Duality Methods in Portfolio Allocation with Transaction Constraints and Uncertainty

Christopher W. Miller Department of Mathematics University of California, Berkeley

December 1, 2013

Abstract

We examine the application of duality methods to portfolio allocation with transaction constraints. We consider various models of asset returns and transaction constraints. In each model, we propose algorithms to efficiently find optimal solutions to the dual problem and examine when the solution to the dual problem is a minimizer of the primal problem. In particular, we consider mean-variance objective functions, both with correlation and without, as well as uncertainty in return and volatility parameters. We also consider transactions cost models with fixed costs that are different for long and short positions. Finally, we implement a general algorithm for locating optimal solutions to these dual portfolio allocation problem.

1 Problem Description

We consider a single-period investment model involving n assets. For each asset, we assume that the risk-return preference is given by a convex function $f_i : \mathbb{R} \to \mathbb{R}$. Our basic goal throughout this paper is to minimize the objective function $f : \mathbb{R}^n \to \mathbb{R}$ given by

$$f(w) = \sum_{i=1}^{n} f_i(w_i),$$

over the set $\mathcal{W} = \mathcal{W}_1 \times \cdots \times \mathcal{W}_n$, where every \mathcal{W}_i is a given interval $[\underline{w}_i, \overline{w}_i]$, where $\underline{w}_i < \overline{w}_i$. This corresponds to choosing a portfolio of uncorrelated assets which minimize the risk-return preference under constraints on sizes of both long and short positions.

We also assume that there are transaction costs. Given an initial position $w^0 \in \mathbb{R}^n$, the total transaction cost for taking a position w from the initial position w^0 is given by

$$g(w) = \sum_{i=1}^{n} g_i (w_i - w_i^0)$$

where, for every i = 1, ..., n, the functions $g_i : \mathbb{R} \to \mathbb{R}$ are given. We do not assume that these cost functions are convex or even continuous.

The crucial assumption which we will rely on in Sections 3 and 4 is that both f and g are decomposable functions. In addition, we will make the assumption that for any scalar $\lambda \ge 0$, and every $i = 1, \ldots, n$, finding a minimizer for the problem

$$\min_{\xi \in \mathcal{W}_i} f_i(\xi) + \lambda g_i(\xi)$$

can be efficiently done, algorithmically or otherwise.

Then in general, we consider an investment problem of the form

$$p^* = \min_{w \in \mathcal{W}} \left\{ f(w) : g(w) \le \tau \right\}$$

where $\tau \ge 0$ is a bound on the total amount of transaction costs. We henceforth refer to this as the primal problem.

2 Lagrange Relaxation for the Generic Problem

Rather than considering the primal problem above with inequality constraints, we will consider a dual problem where a Lagrange multiplier $\lambda \ge 0$ controls the cost of breaking the transaction bound. For more information on the relationship between primal and dual problems in convex optimization, see [1].

We consider an associated Lagrangian defined by

$$\mathcal{L}(w,\lambda) = f(w) + \lambda \left(g(w) - \tau\right).$$

Then we can define a function of λ as

$$D(\lambda) = \min_{w \in \mathcal{W}} \mathcal{L}(w, \lambda).$$

We henceforth refer to this minimization as the Lagrangian sub-problem. Note, this is a concave function of λ as a minimum over functions which are linear in λ . Then we can define an associated maximization problem as

$$d^* = \max_{\lambda \ge 0} D(\lambda).$$

We henceforth refer to this maximization as the dual problem.

2.1 General results about dual problem

We can make some immediate observations about the relationship between p^* and d^* without assuming a particular form for the objective functions or transaction costs.

Proposition 1. With the dual problem defined as above, $p^* \ge d^*$.

Proof. Let $\lambda \ge 0$. Then for any $w \in \mathcal{W}$ such that $g(w) \le \tau$, we have

$$f(w) \ge f(w) - \lambda (\tau - g(w)) = \mathcal{L}(w, \lambda).$$

Now, taking the minimum over $w \in W$ without the restriction on g(w) can only result in a smaller value, so

$$p^* = \min_{w \in \mathcal{W}} \left\{ f(w) : g(w) \leqslant \tau \right\} \ge \min_{w \in \mathcal{W}} \mathcal{L}(w, \lambda) = D(\lambda).$$

This inequality holds for all $\lambda \ge 0$, so

$$p^* \ge \max_{\lambda \ge 0} D(\lambda) = d^*.$$

Proposition 2. Let λ^* be a maximizer of $D(\lambda)$, and w^* be a corresponding minimizer of $\mathcal{L}(w,\lambda^*)$. Suppose that $g(w^*) = \tau$. Then $p^* = d^*$.

Proof. We already showed $p^* \ge d^*$, so it suffices to show that if $g(w^*) = \tau$ then $d^* \ge p^*$. But then

$$d^* = \mathcal{L}(w^*, \lambda^*) = f(w^*) - \lambda^* (\tau - g(w^*)) = f(w^*).$$

Furthermore, because $g(w^*) \leq \tau$, we conclude that

$$d^* = f(w^*) \ge \min_{w \in \mathcal{W}} \left\{ f(w) : g(w) \le \tau \right\} = p^*.$$

2.2 Algorithm to solve general dual problem

We now consider a general purpose algorithm which may be used to find sub-optimal solutions which approximate the optimal solution of the dual problem described above. We assume that we know an upper bound on the optimal λ^* , given by $\overline{\lambda}$.

Recall that $D(\lambda)$ is a concave function. The key to our algorithm is to note that we can easily compute super-gradients of $D(\lambda)$ and use these to locate a maximum.

Lemma 1. For any λ_* , let w_* be a minimizing allocation of $\mathcal{L}(w, \lambda_*)$. Then $g(w_*) - \tau$ is a super-gradient of $D(\lambda)$ at λ_* .

Proof. Fix λ_* and let w_* be a minimizing allocation of $\mathcal{L}(w, \lambda_*)$. Consider any other λ . Then

$$D(\lambda) \leq \mathcal{L}(w_*, \lambda) = -\lambda\tau + f(w_*) + \lambda g(w_*)$$

= $(g(w_*) - \tau) (\lambda - \lambda_*) - \lambda_*\tau + f(w_*) + \lambda_* g(w_*)$
= $(g(w_*) - \tau) (\lambda - \lambda_*) + D(\lambda_*).$

Then we conclude that $g(w_*) - \tau$ is a super-gradient of $D(\lambda)$ at λ_* .

