process.

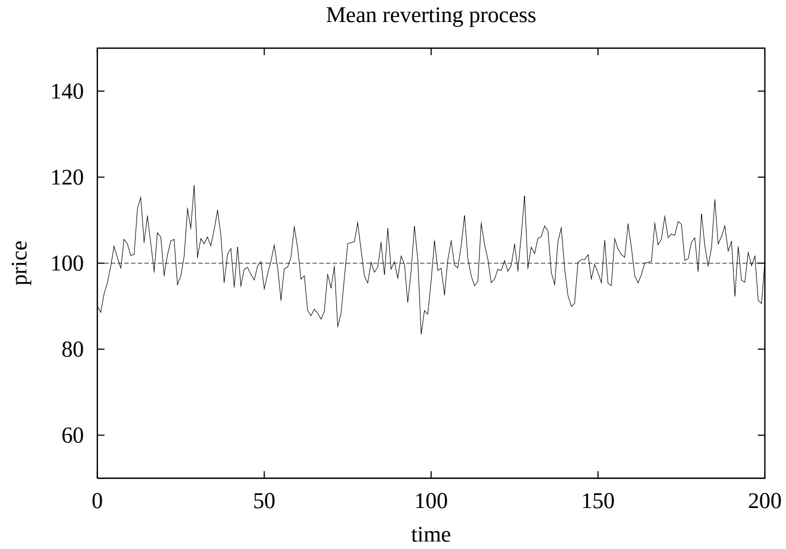

Figure 1: An example of a price trajectory for the mean reverting process with $\mu = 100, \eta = 0.6, \sigma = 7$.

Commodity prices at all locations follow mean reverting processes, each with different parameters and with possible correlations between their random components ϵ's. Their combined fluctuations are fully described by a multivariate normal distribution $N(\mathbf{0}, \Sigma)$, with a zero mean vector and a covariance matrix Σ. We assume that price movements are independent of our trading activities. We simplify the treatment of the problem by assuming that price spread (the difference between buy and sell prices at a given location) is negligible.[2]

There are natural capacity constraints on the number of commodity units we can transport (store) between the two locations at any time step. However, there are no constraints on buy and sell activities.

1. We note that normally distributed random components of the price process may lead to negative prices. One way to deal with this issue is to use a geometric version of the mean reverting process, where the logarithm of the price follows the mean reverting model. However, the behavior of such a model is quite different, and price curves of the standard model are more realistic.
2. In the more general setting (not considered here) prices can also fluctuate based on our demand and supply for the commodity or transportation service.

2.2 Valuation

Profit is measured by the standard *expected net present value (ENPV)* (see e.g. [8, 26]):

$$V^{\pi}(s) = E(\sum_{t=0}^{T} \gamma^t m^{(t)} | \pi, s) \tag{2}$$

where s denotes an initial state, π is the trading strategy, $\gamma = \frac{1}{1+r}$ is a discount factor, with r denoting the interest rate (present cost of money), T is the decision horizon, and $m^{(t)}$ is the cash flow at time t. We focus primarily on problems with infinite horizon ($T \to \infty$).

3. Markov decision process formulation of the problem

A *Markov decision process (MDP)* [3, 16, 24] describes a stochastic controlled process represented by a 4-tuple (S, A, T, R), where S is a set of process states; A is a set of actions; $T : S \times A \times S \to [0, 1]$ is a probabilistic transition model describing the dynamics of the modeled system; and $R : S \times A \times S \to \mathcal{R}$ models rewards assigned to transitions.

In the multi-site commodity trading problem the state of a process is determined by a price vector

$$\mathbf{p} = \{p_1, p_2, \cdots, p_i, \cdots, p_n, p_{11}, p_{12}, \cdots, p_{nn}\},$$

where the p_i's give the commodity price at location i, the $p_{i,j}$'s give the transportation price from i to j, and the $p_{i,i}$'s give the storage price at site i. Actions represent trading activities at a specific time step, and are defined as

$$\mathbf{a} = \{a_{11}, a_{12}, \cdots, a_{ij}, \cdots a_{nn}\},$$

where a_{ij} is the amount of commodity to be transported between i and j, or stored at location i if $j = i$. Thus, actions define allocations of commodity to different transportation (storage) edges. [3] Individual edge allocations in $\mathbf{a}$ are restricted to $C_{ij} \geq a_{ij} \geq 0$, where C_{ij} is the transportation limit from location i to j. The allocation weights in $\mathbf{a}$ must be non-negative since there is no meaning in our model to negative investment.[4] Also, weights a_{ij} can have real or integral values (depending on the problem definition). In the following we assume real-valued allocations, the extension to integral problems is straightforward.

The transition model is defined by a set of mean-reverting price functions (Equation 1), one for each location. For example, the price movements for location i is

$$p'_i = \mu_i - e^{-\eta_i}(\mu_i - p_i) + \epsilon_i,$$

where p_i and p'_i is the current and next step price, η_i and μ_i are the parameters of the mean-reverting process and ϵ_i is the random component. Similarly, we assume the mean-reverting model of transporation price dynamics.[5]

Rewards represent partial profits from applying the strategy and are modeled in terms of *step-wise gains*. The gain for transporting one unit of commodity from location i to j is defined by

$$g_{ij}(\mathbf{p}) = -p_i - p_{ij} + \gamma p'_j,$$

3. It is easy to see that the number of units to be transported between different locations is sufficient to define all trading activities. Simply, the number of units to buy and sell at different locations can be obtained by comparing the number of units currently held and the number of units to be transported from that location in the next step.
4. We note that in some of the problems in finance (e.g. portfolio optimization), constraints on weights can be lifted. This is the case when short-selling of an asset or security is possible. In that case, negative weights in the portfolio will reflect a short position.
5. The main results of the paper can be extended to other price models with Gaussian random component in a straightforward way.

where p_i is the current price of the commodity in location i, p_{ij} is the cost of transportation and p'_j is the price of the commodity in location j in the next step. The gain for an action $\mathbf{a}$ that allocates commodity to different transportation edges is the combination of partial gains

$$g(\mathbf{p},\mathbf{a}) = \sum_{i=1}^{n}\sum_{j=1}^{n} g_{ij}(\mathbf{p})a_{ij}.$$

Using our model, a sequence of cash flows for any strategy can be expressed in terms of step-wise gains (rewards) rather than actual money inflow and outflow. Intuitively, we can replicate payoffs from any strategy by buying the commodity at the beginning of a decision step and selling it at the end of that step. Therefore, the expected NPV model from Equation 2 for a strategy π can be expressed in terms of gains as

$$V^{\pi}(\mathbf{p}) \quad = \quad \lim_{T\to\infty} E(\sum_{t=0}^{T}\gamma^t g^{(t)}|\pi,\mathbf{p}), \tag{3}$$

where $g^{(t)}$ is the gain at time t. This is exactly the discounted, infinite-horizon criterion used commonly in MDPs [24]. Thus, our multi-site investment problem for expected NPV model can be expressed and solved as a Markov decision problem.