Theorem 1. Suppose we are given an upper-bound $\overline{\lambda}$ on λ^* . Furthermore, suppose that we can find minimizing allocations of $\mathcal{L}(w,\lambda)$ for any λ in time O(f(n)), where f does not depend on λ . Then for any $\epsilon > 0$, we can construct an ϵ -suboptimal solution (λ_*, w_*) to the dual problem in time $O(-f(n)\log(\epsilon))$ via the algorithm shown in Algorithm 1.

Proof. Let λ^* be any maximizing value of $D(\lambda)$. By hypothesis, we have that $0 \leq \lambda^* \leq \overline{\lambda}$. We will reduce the size of this closed set by one-half in each step, while guaranteeing an inductive hypothesis that some λ^* is always contained inside.

For any λ , we will denote the super-gradient at λ determined in the previous lemma by $D'(\lambda) = g(w_{\lambda}) - \tau$. First, note that if $D'(\lambda) = 0$ then λ must be a maximizing value of $D(\lambda)$. However, the converse is not necessarily true. Furthermore, consider any two points x, y such that there is a maximizing point between them, $x < \lambda^* < y$. Then $D'(x) \ge 0$ and $D'(y) \le 0$.

Consider a step where we are examining the interval with $\lambda_l < \lambda_r$. We assume by the inductive hypothesis that there is a maximizing value of $D(\lambda)$ in the interval $[\lambda_l, \lambda_r]$. Then if

Input: Lagrangian function \mathcal{L} , a cost function g, a transaction limit τ , an error tolerance $\epsilon > 0$, and an upper bound on the maximizing point $\overline{\lambda}$.

Output: An ϵ -suboptimal solution (λ_*, w_*) to the dual problem.

```
\lambda_l \leftarrow 0
w_l \leftarrow \arg\min_{w \in \mathcal{W}} \mathcal{L}(w, \lambda_l)
m_l \leftarrow g(w_l) - \tau
\lambda_r \leftarrow \overline{\lambda}
w_r \leftarrow \arg\min_{w \in \mathcal{W}} \mathcal{L}(w, \lambda_r)
m_r \leftarrow g(w_r) - \tau
while \lambda_r - \lambda_l > \epsilon or ||w_r - w_l||_{\infty} > \epsilon do
       if m_l \leq 0 then
               (\lambda_r, w_r, m_r) \leftarrow (\lambda_l, w_l, m_l)
       else if m_r \ge 0 then
               (\lambda_l, w_l, m_l) \leftarrow (\lambda_r, w_r, m_r)
       else
               \lambda_m \leftarrow \frac{1}{2} \left( \lambda_l + \lambda_r \right)
              w_m \leftarrow \operatorname{arg\,min}_{w \in \mathcal{W}} \mathcal{L}(w, \lambda_m)
               m_m \leftarrow g(w_m) - \tau
               if m_m \leq 0 then
                     (\lambda_r, w_r, m_r) \leftarrow (\lambda_m, w_m, m_m)
               else
                      (\lambda_l, w_l, m_l) \leftarrow (\lambda_m, w_m, m_m)
end
(\lambda_*, w_*) \leftarrow (\lambda_l, w_l)
return (\lambda_*, w_*)
```



 $D'(\lambda_l) \leq 0$, it must be a maximizing value by the argument in the previous paragraph. Similarly, if $D'(\lambda_r) \geq 0$, it must be a maximizing value. This case is illustrated in Figure 1a.

Now, assume that $D'(\lambda_l) > 0$ and $D'(\lambda_r) < 0$. Let λ_m be the mid-point of λ_l and λ_r . If $D'(\lambda_m) \leq 0$, then the maximizing point in $[\lambda_l, \lambda_r]$ cannot be in $(\lambda_m, \lambda_r]$, so we conclude that there must be a maximizing point in $[\lambda_l, \lambda_m]$. Otherwise, we conclude that there must be a maximizing point in $[\lambda_m, \lambda_r]$. This case is illustrated in Figure 1b.

In either case, each step of this process reduces the size of the interval examined by one-half.



Figure 1: Illustrations of different super-gradient cases in the algorithm being described.

By compactness, there is a uniform $\delta > 0$, independent of λ , within which optimal allocations can differ by at most ϵ , so we reach an ϵ -suboptimal solution in $O(-\log(\epsilon))$ steps. Each of these requires one evaluation of a super-gradient, which takes O(f(n)) time by assumption, so this algorithm will return an ϵ -suboptimal solution in time $O(-f(n)\log(\epsilon))$.

3 Binary Model

In this section we consider a binary model for transaction costs where initial positions w^0 are all zero, each asset has a fixed transaction cost $b_i > 0$ for purchases, and short-sales are not possible. In particular, we consider the special case of

$$g_i(\xi) = \begin{cases} +\infty & \text{if } \xi < 0\\ 0 & \text{if } \xi = 0\\ b_i & \text{if } \xi > 0 \end{cases}$$

3.1 Homogeneous transaction costs

For simplicity, we first assume that all costs b_i are equal and, without loss of generality, equal to one. Furthermore, we let τ be an integer taken in $\{0, 1, \ldots, n\}$. In this case we can efficiently find a minimizer of the dual problem and prove that there is no duality gap.

Proposition 3. The solution to the dual problem under the binary model may be written

$$d^* = \max_{\lambda \ge 0} \left\{ -\tau\lambda + \sum_{i=1}^n \min\left(f_i^0, \lambda + f_i^+\right) \right\}$$

for appropriate vectors f^0 and f^+ .

Proof. If we consider the dual problem from the previous section, immediate simplifications are possible. In particular, we note that for any $\lambda \ge 0$ by decomposability,

$$D(\lambda) = \min_{w \in \mathcal{W}} \left\{ -\lambda \tau + \sum_{i=1}^{n} \left(f_i(w_i) + \lambda g_i(w_i) \right) \right\}$$
$$= -\lambda \tau + \sum_{i=1}^{n} \min_{w_i \in \mathcal{W}_i} \left\{ f_i(w_i) + \lambda g_i(w_i) \right\}.$$

But because $g_i(\xi)$ only takes on two values, we can write each minimization problem as

$$\min_{w_i \in \mathcal{W}_i} \left\{ f_i(w_i) + \lambda g_w(w_i) \right\} = \min\left(f_i(0), \lambda + \min_{0 < w_i \leq \overline{w}_i} f(w_i) \right).$$

For notational convenience, we define

$$\begin{cases} f_i^0 = f_i(0) \\ f_i^+ = \min \left\{ f_i(w_i) : 0 < w_i \leqslant \overline{w}_i \right\}, \end{cases}$$

as well as

$$w_i^* = \operatorname*{arg\,min}_{0 < w_i \leqslant \overline{w}_i} f_i(w_i).$$

Then we can write the dual problem as

$$d^* = \max_{\lambda \ge 0} \left\{ -\lambda \tau + \sum_{i=1}^n \min\left(f_i^0, \lambda + f_i^+\right) \right\}.$$

The usefulness of this result is that $D(\lambda)$ can be easily understood as the sum of a linear function and n kinked piecewise linear functions. This interpretation is useful is finding maximizing values λ^* of $D(\lambda)$.

Theorem 2. We can construct an optimal solution (λ^*, w^*) to the dual problem in the binary model in time O(n) via the algorithm shown in Algorithm 2.

Input: Objective functions f_i , maximal allocations \overline{w}_i , and trade cap τ . Output: An optimal solution (λ^*, w^*) to the dual problem. for $i \leftarrow 1$ to n do $\begin{vmatrix} f_i^0 \leftarrow f_i(0) \\ w_i^* \leftarrow \arg\min\{f_i(w_i) : 0 < w_i \leq \overline{w}_i\} \\ f_i^+ \leftarrow f_i(w_i^*) \\ \lambda_i \leftarrow f_i^0 - f_i^+ \end{vmatrix}$ end $\lambda^* \leftarrow (\tau + 1)$ th largest value of λ_i counting multiplicity $\lambda^* \leftarrow \max\{\lambda^*, 0\}$ for $i \leftarrow 1$ to n do $\begin{vmatrix} \mathbf{if} \ \lambda_i \leq \lambda^* \ \mathbf{then} \\ \ w_i^* \leftarrow 0 \\ \mathbf{end} \end{vmatrix}$ end return (λ^*, w^*)

Algorithm 2: BINARYPROBLEMSOLVER

Proof. Consider the function which we are maximizing

$$D(\lambda) = -\lambda\tau + \sum_{i=1}^{n} \min\left(f_i^0, \lambda + f_i^+\right)$$

Define $\lambda_i = f_i^0 - f_i^+$. Then for each asset, the graph of min $(f_i^0, \lambda + f_i^+)$ is piecewise linear with slope 1 for $\lambda < \lambda_i$ and slope 0 for $\lambda > \lambda_i$. As a sum of these, $D(\lambda)$ is a piecewise linear function with kinks at each λ_i . Furthermore, for large enough λ , $D(\lambda)$ has slope $-\tau$, and for small enough λ , $D(\lambda)$ has slope $n - \tau$. An illustration of $D(\lambda)$ with examples values is shown in Figure 2.

Now, consider a permutation σ such that

$$\lambda_{\sigma(1)} \ge \lambda_{\sigma(2)} \ge \cdots \ge \lambda_{\sigma(n)}.$$

Because the slope changes by one at each λ_i , the slope is zero on the interval $[\lambda_{\sigma(\tau+1)}, \lambda_{\sigma_{\tau}}]$. Furthermore, $D(\lambda)$ must be non-increasing above $\lambda_{\sigma(\tau+1)}$. Then either $\lambda_{\sigma(\tau+1)}$ is a maximizing value, if it is non-negative, or otherwise a maximizing value is at $\lambda^* = 0$.

Once a maximizing value λ^* is chosen, a corresponding optimal allocation can be created by buying all assets with $\lambda_i > \lambda^*$. This is a feasible allocation because there are not more than τ assets that satisfy this by our construction of λ^* .

The limiting step in this algorithm is identifying the $(\tau + 1)$ th largest value of λ_i . The key to the O(n) time is to recognize that we do not actually have to sort all of the λ_i values. The

Figure 2: Example of $D(\lambda)$ in the binary model with n = 3 and $\tau = 2$.



 $(\tau + 1)$ th largest value may be obtained in linear time via the QuickSelect algorithm[4].

Corollary 1. There is no duality gap in the binary model with homogeneous transaction costs and integer-valued τ .

Proof. First, assume there is no multiplicity in the λ_i in this problem. Then the algorithm above results in an optimal allocation w^* such that $g(w^*) = \tau$. Then by Proposition 2 in Section 2, there is no duality gap.

Now, assume that there is multiplicity in the λ_i . If there is no multiplicity on the τ th largest value, then there are still no problems. However, if there is multiplicity on the τ th largest value, then this algorithm returns an optimal allocation (w^*, λ^*) with $g(w^*) \leq \tau$. However, if $g(w^*) < \tau$ then there are $\tau - g(w^*)$ assets with $\lambda_i = \lambda^*$ which are not being purchased. At this value of λ , the investor is indifferent between purchasing them or not, so by purchasing them, there is a \tilde{w}^* such that $g(\tilde{w}^*) = \tau$ and $f(\tilde{w}^*) = f(w^*)$. Then $p^* = d^*$, even if this particular algorithm does not return an allocation with $g(w^*) = \tau$.

The homogeneous transaction cost binary model algorithm described above was implemented in MATLAB. In Figure 3, we show a graph of the average run-time versus number of assets. It seems that this algorithm successfully runs in approximately linear time with respect to the number of assets. This feature will hold neither in the models to come nor once we add constraints on net allocations of wealth.

3.2 Fully general binary model

To begin generalizing the binary model problem, we consider the case when $\tau \in [0, n]$ is not an integer. Intuitively, because all assets have a transaction cost of one, there is no way that the

Figure 3: Runtime of the homogeneous binary model algorithm compared to number of assets.



solution would be to purchase more than the floor of τ assets. Then we expect that the minimizer of the problem with the floor of τ will be a minimizer of the problem with non-integral τ .

Proposition 4. If $\tau \in [0, n]$ is not an integer, then if we modify the normal algorithm to find the $|\tau|$ th largest λ_i then it will output an optimal solution to the dual problem.

Proof. Let $\tau \in [0, n]$ not be an integer, so $[\tau] < \tau < [\tau]$. Then following the notation of the theorem about the algorithm for the homogeneous binary model, let σ be a permutation such that

$$\lambda_{\sigma(1)} \geq \cdots \geq \lambda_{\sigma(n)}.$$

Then $D(\lambda)$ will change sign at $\lambda_{\sigma([\tau])}$. So $\lambda^* = \lambda_{\sigma([\tau])}$. Then by construction, the algorithm will produce an allocation w^* with $g(w^*) \leq [\tau] < \tau$.

Unfortunately, as soon as we have non-integer transaction cost limits, there may be a duality gap. This is best illustrated by an example.