The optimal trading strategy for the discounted, infinite horizon Markov decision problem is stationary (see [3, 24]) and maps states of the process to actions. Therefore, the optimal strategy for our problem is $\pi^* : \mathbb{R}^n \times \mathbb{R}^{n^2} \to \mathbb{R}^{n^2}$, mapping the current commodity and transportation prices to amounts of units to be allocated to different transportation/storage edges.[6]

4. Solving the expected NPV problem

Using the MDP formulation, Equation 3 for the expected NPV model and a fixed policy π can be rewritten in Bellman's form [3] as

$$V^{\pi}(\mathbf{p}) = E(g(\mathbf{p},\pi(\mathbf{p}))) + \gamma \int_{-\infty}^{\infty}\cdots\int_{-\infty}^{\infty} V^{\pi}(\mathbf{p}')f(\mathbf{p}'|\mathbf{p})d\mathbf{p}', \tag{4}$$

where $E(g(\mathbf{p},\pi(\mathbf{p})))$ is the expected one-step gain for $\pi(\mathbf{p})$ and $f(\mathbf{p}'|\mathbf{p})$ is the conditional probability density function of the next step prices.

4.1 Myopic property

We see that $V^{\pi}(\mathbf{p})$ is hard to compute exactly. However, despite this difficulty the optimal strategy that maximizes ENPV can be computed efficiently. A key feature of our model is that prices change independently of our trading decisions (see Equation 4). Thus, the optimal policy is *myopic* (a greedy one-step policy is globally optimal) and can be easily computed (see [15]).

Theorem 1 *The optimal trading strategy for the expected NPV model is myopic.*

Proof The value of the optimal trading strategy is obtained from Equation 4 by maximizing over all possible actions

$$V^{*}(\mathbf{p}) = \max_{\mathbf{a}}\left[E(g(\mathbf{p},\mathbf{a})) + \gamma \int_{-\infty}^{\infty}\cdots\int_{-\infty}^{\infty} V^{*}(\mathbf{p}')f(\mathbf{p}'|\mathbf{p})d\mathbf{p}'\right].$$

6. For integral allocations the optimal policy is $\pi^* : \mathbb{R}^n \times \mathbb{R}^{n^2} \to \mathbb{N}^{n^2}$.

As the next step prices are independent of the action choice, the value can be rewritten as

$$V^*(\mathbf{p}) = \max_{\mathbf{a}} [E(g(\mathbf{p},\mathbf{a}))] + \gamma \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} V^*(\mathbf{p}') f(\mathbf{p}'|\mathbf{p}) d\mathbf{p}'.$$

We see that in order to get the optimal solution for $\mathbf{a}$ it is sufficient to optimize $\mathbf{a}$ only with regard to $E(g(\mathbf{p},\mathbf{a}))$. Thus the optimal strategy is myopic. □

The myopic property of the optimal investment strategy is critical for computing the solution for the commodity problem. The complete optimal investment strategy $\pi : R^n \times R^{n^2} \rightarrow R^{n^2}$ allocates the commodity units to different transportation edges for every price vector $\mathbf{p}$. As the number of possible prices and corresponding allocations is very large, it is not feasible to represent and store the optimal policy.

4.2 On-line algorithm

One way to avoid the computation of the complete policy is to compute individual price-specific allocations on-line. The on-line algorithm is invoked repeatedly in every step. In the general case, the on-line phase may be very time consuming as it may require to examine multiple price trajectories spanning multiple time steps. The myopic property of the decision process (Theorem 1) assures that we can obtain the optimal solution just by looking on what can happen in the next step. Simply, in order to decide the best allocation of investment for some price vector $\mathbf{p}$ it is sufficient to choose the allocation with the best one-step expected gain, and it is not necessary to consider more distant future and possible later price movements.

4.3 Optimal allocation

To find the optimal trading strategy for the expected NPV model it is sufficient to optimize expected one-step gains. Let $\mathbf{a}$ be some allocation of units to different transportation edges. The expected gain for $\mathbf{a}$ is

$$E(g(\mathbf{p},\mathbf{a})) = \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij} E(g_{ij}(\mathbf{p})).$$

To maximize the expectation we need to maximize the components of the sum. Assuming that C_{ij} is the constraint on the number of units we can transport between location i and j, the optimal allocation of a_{ij} is easy:

$$a^*_{ij} = \begin{cases} C_{ij} & \text{if } E(g_{ij}(\mathbf{p})) > 0; \\ 0 & \text{otherwise.} \end{cases}$$

Simply, we invest the limit to every edge with a positive expected gain.

5. Objective functions with one-step risk models

Once risk is taken into account, the above strategy of investing the limit on all edges with positive expected gains may not be optimal anymore.

Investment risk can be incorporated into the model in various ways. We focus here on objective functions that penalize or bound risk in any single step. In particular, we investigate:

- Mean-Variance model [21, 1, 5] that explicitly relates expected one-step gain and the gain variance;
- Value at Risk (VaR) model [19] which maximizes the expected present value of the investment, but at the same time limits possible step losses.

The important property of both models is that their value function is time-decomposable and can be expressed in the form similar to the expected NPV model

$$\begin{aligned} V^*(\mathbf{p}) &= && (5) \\ &= \max_{\mathbf{a}} \left[h(g(\mathbf{p},\mathbf{a})) + \gamma \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} V^*(\mathbf{p}') f(\mathbf{p}'|\mathbf{p}) d\mathbf{p}' \right] \\ &= \max_{\mathbf{a}} \left[h(g(\mathbf{p},\mathbf{a})) \right] + \gamma \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} V^*(\mathbf{p}') f(\mathbf{p}'|\mathbf{p}) d\mathbf{p}'. \end{aligned}$$

Here, $h(g(\mathbf{p},\mathbf{a}))$ is a function of a one-step gain (a random variable), not just its expectation. Different risk models use different forms of h. Note that the optimal policies must be myopic for this formalization.