Proposition 5. When $\tau \in [0, n]$ is not an integer, it is not true that $p^* = d^*$ in general.

Proof. Consider a counterexample with only one asset. Let $f(\xi) = \frac{1}{2}\xi^2 - \xi$, b = 1, and $\tau = \frac{1}{2}$. Then it is immediately clear that $p^* = 0$.

On the other hand, we can easily check that

$$D(\lambda) = -\frac{1}{2}\lambda + \min\{0, \lambda - \frac{1}{2}\}$$

This function is piecewise linear and maximized at $\lambda = \frac{1}{2}$. Here it takes on a value of $d^* = -\frac{1}{4}$.

Therefore, we conclude that $p^* > d^*$.

This example illustrates that in the general case, we cannot expect strong duality to hold. There will be a potential duality gap, even in the binary model with non-integer transaction cost limits.

We next consider the case when not all b_i all equal, which we refer to as the inhomogeneous transaction cost binary model. The first thing to notice is not it is natural to ask that $0 \leq \tau \leq \sum_{i=1}^{n} b_i$. Furthermore, there is no reason to ask τ to be an integer at all.

In this case we can simplify $D(\lambda)$ to

$$D(\lambda) = -\tau\lambda + \sum_{i=1}^{n} \min\{f_i^0, b_i\lambda + f_i^+\}.$$

Furthermore, we can generalize the algorithm above. In particular, we now compute $\lambda_i = b_i^{-1} (f_i^0 - f_i^+)$ and find the smallest $\lambda^* \ge 0$ such that the sum of all b_i corresponding to $\lambda_i > \lambda^*$ is less than or equal to τ . This can be done in $O(n \log n)$ by sorting all the λ_i .

Once again, however, it is easy to see that as soon as b_i are not equal, there is no reason to expect $p^* = d^*$, even when τ is an integer. This is illustrated by an example.

Proposition 6. When b_i are not all equal, it is not true that $p^* = d^*$ in general.

Proof. Consider a counterexample with two assets. Let $f_1(\xi) = f_2(\xi) = \frac{1}{2}\xi^2 - 2\xi$, $b_1 = 1$, $b_2 = 2$, and $\tau = 2$.

First, we compute p^* directly. Note, buying neither asset or buying exactly one is feasible, but buying both is infeasible. If we buy an asset, it is optimal to purchase 2 units of it and achieve a payoff of -2. If we purchase neither, we get a payoff of 0. Then $p^* = -2$.

On the other hand, let us compute d^* and compare. It is easy to check that

$$D(\lambda) = -2\lambda + \min\{0, 2\lambda - 2\} + \min\{0, \lambda - 2\}.$$

This function is piecewise linear and maximized at $\lambda = 1$. Here it takes on a value of $d^* = -3$.

Therefore, we conclude that $p^* > d^*$.

These two examples illustrate that general models will always have a potential duality gap. In Section 4 we will consider the more general ternary model, and also provide a method for estimating the size of this duality gap.

4 Ternary Model

In this section we consider a ternary model for transaction costs where each asset has a fixed transaction cost $b_i > 0$ for purchases, and a fixed transaction cost $s_i > 0$ for short-sales. In particular, we consider the special case of

$$g_i(\xi) = \begin{cases} s_i & \text{if } \xi < 0\\ 0 & \text{if } \xi = 0\\ b_i & \text{if } \xi > 0. \end{cases}$$

Furthermore, let us assume that we have the restriction on τ that

$$0 \leqslant \tau \leqslant \sum_{i=1}^{n} \max\left(s_i, b_i\right)$$

4.1 General results in ternary model

Similar to the binary problem, we can simplify the dual problem. Let $\lambda \ge 0$, and note by decomposability that

$$D(\lambda) = \min_{w \in \mathcal{W}} \left\{ -\lambda \tau + \sum_{i=1}^{n} \left(f_i(w_i) + \lambda g_i(w_i - w_i^0) \right) \right\}.$$

As in the binary model, we can separate into separate regimes of buying, holding, and selling. However, in order to fully simplify this problem we require the following lemma about when minimums and maximums may be interchanged.

Lemma 2 (Minimax Lemma). Let K be a compact convex subset of a finite-dimensional vector space and C be a convex subset of a vector space. Let f be a real-valued function defined on $K \times C$ such that

1. $x \mapsto f(x, y)$ is convex and lower-semicontinuous for each y,

2. $y \mapsto f(x, y)$ is concave for each x.

Then

$$\inf_{x \in K} \sup_{y \in C} f(x, y) = \sup_{y \in C} \inf_{x \in K} f(x, y).$$

Proof. For a proof of this theorem, see [6].

Proposition 7. The solution of the dual problem under the ternary model may be written

$$d^* = 1^{\top} f^0 + \min_{u^{\pm}} \left\{ \left(h^+ \right)^{\top} u^+ + \left(h^- \right)^{\top} u^- : u^{\pm} \ge 0, \, u^+ + u^- \le 1, \, b^{\top} u^+ + s^{\top} u^- \le \tau \right\}$$

for appropriate vectors f^0 , h^+ , and h^- .

Proof. Recall that we can immediately write the dual problem in the ternary model as

$$D(\lambda) = \min_{w \in \mathcal{W}} \left\{ -\lambda \tau + \sum_{i=1}^{n} \left(f_i(w_i) + \lambda g_i(w_i - w_i^0) \right) \right\}$$
$$= -\lambda \tau + \sum_{i=1}^{n} \min_{w_i \in \mathcal{W}_i} \left\{ f_i(w_i) + \lambda g_i(w_i - w_i^0) \right\}.$$

But because $g_i(\xi)$ only takes on three values, we can write each minimization problem as

$$\min_{w_i \in \mathcal{W}_i} \left\{ f_i(w_i) + \lambda g_w(w_i - w_i^0) \right\} = \min\left(f_i(w_i^0), b_i \lambda + \min_{w_i^0 < w_i \leqslant \overline{w}_i} f(w_i), s_i \lambda + \min_{\underline{w}_i \leqslant w_i < w_i^0} f(w_i) \right).$$

For notational convenience, we define

$$\begin{cases} f_i^0 = f_i(w_i^0) \\ f_i^+ = \min \left\{ f_i(w_i) : w_i^0 < w_i \leqslant \overline{w}_i \right\} \\ f_i^- = \min \left\{ f_i(w_i) : \underline{w}_i \leqslant w_i < w_i^0 \right\}, \end{cases}$$

as well as

$$\begin{cases} w_i^+ = \arg \min_{w_i^0 < w_i \le \overline{w}_i} f_i(w_i) \\ w_i^- = \arg \min_{\underline{w}_i \le w_i < w_i^0} f_i(w_i). \end{cases}$$

Finally, if we let $h_i^+ = f_i^+ - f_i^0$ and $h_i^- = f_i^- - f_i^0$, then we can write the dual problem as

$$\begin{aligned} d^* &= 1^{\top} f^0 + \max_{\lambda \ge 0} \left\{ -\lambda \tau + \sum_{i=1}^n \min\left(0, h_i^+ + \lambda b_i, h_i^- + \lambda s_i\right) \right\} \\ &= 1^{\top} f^0 + \max_{\lambda \ge 0} \left\{ -\lambda \tau + \sum_{i=1}^n \min_{u_i^\pm} \left\{ \left(h_i^+ + \lambda b_i\right) u_i^+ + \left(h_i^- + \lambda s_i\right) u_i^- : u_i^\pm \ge 0, u_i^+ + u_i^- \le 1 \right\} \right\} \\ &= 1^{\top} f^0 + \max_{\lambda \ge 0} \min_{u^\pm} \left\{ \left(h^+\right)^{\top} u^+ + \left(h^-\right)^{\top} u^- + \lambda \left(b^{\top} u^+ + s^{\top} u^- - \tau\right) : u^\pm \ge 0, u^+ + u^- \le 1 \right\}. \end{aligned}$$

At this point, however, we have an affine objective function being minimized over a convex compact space in u^{\pm} and maximized over a convex space in λ . Then we can apply the Minimax Lemma to interchange the min and max, and obtain

$$d^{*} = 1^{\top} f^{0} + \min_{u^{\pm}} \left\{ \left(h^{+}\right)^{\top} u^{+} + \left(h^{-}\right)^{\top} u^{-} + \max_{\lambda \ge 0} \left\{\lambda \left(b^{\top} u^{+} + s^{\top} u^{-} - \tau\right)\right\} : u^{\pm} \ge 0, \ u^{+} + u^{-} \le 1 \right\}$$
$$= 1^{\top} f^{0} + \min_{u^{\pm}} \left\{ \left(h^{+}\right)^{\top} u^{+} + \left(h^{-}\right)^{\top} u^{-} : u^{\pm} \ge 0, \ u^{+} + u^{-} \le 1, \ b^{\top} u^{+} + s^{\top} u^{-} \le \tau \right\},$$

where the extra constraint follows because the maximization over λ is $+\infty$ when $b^{\top}u^{+}+s^{\top}u^{-} > \tau$ and zero otherwise.

At this point, it is worth noting that we have transformed a general convex optimization problem into a linear programming problem. Then we immediately have major results and algorithms for this problem.

Theorem 3. We can construct an optimal solution (λ^*, w^*) to the dual problem in the ternary model in polynomial time via the algorithm shown in Algorithm 3.

Proof. Because the objective function is decomposable into quadratics, the computation of all w_i^+ and w_i^- can be done in linear time. The limiting step is obtaining an optimal solution u^{\pm} to the equivalent problem from Proposition 7. This can be done in average-case polynomial time with simplex methods, or even in polynomial worst-case time with ellipsoid methods[3].

Because the value of u_i^{\pm} corresponds to the minimum of

$$\min\left(0, h_i^+ + \lambda b_i, h_i^- + \lambda s_i\right)$$

being attained in the holding, buying, or selling regime, once we have obtained an optimal u^{\pm} from the equivalent linear programming problem, we can convert back to an optimal portfolio by holding at w_i^0 if $u^+ = u^- = 0$, buying the optimal amount w_i^+ if $u^+ = 1$, or selling the optimal amount w_i^- if $u^- = 1$. Note, the linear programming problem may have fractional allocations, which correspond to assets which we are indifferent towards buying, selling, or holding. We choose to hold these in the algorithm in order to guarantee that $g(w^*) \leq \tau$.

Finally, in order to obtain λ^* , we need to solve the implicit equation

$$d^* = \mathcal{L}(w^*, \lambda^*).$$

Input: Objective functions f_i , initial allocations w_i^0 , allocation limits \underline{w}_i and \overline{w}_i , trading fees b_i and s_i , and the trading fee limit τ .

Output: An allocation w^* corresponding to an optimal solution (λ^*, w^*) to the dual problem.

```
for i \leftarrow 1 to n do

\begin{vmatrix} w_i^+ \leftarrow \arg\min\{f_i(w_i) : w_i^0 < w_i \le \overline{w}_i\} \\ w_i^- \leftarrow \arg\min\{f_i(w_i) : \underline{w}_i \le w_i < w_i^0\} \\ h_i^+ \leftarrow f_i(w_i^+) - f_i(w_i^0) \\ h_i^- \leftarrow f_i(w_i^+) - f_i(w_i^0) \\ end \\ (d^*, u^{\pm}) \leftarrow \text{optimal solution to linear programming problem} \\ for <math>i \leftarrow 1 to n do

\begin{vmatrix} w_i^* \leftarrow 0 \\ \text{if } u_i^+ = 1 \text{ then} \\ | w_i^* \leftarrow w_i^+ \\ end \\ \text{if } u_i^- = 1 \text{ then} \\ | w_i^* \leftarrow w_i^- \\ end \\ \lambda^* \leftarrow \text{solve } \mathcal{L}(w^*, \lambda) = d^* \\ return (\lambda^*, w^*) \\ \end{vmatrix}
Algorithm 3: TERNARYPROBLEMSOLVER
```

But for fixed w^* , this is linear in λ , so this can be done in linear time. Therefore, we obtain an optimal solution (λ^*, w^*) to the dual problem overall in polynomial time.

The ternary model algorithm described above was implemented in MATLAB. In particular, we use the MATLAB linear programming routines with an active-set algorithm and sparse matrices. In Figure 4, we show a graph of the average run-time versus number of assets. The major difference between this and the general Algorithm 1 is that the latter much iteratively determine λ^* . Therefore, we expect this algorithm will be much faster when transaction cost constraints bind (e.g. $\lambda^* > 0$).

4.2 Remarks and special cases

We next consider some special cases and remarks concerning the ternary model. In particular, we focus on when we can simplify the solution to the linear programming problem and potentially speed up the algorithm, and when we can determine information about the existence and size of the duality gap.

Consider the case when $b_i = s_i$ for all assets *i*. It is clear that we can still have a duality gap in this case, by extending the counterexample in Proposition 6 to the Ternary case. However, we can speed up the ternary algorithm above substantially.