5.1 Mean-Variance (MV) model

The mean-variance model [21, 1, 5] quantifies the risk in terms of the gain volatility. The model is additive and combines the expected one-step gain and the gain volatility into a single objective function $h(g(\mathbf{p},\mathbf{a}))$:

$$h(g(\mathbf{p},\mathbf{a})) = \alpha E(g(\mathbf{p},\mathbf{a})) - \beta Var(g(\mathbf{p},\mathbf{a})), \qquad (6)$$

where $\alpha, \beta \geq 0$. Intuitively the function reflects the fact that investors like the mean to be large but dislike the variance. Parameters α, β quantify this relation. We note that this valuation corresponds to the quadratic utility function [21].

Using the valuation function from equation 6, our goal is to find the allocation of commodity maximizing it. That is:

$$\pi^*(\mathbf{p}) = \arg\max_{\mathbf{a}} \left[\alpha E(g(\mathbf{p},\mathbf{a})) - \beta Var(g(\mathbf{p},\mathbf{a})) \right], \qquad (7)$$

subject to constraints $C_{ij} \geq a_{ij} \geq 0$ for all a_{ij}. The variance of the gain for $\mathbf{a}$ is:

$$Var(g(\mathbf{p},\mathbf{a})) = \mathbf{a}^T \mathbf{\Sigma}' \mathbf{a},$$

where $\mathbf{\Sigma}'$ is the gain covariance matrix (an $n^2 \times n^2$ matrix) obtained from the price covariance matrix $\mathbf{\Sigma}$ as:

$$\mathbf{\Sigma}'_{(ij)(kl)} = Cov(g_{ij}(\mathbf{p}), g_{kl}(\mathbf{p})) = \gamma^2 Cov(\epsilon_j, \epsilon_l) = \gamma^2 \mathbf{\Sigma}_{jl}.$$

5.1.1 Solution for the model

Equation 7 defines a quadratic optimization problem with linear constraints. The important property of this problem is that the h function has a unique global optimum solution. We can observe this from the fact that the Hessian of our function is a constant negative definite matrix (equal to $-2\beta\mathbf{\Sigma}'$).[7] Therefore, the function is concave. Figure 2 illustrates the shape of the function for the 2-dimensional case. This special case of Quadratic Programming is known to have a polynomial time solution [27]. Note, however, that by considering only integer allocation weights the problem becomes an integer programming problem and may not be solvable efficiently (see [23]).

5.2 Value at Risk (VaR) model

Let K be a loss threshold and δ the maximum probability of losing K or more units. The value of K is called the *value at risk* for δ (see [19]).

7. Recall that the covariance matrix Σ' is symmetric, positive definite.

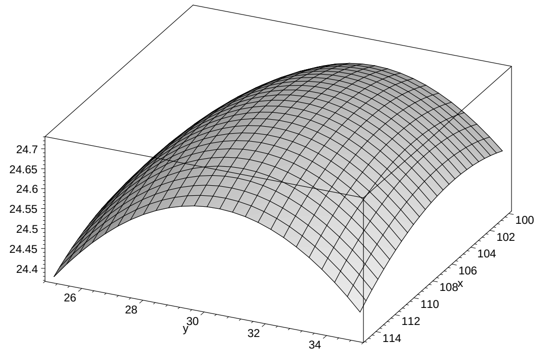

Figure 2: An example of a concave quadratic function for two dimensions.

This optimization problem has the form of Equation 5, where we maximize

$$h(g(\mathbf{p},\mathbf{a})) = E(g(\mathbf{p},\mathbf{a}))$$

subject to

$$C_{ij} \geq a_{ij} \geq 0 \quad \text{for all} \quad i,j = 1,\cdots,n;$$
$$P(g(\mathbf{p},\mathbf{a}) \leq -K) \leq \delta. \tag{8}$$

This is a linear optimization problem with linear and quadratic constraints. Inequality 8 reduces to a quadratic constraint by the properties of the normal distribution. Let x be a normally distributed random variable with mean μ and variance σ^2. Let k be a value such that $P(x \leq \mu - k\sigma) \leq \delta$ holds. The value of k measures the distance from the mean in terms of a standard deviation σ, such that values smaller than $\mu - k\sigma$ occur with probability less than δ. In the case of a normal distribution, k is only a function of δ, and it is independent of μ and σ. Therefore, in order to limit the losses of more than K units with probability $1-\delta$, we set the value of k_δ such that it satisfies $\mu - k_\delta\sigma \geq -K$.

Therefore, the constraint 8 can be rewritten as

$$[E(g(\mathbf{p},\mathbf{a})) + K]^2 - k_\delta^2 Var(g(\mathbf{p},\mathbf{a})) \geq 0 \tag{9}$$

which is quadratic in allocation weights $\mathbf{a}$. We can rewrite the constraint in terms of mean one-unit gains (vector $\boldsymbol{m}$) and covariances ($\boldsymbol{\Sigma'}$) as:

$$\mathbf{a}^T \left[\boldsymbol{m}\boldsymbol{m}^T - k_\delta^2 \boldsymbol{\Sigma'}\right] \mathbf{a} + 2K\boldsymbol{m}^T\mathbf{a} + K^2 \geq 0. \tag{10}$$

Let $W = \left[\boldsymbol{m}\boldsymbol{m}^T - k_\delta^2 \boldsymbol{\Sigma'}\right]$ be the $n^2 \times n^2$ matrix defining the quadratic term. We note that if the matrix W is negative definite, the problem corresponds to the linear optimization over the convex space. Thus, it can be solved efficiently in polynomial time [23]. However, when the matrix W is not negative definite we have a non-convex space over which we optimize. To solve this problem we can apply standard augmented Lagrangian techniques (see e.g. [4]).

5.3 Example

Figures 3, 4 and Table 1 illustrate and compare the performance of investment strategies for the three optimality criteria on a problem with five different trading sites. In this problem, both commodity and transporation/storage prices follow (partly correlated) mean reverting processes.[8]

8. We also assured that prices satisfy no-arbitrage conditions in means. That is, mean price of commodity at site X is never larger than the mean price at Y plus (mean) transportation cost between Y and X.