Corollary 2. If $b_i = s_i$ for all assets *i*, then the solution of the dual problem under the ternary

Figure 4: Runtime of the ternary model algorithm compared to number of assets.



model may be written

$$d^* = 1^\top f^0 + \min_u \left\{ h^\top u : 0 \leqslant u \leqslant 1, b^\top u \leqslant \tau \right\}$$

for appropriate vectors f^0 and h. In particular, we can still construct an optimal solution (λ^*, w^*) to the dual problem in polynomial time.

Proof. We reproduce the proof for the general case from above, but with additional simplifications because $b_i = s_i$ for all assets. Where details of the argument are omitted, refer to the previous argument.

Because $g_i(\xi)$ only takes on two values now, we can write each separated minimization problem as

$$\min_{w_i \in \mathcal{W}_i} \left\{ f_i(w_i) + \lambda g_w(w_i - w_i^0) \right\} = \min\left(f_i(w_i^0), b_i \lambda + \min_{w_i \in \mathcal{W}_i} f(w_i) \right).$$

Then we define

$$\begin{cases} f_i^0 = f_i(w_i^0) \\ f_i = \min \{f_i(w_i) : w_i \in \mathcal{W}_i\} \end{cases}$$

as well as

$$w_i = \operatorname*{arg\,min}_{w_i \in \mathcal{W}_i} f_i(w_i).$$

If let $h_i = f_i - f_i^0$, then we can write the dual problem as

$$d^* = 1^{\top} f^0 + \max_{\lambda \ge 0} \left\{ -\lambda \tau + \sum_{i=1}^n \min(0, h_i + \lambda b_i) \right\}$$
$$= 1^{\top} f^0 + \max_{\lambda \ge 0} \min_u \left\{ h^{\top} u + \lambda \left(b^{\top} u - \tau \right) : 0 \le u \le 1 \right\}.$$

Applying the Minimax Lemma as before to switch the minimum and maximum, we obtain

$$d^* = 1^{\top} f^0 + \min_u \left\{ h^{\top} u + \max_{\lambda \ge 0} \left\{ \lambda \left(b^{\top} u - \tau \right) \right\} : 0 \le u \le 1 \right\}$$
$$= 1^{\top} f^0 + \min_u \left\{ h^{\top} u : 0 \le u \le 1, \ b^{\top} u \le \tau \right\},$$

where the extra constraint follows because the maximization over λ is $+\infty$ when $b^{\top}u > \tau$ and zero otherwise. As before, we can write an algorithm to solve this linear programming problem and transform the solution u^* into an optimal solution (λ^*, w^*) to the dual problem in polynomial time.

Note, the overall complexity of the algorithm is the same as that when $b_i \neq s_i$ because it relies on the complexity of the linear programming problem. However, the importance of this result is that we can increase the speed by a constant factor across the board by utilizing this symmetry in the problem.

Because we have shown that there in general may be a duality gap in the ternary model, we would like to be able to estimate the size of this gap. We show below how we can get an upper bound on the duality gap once we have solved the dual problem.

Proposition 8. Let (λ^*, w^*) be an optimal solution to the dual problem obtained from the algorithm above. Then the duality gap is bounded by

$$0 \leq p^* - d^* \leq \lambda^* \left(\tau - g(w^*)\right).$$

Proof. First, recall from the proof of the algorithm's validity that we will always have $g(w^*) \leq \tau$ for the output. Then any allocation is a feasible allocation of the original problem. Therefore, we have the following inequality

$$d^* = f(w^*) - \lambda^* (\tau - g(w^*)) \ge p^* - \lambda^* (\tau - g(w^*)).$$

If we rearrange this inequality and also recall from Section 2 that $p^* \ge d^*$ in general, then we obtain the bound

$$0 \leq p^* - d^* \leq \lambda^* \left(\tau - g(w^*) \right).$$

While this result allows estimation of the duality gap, it is important to keep in mind that we can not control the optimal value of λ^* in the dual problem a priori. Therefore, this result cannot be used to obtain any sort of convergence of the optimal solution to the dual problem to the optimal solution of the primal problem.

5 Taking into account uncertainty and correlations

Our model so far takes into account neither any correlation between the assets nor any uncertainty in the model itself. In this section we generalize to an objective function

$$f(w,\hat{r}) = \frac{1}{2}w^{\top}\Sigma w - \hat{r}^{\top}w,$$

where $\hat{r} \in \mathcal{R} \subset \mathbb{R}^n$ contains predicted returns and Σ is an estimate of the covariance matrix. For simplicity, we assume that $\mathcal{R} = \mathcal{R}_1 \times \cdots \times \mathcal{R}_n$, where \mathcal{R}_i is a given interval $[\underline{r}_i, \overline{r}_i]$, where $\underline{r}_i < \overline{r}_i$. We make the same assumptions of the cost function g as in the ternary model. Then we are considering an optimization problem of the form

$$p^* = \min_{w \in \mathcal{W}} \max_{\hat{r} \in \mathcal{R}} \left\{ f(w, \hat{r}) : g(w) \leqslant \tau \right\}.$$

As before, we consider the associated Lagrangian defined by

$$\mathcal{L}(w, \hat{r}, \lambda) = f(w, \hat{r}) + \lambda \left(g(w) - \tau \right).$$

Then we define a function of λ as

$$D(\lambda) = \min_{w \in \mathcal{W}} \max_{\hat{r} \in \mathcal{R}} \mathcal{L}(w, \hat{r}, \lambda).$$

Then the corresponding dual problem is given by

$$d^* = \max_{\lambda \ge 0} D(\lambda)$$

and the theory developed in Section 2 for general dual problems still holds.

5.1 Uncertainty in returns and no correlation

We consider the case when Σ is diagonal. In this case, the assumption of decomposability of f and g still holds. In particular, the dual problem reduces to

$$d^* = \max_{\lambda \ge 0} \min_{w \in \mathcal{W}} \max_{\hat{r} \in \mathcal{R}} \left\{ -\lambda \tau + f(w, \hat{r}) + \lambda g(w) \right\}$$

=
$$\max_{\lambda \ge 0} \left\{ -\lambda \tau + \sum_{i=1}^n \min_{w_i \in \mathcal{W}_i} \max_{\hat{r}_i \in \mathcal{R}_i} \left\{ \frac{1}{2} \sigma_i^2 w_i^2 - \hat{r}_i w_i + \lambda g_i (w_i - w_i^0) \right\} \right\}.$$

Then we can solve this dual problem with the same algorithm as given in Section 2 as long as finding an optimizer for the problem

$$\min_{w_i \in \mathcal{W}_i} \max_{\hat{r}_i \in \mathcal{R}_i} \left\{ \frac{1}{2} \sigma_i^2 w_i^2 - \hat{r}_i w_i + \lambda g_i (w_i - w_i^0) \right\}$$

can be efficiently done, algorithmically or otherwise. We will consider alternative methods more generally in the case with correlations.