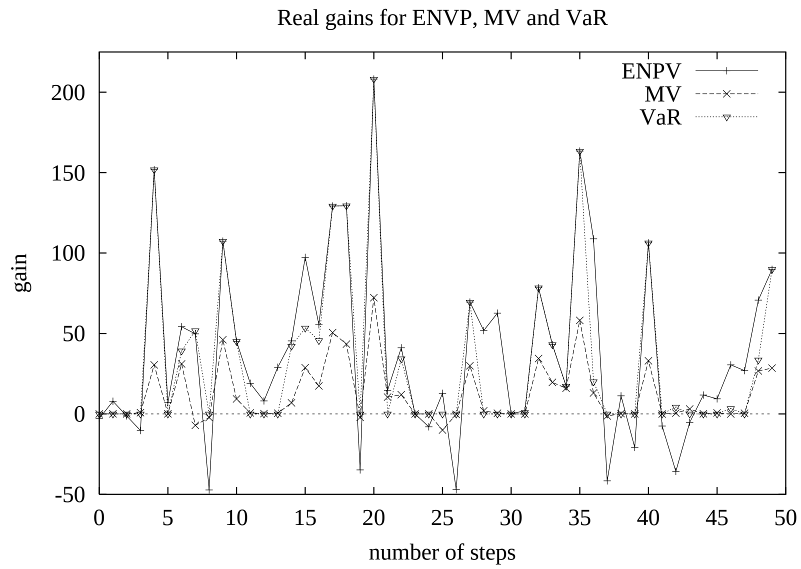

Figure 3: Comparison of three different optimization criteria: Expected NPV (ENPV), Mean-Variance (MV) and Value at Risk (VaR) on a problem with 5 trading sites and 50-step trajectory of prices' fluctuations, each following a mean-reverting process. For each step we plot the real gains for that step. The parameters of the MV model we use are $\alpha = 1$ and $\beta = 0.005$. We use $K = 0$ and $\delta = 0.0005$ for the VaR model.

	ENPV	MV	VaR
average real gains	38.08	12.08	33.34
standard deviation	56.87	18.61	49.65

Table 1: Average of the real gains and their standard deviation for ENPV, MV and VaR criteria and data from Figure 3.

Figure 3 shows the actual step-wise gains obtained for the three criteria using a fixed 50-step trajectory of prices' fluctuations; each price following a mean-reverting process. Table 1 summarizes the results in Figure 3 by showing real gain averages and their standard deviations. Finally, Figure 4 compares expectations of gains under different strategies.

We see that ENPV always leads to the maximum expected gain and it also achieves higher real gains on average compared to the other two criteria. However, step-wise gains for ENPV are also subject to higher fluctuations. Mean-Variance (MV) criterion yields gains that fluctuate less, but at the same time lead to considerable lower expected gains and also real gains on average. On the other hand, the VaR choices do not penalize a large variance when expectation is also high. Instead, it only tries to limit the probability of losses. Thus the real gains obtained for the VaR model vary more than those of the MV model and also tend to achieve higher gains (both under expectation and on average). From the graphs we observe that in many instances the allocations for the VaR criterion replicate exactly the ENPV choices. However, in some instances, when a chance of losses

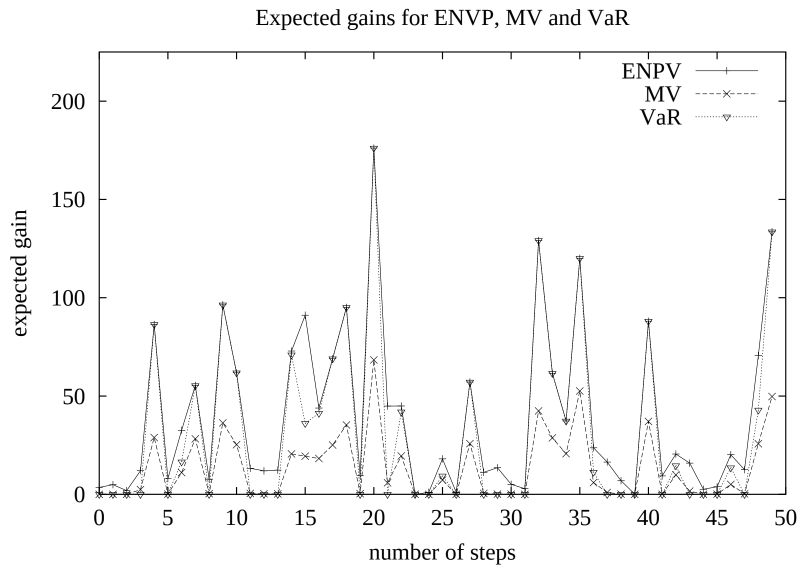

Figure 4: Comparison of expected gains for three different optimization criteria: Expected NPV (ENPV), Mean-Variance (MV) and Value at Risk (VaR) on a problem with 5 trading sites and 50 step long prices' trajectories.

exceeds the confidence threshold, the approach is more conservative and the allocation it chooses is different. For example, in 50 simulation steps in Figure 3 the VaR approach (with threshold gain 0) never lead to the negative gain, while there are eight different cases of negative gains for ENPV and two for the MV criterion.

Besides experiments comparing ENPV, MV and VaR criteria, we have tested the performance of the MV and VaR models for different combinations of parameters. Figures 5 and 6 illustrate the changes in real gains resulting from changes in these parameters. As expected, higher values of β lead to smaller average gains and smaller gain fluctuations. Simply, for higher values of β we penalize the variance more and thus we are likely to sacrifice the opportunity to capture higher gains. Similarly, for higher threshold prices and larger confidence factor δ, the VaR criterion becomes more conservative and leads to the reduction of both real and expected gains.

6. Exploiting the structure of a multi-site problem

Solving the optimization problem for the multi-site case requires to optimize all n^2 possible allocation weights. We show that this optimization can be carried out more efficiently by taking advantage of the problem structure and by solving a sequence of optimizations of smaller complexity.

The idea of our solution is to exploit the regularities of the covariance matrix $\mathbf{\Sigma'}$ of one-step gains for all transportation edges, in particular the fact that random components of transportation links leading to the same location are the same (fully correlated). Combining this property with certain objective criteria makes it possible to find the optimal allocation incrementally. The idea of the approach is based on the following theorem.

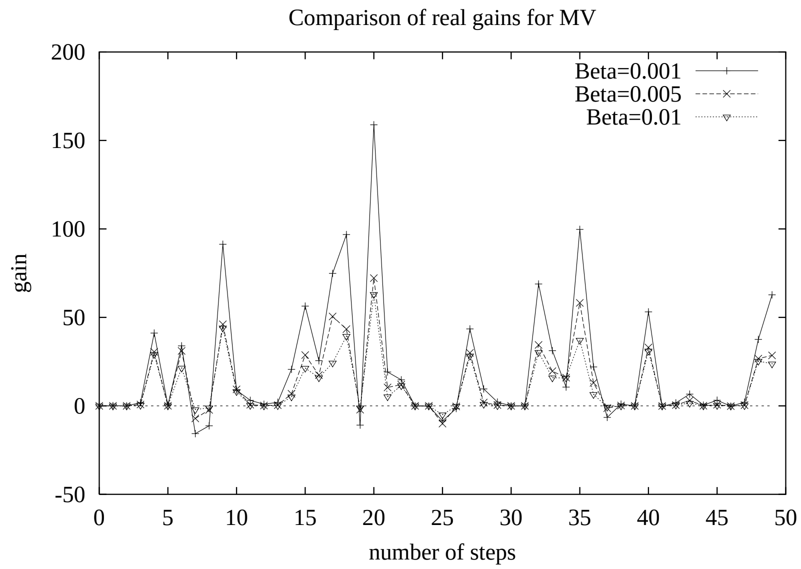

Figure 5: Comparison of performance of the Mean-Variance criterion for $\alpha = 1$ and three different settings of parameter β.

Theorem 2 *Let* $\mathbf{a}^*$ *be the optimal allocation of commodity maximizing expected gains (returns) and penalizing risk (volatility). Let,* (i, j) *and* (k, j) *be two different transportation links ending in the same target location* j *such that* $-p_k - p_{kj} < -p_i - p_{ij}$ *holds. Then* $a^*_{kj} > 0$ *only if* $a^*_{ij} = C_{ij}$*, otherwise* $a^*_{kj} = 0$.

Proof Gains from transporting one unit of commodity from i to j and k to j are

$$g_{ij}(\mathbf{p}) = -p_i - p_{ij} + \gamma\left[\mu_j - e^{\eta_j}(p_j - \mu_j) + \epsilon_j\right]$$

$$g_{kj}(\mathbf{p}) = -p_k - p_{kj} + \gamma\left[\mu_j - e^{\eta_j}(p_j - \mu_j) + \epsilon_j\right]$$

As the two gains share the same stochastic component and their difference is always deterministic

$$g_{ij}(\mathbf{p}) - g_{kj}(\mathbf{p}) = -p_i - p_{ij} - [-p_k - p_{kj}].$$

Moreover their covariance terms in Σ are the same. Thus, if $-p_k - p_{kj} < -p_i - p_{ij}$, there is no value in allocating the commodity to the transport link choice from k before we allocate the maximum, C_{ij}, to a_{ij}. Therefore if $a^*_{kj} > 0$, a^*_{ij} must be saturated ($a^*_{ij} = C_{ij}$). By similar argument, $a^*_{ij} < C_{ij}$ implies $a^*_{kj} = 0$. $\square$

The main consequence of the theorem is that we can perform the allocation of commodity to different transportation edges incrementally by allocating commodity to edges according to their expected gains, i.e., edges with higher expected gains for the same target location are allocated first. Then the problem of finding optimal allocations of n^2 weights can be decomposed into a sequence of optimization tasks with at most n variables.

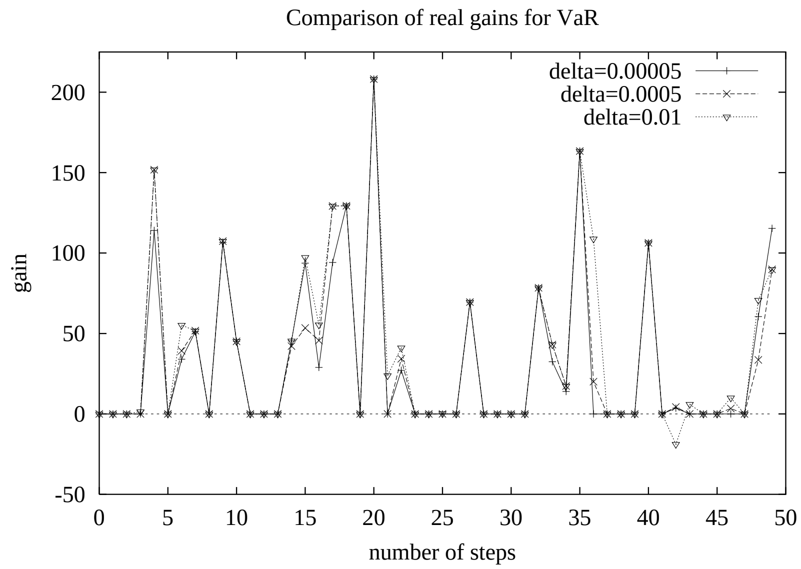

Figure 6: Comparison of performance of the Value at Risk criterion for three different settings of precision parameter δ. The threshold gain $K = 0$ is used for all three cases.

6.1 Structured algorithm

The structured algorithm builds the solution incrementally starting from edges with highest expected values and continuing with edges with smaller expectations afterwards. Let $\mathbf{a} = \{a_{11}, a_{12}, \cdots, a_{nn}\}$ be a vector of all previous allocations, reflecting, for each edge, the number of units of commodity already allocated to it. Initially this vector is set to $\bar{0}$. Let $\xi = \{\xi_1, \xi_2, \cdots, \xi_n\}$ be a vector of n indexes, such that ξ_j indexes the source location for the current best choice of a nonsaturated edge leading to location j (nonsaturated edge has not reached its full capacity). That is:

$$\xi_j = \arg \max_{i=1\cdots n} E(g_{ij}(\mathbf{p})) \quad \text{such that} \quad C_{ij} > a_{ij},$$

and where ties are broken arbitrarily. The edge (ξ_j, j) is called an *active edge* for location j. According to Theorem 2 an active edge must be allocated a full capacity before other nonsaturated edges incident to the same target location are used. To reflect this we always optimize the allocations over active edges (ξ) only.