5.2 Uncertainty in returns and certain correlation

We next consider the case when $\Sigma = D + uu^{\top}$, where D is a diagonal matrix containing asset variances and $u \in \mathbb{R}^n$ is a vector that models how a specific factor affects the entire asset universe. It is important to note that Σ is still positive semi-definite because

$$w^{\top}\Sigma w = w^{\top}Dw + w^{\top}uu^{\top}w = w^{\top}Dw + (w^{\top}u)^2 \ge 0,$$

because D is positive semi-definite by assumption. Then the objective function f is still a convex function.

The major issue here, however, is that we no longer can assume decomposability of $f + \lambda g$ into different assets. One might hope to change basis to diagonalize Σ , but the transaction costs are not basis-invariant, so this direction is fruitless. In this case, we apply the general algorithm from Section 2. Note that in the dual formulation, we now have

$$D(\lambda) = \min_{w \in \mathcal{W}} \max_{\hat{r} \in \mathcal{R}} \left\{ \frac{1}{2} w^{\top} \Sigma w - \hat{r}^{\top} w + \lambda g(w) - \lambda \tau \right\}.$$

Fortunately, $D(\lambda)$ is still a concave function with a minimizer λ^* . Furthermore, because the objective function is linear in \hat{r} , it must be maximized in this variable at a boundary point. Because g is lower semi-continuous, we can switch the minimum and maximums by the Minimax Lemma and write this as

$$D(\lambda) = \max_{\hat{r}_1 \in \{\underline{r}_1, \overline{r}_1\}} \cdots \max_{\hat{r}_n \in \{\underline{r}_n, \overline{r}_n\}} \min_{w \in \mathcal{W}} \left\{ \frac{1}{2} w^\top \Sigma w - \hat{r}^\top w + \lambda g(w) - \lambda \tau \right\}.$$

Next, we partition \mathcal{W} into 3^n regions corresponding to holding, buying, or selling each asset. Consider the following notation:

$$\left\{\begin{array}{rrr} \mathcal{W}_i^0 &=& \{w_i^0\}\\ \mathcal{W}_i^+ &=& \left[w_i^0, \overline{w}_i\right]\\ \mathcal{W}_i^- &=& \left[\underline{w}_i, w_i^0\right]. \end{array}\right.$$

Then if for any string $\alpha \in \{0, +, -\}^n$, we define

$$\mathcal{W}^{\alpha} = \bigotimes_{i=1}^{n} \mathcal{W}_{i}^{\alpha_{i}},$$

we can write the optimization problem as

$$D(\lambda) = \max_{\hat{r}_1 \in \{\underline{r}_1, \overline{r}_1\}} \cdots \max_{\hat{r}_n \in \{\underline{r}_n, \overline{r}_n\}} \min_{\alpha \in \{0, +, -\}^n} \min_{w \in \mathcal{W}^\alpha} \left\{ \frac{1}{2} w^\top \Sigma w - \hat{r}^\top w + \lambda g(w) - \lambda \tau \right\}.$$

It is now clear that, this problem consists of solving 6^n quadratic programming problems. However, we can cut down on this complexity by employing branch-and-bound techniques to only explore a subset of these combinations.

The idea of our branch-and-bound procedure is to consider a tree generated by choices to hold, buy, or sell each subsequent asset. If we can obtain a lower-bound to an entire sub-tree in constant time, and find that this lower-bound is greater than an upper-bound for the entire tree, then we can ignore that sub-tree. Then the key is to describe a constant time lower-bound of a sub-tree, as well as a procedure for obtaining an upper-bound to an entire tree. For simplicity, we let $\underline{r} = \overline{r}$ and $w^0 = 0$, though all proceeding arguments could be modified for the more general case.

Lemma 3. Let $\alpha \in \{0, +, -\}^n$ and $1 \leq k < n$. Consider the problem

$$d_k^{\alpha} = \min\left\{\frac{1}{2}w^{\top}\Sigma w - \hat{r}^{\top}w - \lambda\tau + \sum_{i=1}^k \lambda g_i(w_i) : w \in \mathcal{W}_1^{\alpha_1} \times \cdots \mathcal{W}_k^{\alpha_k} \times \mathcal{W}_{k+1} \times \cdots \times \mathcal{W}_n\right\}.$$

Then we have

 $d_{k+1}^{\alpha} \ge d_{k}^{\alpha} + \lambda b_{k+1} \chi_{\{\alpha_{k+1}=+\}} + \lambda s_{k+1} \chi_{\{\alpha_{k+1}=-\}},$

where χ is an indicator function.

Proof. This inequality is nearly trivial from the definition of d_k^{α} and the fact that $b_i, s_i \ge 0$. In particular, for $1 \le k < n$, let w be a minimizer of the (k + 1)-problem. Then this is also a feasible solution of the k-problem, so we have

$$d_{k+1}^{\alpha} = \frac{1}{2} w \top \Sigma w - \hat{r}^{\top} w - \lambda \tau + \sum_{i=1}^{k+1} \lambda g_i(w_i)$$

$$= \lambda g_{k+1}(w_{k+1}) + \frac{1}{2} w \top \Sigma w - \hat{r}^{\top} w - \lambda \tau + \sum_{i=1}^k \lambda g_i(w_i)$$

$$\geqslant d_k^{\alpha} + \lambda g_{k+1}(w_{k+1})$$

$$= d_k^{\alpha} + \lambda b_{k+1} \chi_{\{\alpha_{k+1}=+\}} + \lambda s_{k+1} \chi_{\{\alpha_{k+1}=-\}}.$$

Figure 5: Illustration of branch-and-bound method for decision to hold, buy, or sell at asset *i*. Sub-trees that have a lower-bound greater than the entire tree's upper-bound are ignored. Otherwise, the process is iterated.



The key to this definition is that each d_k^{α} is only a normal quadratic programming problem because the cost function is a constant in the region over which we are minimizing. Furthermore, given a minimal value of d_k^{α} , we can obtain a lower-bound on the value of d_{k+1}^{α} in constant-time for use in the branch-and-bound method. Furthermore, for at least one α , d_n^{α} will correspond to the actual Lagrangian subproblem, $D(\lambda)$.