Let $\tilde{\mathbf{a}} = \{\tilde{a}_{11}, \tilde{a}_{12}, \cdots, \tilde{a}_{nn}\}$ be a vector of (partial) allocations to all edges for the current optimization step and $\tilde{\mathbf{a}}^A = \{\tilde{a}_1^A, \tilde{a}_2^A, \cdots, \tilde{a}_n^A\}$ be a projection of $\tilde{\mathbf{a}}$ to the active edge set, that is, $\tilde{a}_j^A = \tilde{a}_{\xi_j j}$. By restricting our attention to active edges only, the optimization step becomes

$$\max_{\tilde{\mathbf{a}}^A} h(g(\mathbf{p}, \mathbf{a} + \tilde{\mathbf{a}}))$$

subject to

$$0 \leq \tilde{a}_{\xi_j j} \leq C_{\xi_j j} - a_{\xi_j j} \quad \text{for all} \quad j = 1, \cdots, n,$$

$$\tilde{a}_{ij} = 0 \quad \text{for all} \quad i, j = 1, \cdots, n \quad \text{such that} \quad i \neq \xi_j.$$

Structured-optimization (p, h)

initialize a (set all $a_{ij} = 0$);
choose a set of active edges ξ;
repeat
  set $\xi' = \xi$;
  solve $\tilde{\mathbf{a}}^{A,*} = \arg\max_{\tilde{\mathbf{a}}^A} h(g(\mathbf{p}, \mathbf{a} + \tilde{\mathbf{a}}))$
   subject to: $0 \leq \tilde{a}_{\xi_j j} \leq C_{\xi_j j} - a_{\xi_j j}$ for all $j = 1, \cdots, n$,
    $\tilde{a}_{ij} = 0$ for all $i, j = 1, \cdots, n$ such that $i \neq \xi_j$.
  for all $j = 1, \cdots, n$
    update $a_{\xi_j j} = a_{\xi_j j} + \tilde{a}_j^{A,*}$;
  update the set of active edges ξ;
 until $\xi' = \xi$ (no change in the active edge set)
return a;

Figure 7: Structured algorithm.

By repeating the restricted optimization, while dynamically changing the active edge set we can construct the optimal allocation for all n^2 edges gradually by solving a sequence of optimization tasks over n variables only (the size of the active edge set is at most n). Figure 7 shows a structured algorithm implementing this idea.

The algorithm starts by considering transportation choices with the highest expected gains, one for each target location. The restricted optimization procedure is then applied to the current set of active edges yielding partial allocations for these edges ($\tilde{\mathbf{a}}^{\mathbf{A},*}$). If the edge reaches its maximum capacity its allocation is fixed and a new edge for the same target location is considered as a choice (becomes active). This is repeated until the active edge set in the two consecutive cycles remains unchanged, signaling that either all edges have been exhausted or none of the edges were saturated in the last step (and no more improvements are possible).

The basic structured algorithm presented above can be further refined to fit better a specific objective function. In the following we apply the structure-based idea to solve the edge allocation problem for both the Mean-Variance and Value at Risk criteria and show the running-time improvements over the respective unstructured implementations.

6.2 Structured algorithm for the Mean-Variance model

The Mean-Variance criterion leads to a quadratic optimization problem over n^2 variables (see Section 5.1). By applying the structured approach the optimization can be decomposed into a sequence of quadratic optimizations with at most n variables. The reduced (partial) optimization task for the Mean-Variance model can be written as

$$\tilde{\mathbf{a}}^{A,*} = \arg\max_{\tilde{\mathbf{a}}^A} \left\{ \alpha \left[E(g(\mathbf{p}, \tilde{\mathbf{a}}^A)) \right] - \beta \left[(\mathbf{s}(\mathbf{a}) + \tilde{\mathbf{a}}^A)^T \tilde{\mathbf{\Sigma}} (\mathbf{s}(\mathbf{a}) + \tilde{\mathbf{a}}^A) \right] \right\} \tag{11}$$

subject to constraints:

$$0 \leq \tilde{a}_j^A \leq C_{\xi_j j} - a_{\xi_j j} \quad \text{for all} \quad j = 1, \cdots, n,$$

where $E(g(\mathbf{p}, \tilde{\mathbf{a}}^A)) = \sum_{j=1}^n E(g_{\xi_j j}(\mathbf{p}))\tilde{a}_j^A$ is the expected gain for the portfolio of active edges, and $\tilde{\mathbf{\Sigma}}$ is the *reduced gain covariance matrix*, an $n \times n$ matrix of the gain fluctuations for target locations ($\tilde{\mathbf{\Sigma}}_{kl} = \gamma^2 \mathbf{\Sigma}_{kl}$). The vector function $\mathbf{s}(\mathbf{a}) = \{s_1, s_2, \cdots, s_n\}$ yields target allocation counts for the

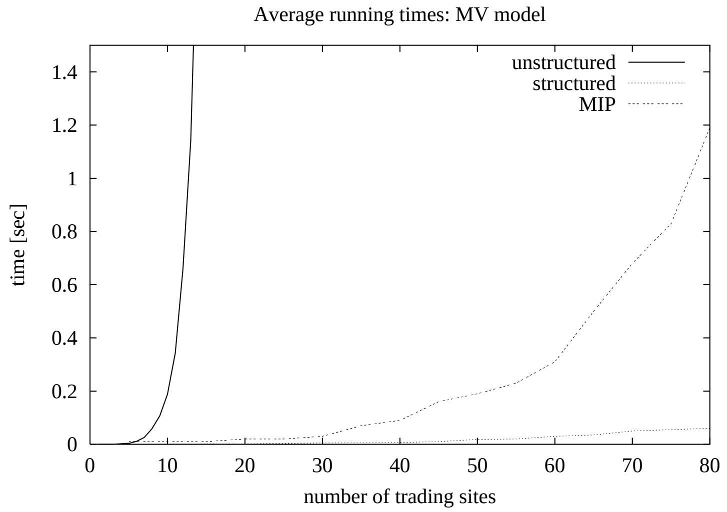

Figure 8: Average running times for unstructured, MIP-based and structured versions of the algorithm for the Mean-Variance criterion and varying number of trading sites.

current allocations $\mathbf{a}$. That is, for each target location, $s_j = \sum_{i=1}^{n} a_{ij}$ is the number of units of commodity already allocated to edges incident to that location. The vector $\mathbf{s}$ is easy to compute (update) after every optimization step and it simplifies the optimization task setup of the basic structured algorithm.

To illustrate the computational advantages of the structured algorithm over the unstructured one on multi-site commmodity trading problem we ran a set of experiments for varying number of trading sites. In addition, to the two approaches we have also tried and tested the mixed-integer programming (MIP) approach that takes some advantage of the underlying problem structure. The MIP approach is described in more detail later in Section 6.4. Figure 8 shows the results. We plot average running times, obtained for different market sizes. 100 different parameter settings (for fixed number of trading sites) were used to compute averages. To solve the underlying quadratic optimization problem we use ISML C/Math/Library implementation based on [13]. To implement the MIP method we rely on the IBM's OSL library [18]. We see that it takes about 1.5 seconds (on average) to compute optimal allocations for 14 trading sites with the unstructured method. In the same time we can comfortably solve problems with 80 sites using both the MIP and the structured approaches. The structured algorithm outperforms the MIP approach which is indicated by the widening difference in running times for larger number of sites. Solving problems with 80 sites with the MIP and the structured approach takes about 1.2 and 0.07 seconds respectively.

6.3 Structured algorithm for the VaR model

The optimization of VaR criterion can be performed more efficiently as well by using the structured optimization. To solve the problem, we optimize repeatedly the (reduced) problem with n variables:

$$\tilde{\mathbf{a}}^{A,*} = \arg\max_{\tilde{\mathbf{a}}^{A}} E(g(\mathbf{p}, \tilde{\mathbf{a}}^{A}))$$

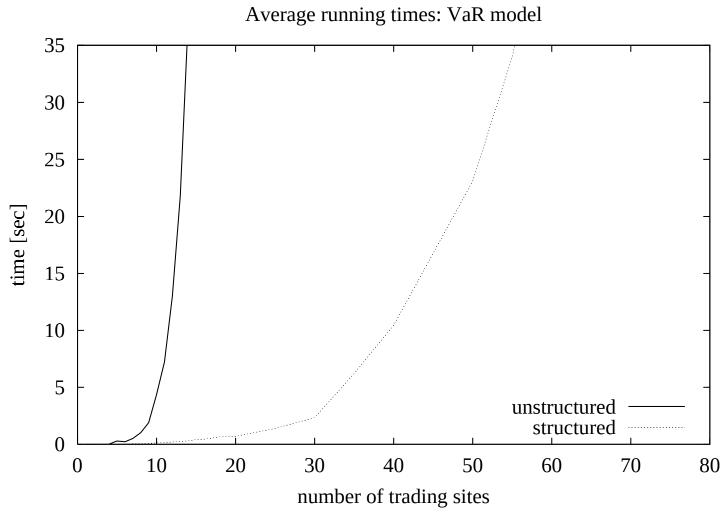

Figure 9: Average running times for unstructured and structured versions of algorithms for the Value at Risk criterion and varying number of trading sites.

subject to,

$$0 \le \tilde{a}_j^A \le C_{\xi_j j} - a_{\xi_j j} \quad \text{for all} \quad j = 1, \cdots, n;$$

$$\left[u(\mathbf{p}, \mathbf{a}) + E(g(\mathbf{p}, \tilde{\mathbf{a}}^A)) + K\right]^2 - k_\delta^2 \left[(\mathbf{s}(\mathbf{a}) + \tilde{\mathbf{a}}^A)^T \tilde{\mathbf{\Sigma}} (\mathbf{s}(\mathbf{a}) + \tilde{\mathbf{a}}^A)\right] \ge 0.$$

The notation used here is the same as in the Mean-Variance case. The only difference is that for the VaR criterion we use an additional quantity $u(\mathbf{p}, \mathbf{a}) = E(g(\mathbf{p}, \mathbf{a}))$ which summarizes the contribution of current allocations ($\mathbf{a}$) to the expected gain. This quantity is updated dynamically after every step and is needed to assure that the non-linear constraint is not violated during the optimization process. Figure 9 illustrates savings and speed-ups resulting from the structured version of the algorithm. We see that the unstructured algorithm can (on average) solve problems with 14 sites in about 35 seconds. [9] In contrast, the structured algorithm can solve problems with approximately 55 sites during the same time. To solve the VaR optimization problem we use ISML C/Math/Library functions based on [25].

6.4 Mixed-integer programming approach

The key to our structured solution is that allocations of commodity to different edges leading to the same target location j are fully correlated and that they can be ordered locally with respect to that target location (Theorem 2). The ordering is possible because the cost component of the gain (includes the cost of commodity and the cost of transportation) for target j decomposes to the sum of costs for different source locations, and because costs for all transportation edges are deterministic and linear functions of the amount of commodity allocated to them.

9. Note that similarly to the Mean-Variance case one can also apply the mixed-integer programming approach to the VaR case. However, we were not able to obtain any suitable library for this type of problem and thus we do not compare it to the other two algorithms.

This special problem structure can be also used to formulate a problem and solve it as a mixed integer program (MIP) [28] with special ordered set variables of type 2 [2].[10] The basic idea here is to represent the gain $g(\mathbf{p}, \mathbf{a})$ for the purpose of optimization as:

$$g(\mathbf{p}, \mathbf{a}) = \sum_{j=1}^{n} [E(g'(\mathbf{p}, a_j)) + a_j \epsilon_j]$$

such that

$$a_j = \sum_{i=1}^{n} a_{ij}$$

is a total number of commodity units transported to a location j, and $E(g'(\mathbf{p}, a_j))$ for a fixed $\mathbf{p}$ is a piecewise-linear function of a_j. Linear components of $E(g'((\mathbf{p}, a_j))$ correspond to indvidual source-target pairs and are ordered according to the values of $E(g_{ij}(\mathbf{p}))$, reflecting the result of Theorem 2. A region of a_j in which a specific linear function is active is determined by capacity constraints of source-target choices and their ordering. Let $\xi_j(k)$ denotes the k-th best choice of a source location for target j. The optimization problem for a risk averse investor is then:

$$\max_{\{a_1, a_2, \cdots, a_n\}} h(\sum_{j=1}^{n} [E(g'(\mathbf{p}, a_j)) + a_j \epsilon_j]),$$

such that,

$$a_j = \sum_{v=1}^{n} b_{vj} x_{vj}, \quad \text{for all} \quad j = 1, \cdots, n;$$

$$E(g'(\mathbf{p}, a_j)) = \sum_{v=1}^{n} l_{vj} x_{vj}, \quad \text{for all} \quad j = 1, \cdots, n;$$

where b_{vj} and l_{vj} are constants defining piecewise-linear function $E(g'(\mathbf{p}, a_j))$:

$$b_{vj} = \sum_{k=1}^{v} C_{\xi_j(k)j};$$

$$l_{vj} = \sum_{k=1}^{v} E(g_{\xi_j(k)j}(\mathbf{p})) C_{\xi_j(k)j};$$

and $x_{1j}, x_{2j}, \cdots, x_{nj}$ (for all $j = 1, \cdots, n$) are special ordered set of type 2 variables satisfying:

$$\sum_{v=0}^{n} x_{vj} = 1.$$

Note that because of the ordering, function $E(g'(\mathbf{p}, a_j))$ is always piecewise linear and concave.

Although, a mixed-integer program with special variables allows us to represent a great deal of problem structure, the formulation is not as specific as we would like to. In particular, it can take arbitrary piecewise-linear functions, no further specific structure is considered. However, in our case, functions $E(g'(\mathbf{p}, a_j))$ are always concave. Indeed, this additional property leads to the decomposition of a problem into a sequence of problems of smaller complexity and our structured solution presented in Section 6.1.

10. We would like to thank an anonymous reviewer for pointing out the MIP formulation of a problem to us.

7. Conclusion

We addressed the complex problem of finding optimal strategies for trading commodity in a multi-market environment. We investigated various objective criteria based on expected net present value (ENPV) and risk preferences of the investor. Different criteria can lead to optimization problems of different complexity. We showed that under the assumption of equal buy and sell prices, a number of criteria lead to the myopic portfolio optimization problem. This is very important as the computation of the optimal strategy needs to take into account only the current and next step prices and not all possible future price trajectories.

We analyzed and solved the problem for the expected NPV criterion and two commonly used risk-based criteria: Mean-Variance and Value at Risk models. We showed that in both risk-based models the optimization problem reduces to some form of the quadratic optimization problem. To further improve the efficiency of the solution we exploited the structure of the covariance matrix, in particular the fact that gains for the same target locations are fully correlated. This allowed us to reduce a large optimization problem for both risk-based criteria into a sequence of problems of smaller complexity. The empirical results obtained for the Mean-Variance and Value at Risk models support the feasibility of the solution and its practical applicability.

We note that our results and algorithms can be applied directly to any multi-site model in which the next-step price fluctuations are normally distributed, and thus not necessarily mean-reverting. The current model can be extended in a number ways. For example, interesting issues will arise if we refine the market models and extend them to include price spreads, trading (buy, sell) constraints, prices sensitive to supply and demands, etc. Another interesting direction is the investigation and application of more complex risk models (see e.g. [17]), reflecting different preferences of an investor and exploitation of our structure-based approach for solving them.

Acknowledgement

We wish to thank Oliver Frankel of Goldman Sachs for introducing us to this problem and for valuable technical discussions. We acknowledge the support of Goldman-Sachs throughout this project.

References

[1] Gordon J. Alexander and Jack C. Francis. *Portfolio Analysis.* Prentice Hall, 1986.

[2] E.M.L. Beale and J.A. Tomlin. Special facilities in a general mathematical programming system for non-convex problems using ordered sets of variables. In *Proceedings the Fifth IFORS Conference*, 1970.

[3] Richard E. Bellman. *Dynamic Programming.* Princeton University Press, Princeton, 1957.

[4] Dimitri P. Bertsekas. *Nonlinear programming.* Athena Scientific, Belmont, MA, 1995.

[5] Zvi Bodie, Alex Kane, and Alan J. Marcus. *Investments.* Richard D. Irwin, 1992.

[6] Craig Boutilier, Thomas Dean, and Steve Hanks. Decision-theoretic planning: Structural assumptions and computational leverage. *Artificial Intelligence*, 11:1–94, 1999.

[7] Craig Boutilier, Richard Dearden, and Moisés Goldszmidt. Exploiting structure in policy construction. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, pages 1104–1111, Montreal, 1995.

[8] Richard A. Brealey and Stewart C. Myers. *Principles of Corporate Finance.* McGraw-Hill, 1991.

[9] Thomas Dean and Robert Givan. Model minimization in Markov decision processes. In *Proceedings of the Fourteenth National Conference on Artificial Intelligence*, pages 106–111, Providence, 1997.

[10] Thomas Dean and Shieu-Hong Lin. Decomposition techniques for planning in stochastic domains. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, pages 1121–1127, Montreal, 1995.

[11] Richard Dearden and Craig Boutilier. Abstraction and approximate decision theoretic planning. *Artificial Intelligence*, 89:219–283, 1997.

[12] Avinash K. Dixit and Robert S. Pindyck. *Investment under Uncertainty.* Princeton University Press, Princeton, 1994.

[13] D. Goldfarb and A. Idnani. A numerical stable dual method for solving strictly convex quadratic programs. *Mathematical Programming*, 27:1–33, 1983.

[14] Milos Hauskrecht, Craig Boutilier, Nicolas Meuleau, Leslie Kaelbling, and Thomas Dean. Hierarchical solution of Markov decision processes using macro-actions. In *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*, pages 220–229, 1998.

[15] Milos Hauskrecht, Gopal Pandurangan, and Eli Upfal. Computing near optimal strategies for stochastic investment planning problems. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*, pages 1310–1315, Stockholm, 1999.

[16] Ronald A. Howard. *Dynamic Programming and Markov Processes.* MIT Press, Cambridge, 1960.

[17] Ronald A. Howard and James E. Matheson. Risk-sensitive Markov decision processes. *Management Science*, 18:883–896, 1972.

[18] M.S. Hung, W.O. Rom, and A.D. Waren. *Optimization Using IBM OSL.* Boyd and Fraser, 1994.

[19] Philippe Jorion. *Value at Risk: The New Benchmark for Controlling Market Risk.* Irwin Professional Pub, 1996.

[20] Harold J. Kushner and Ching-Hui Chen. Decomposition of systems governed by Markov chains. *IEEE Transactions on Automatic Control*, 19(5):501–507, 1974.

[21] Harry M. Markowitz. *Portfolio Selection.* Basil Blackwell, Cambridge, 1991.

[22] Nicolas Meuleau, Milos Hauskrecht, Kee-Eung Kim, Leonid Peshkin, Leslie Pack Kaelbling, Thomas Dean, and Craig Boutilier. Solving very large weakly coupled Markov decision processes. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence*, pages 165–172, Madison, 1998.

[23] Christos Papadimitriou and Kenneth Stieglitz. *Combinatorial Optimization: Algorithms and Complexity.* Dover, 1998.

[24] Martin L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming.* Wiley, New York, 1994.

[25] K. Schittkowski. NLPQL: A Fortran subroutine solving constrained nonlinear programming problems. *Annals of Operations Research*, 5:485–500, 1986.

[26] Lenos Trigeorgis. *Real Options*. MIT Press, Cambridge, 1996.

[27] Stephen A. Vavasis. *Nonlinear optimization: Complexity issues.* Oxford University Press, Oxford, 1991.

[28] Wayne L. Winston. *Introduction to Mathematical Programming*. Duxbury Press, Belmont, CA, 1995.