Lemma 4. There exists an $\alpha \in \{0, +, -\}^n$ such that

$$d_0^{\alpha} \leq d_1^{\alpha} \leq \cdots \leq d_{n-1}^{\alpha} \leq d_n^{\alpha} = D(\lambda).$$

Proof. Let w be a minimizer of the Lagrangian sub-problem, so $\mathcal{L}(w, \lambda) = D(\lambda)$. Define $\alpha \in \{0, +, -\}^n$ via

$$\alpha_k = \begin{cases} 0 & \text{if } w_k = 0 \\ + & \text{if } w_k > 0 \\ - & \text{if } w_k < 0. \end{cases}$$

Then it is clear from the definition that $d_n^{\alpha} = D(\lambda)$. Furthermore, by the previous lemma, when $\lambda \ge 0$, we have

$$d_0^{\alpha} \leq d_1^{\alpha} \leq \dots \leq d_{n-1}^{\alpha} \leq d_n^{\alpha} = D(\lambda).$$

With these results in hand, we can devise an algorithm to solve the Lagrangian sub-problem via the branch-and-bound procedure.

Theorem 4. We can construct an optimal solution w^* to the Lagrangian sub-problem in the correlation model via the branch-and-bound algorithm shown in Algorithm 4.

Input: Correlation matrix Σ , expected returns \hat{r} , initial allocations w_i^0 , allocation limits \underline{w}_i and \overline{w}_i , trading fees b_i and s_i , a Lagrange multiplier λ , and the trading fee limit τ .

Output: An allocation w^* which minimizes the Lagrangian sub-problem $D(\lambda)$.

```
Create priority queue pq
```

 $\begin{aligned} \alpha \leftarrow (0, 0, \dots 0) \\ (w, d) \leftarrow \text{optimal allocation and value corresponding to } d_0^{\alpha} \\ (w^*, d^*) \leftarrow (w, \mathcal{L}(w, \lambda)) \\ \text{Push } (d, w, 1) \text{ to } pq \text{ with priority } d \\ \text{while } pq \text{ is not empty } \mathbf{do} \\ & (d, w, k) \leftarrow \text{pop}(pq) \\ \text{ for } \alpha_k \leftarrow \{0, +, -\} \mathbf{do} \\ & | d' \leftarrow d + \lambda b_k \chi_{\{\alpha_k = +\}} + \lambda s_k \chi_{\{\alpha_k = -\}} \\ & \text{ if } d' \ge d^* \text{ then Continue} \\ & (w', d') \leftarrow \text{ optimal allocation and value corresponding to } d_k^{\alpha} \\ & \text{ if } d' \ge d^* \text{ then Continue} \\ & \text{ if } \mathcal{L}(w', \lambda) \leqslant d^* \text{ then } (w^*, d^*) \leftarrow (w', \mathcal{L}(w', \lambda)) \\ & \text{ if } k < n \text{ then Push } (d', w', k + 1) \text{ to } pq \text{ with priority } d' \\ \text{ end} \\ \text{ end} \\ \text{ return } w^* \end{aligned}$



Proof. The queue in the algorithm above will only add elements up to k = n, so at most there can be 3^n elements added to the queue. Therefore, this algorithm will terminate eventually. To see that it will return an optimal allocation of the Lagrangian sub-problem, we consider the previous lemmas. We know there exists an $\alpha \in \{0, +, -\}^n$ such that

$$d_0^{\alpha} \leqslant d_1^{\alpha} \leqslant \cdots \leqslant d_n^{\alpha} = D(\lambda).$$

At any point in the algorithm, $d^* = \mathcal{L}(w, \lambda)$ for some w. Then for any k, we have

$$d_k^{\alpha} \leqslant D(\lambda) \leqslant d^*.$$

Therefore, the continues inside the while loop will never be triggered along this sequence corresponding to α . Finally, if w_n^{α} is the optimal allocation corresponding to d_k^{α} , then we will have

$$\mathcal{L}(w_n^{\alpha}, \lambda) = D(\lambda) \leqslant d^*,$$

so at some point w^* will be set to this optimal allocation. Because w^* is a minimizer of the Lagrangian, the condition to reset w^* will never we met again unless by another minimizer, so w^* will be an optimal allocation of the Lagrangian sub-problem.

For more information on the theory and general results concerning branch-and-bound methods, see [2]. For our purposes, we will note that this allows us to obtain an optimizing vector w^* to the Lagrangian sub-problem. We can then compute a super-gradient as $D'(\lambda) = g(w^*) - \tau$, and we can proceed with the Algorithm 1 from Section 2.

5.3 Uncertainty in both returns and correlation

Lastly, we consider the case when $\Sigma = D + uu^{\top} + \Delta$, with Δ a positive semi-definite matrix that models error in our measurement of the covariance matrix. Here, Δ is unknown but bounded: we assume $\|\Delta\| \leq \rho$ where $\rho \geq 0$ is a known scalar and $\|\cdot\|$ denotes the largest singular value norm. Once again, it is important that Σ is still a positive semi-definite operator, as discussed in the previous subsection. Because the results of the previous subsection are a special case of this problem, with $\rho = 0$, we expect that the results from the general binary and ternary models will still not hold.

Consider the dual formulation corresponding to this problem. If we hope to apply the general algorithm from Section 2, then we need to be able to solve the optimization problem

$$D(\lambda) = \min_{w \in \mathcal{W}} \max_{\hat{r} \in \mathcal{R}} \max_{\|\Delta\| \leq \rho} \left\{ \frac{1}{2} w^{\top} \Sigma w - \hat{r}^{\top} w + \lambda g(w) - \lambda \tau \right\}.$$

The key to simplifying this problem is to note that the maximization over all positive semidefinite matrices is easy to do analytically.

Lemma 5. Let $w \in \mathbb{R}^n$. Then if we maximize over all positive semi-definite matrices Δ with largest singular value norm $\|\Delta\| \leq \rho$, we have

$$\max_{\|\Delta\|\leqslant\rho} w^\top \Delta w = \rho w^\top w.$$

Proof. If w = 0, then the result is trivial. Otherwise, the key is to recall that the largest singular value norm of a matrix Δ is equal to

$$\max_{w \neq 0} \frac{\|\Delta w\|}{\|w\|} = \|\Delta\| \leqslant \rho.$$

Then for any w, we have $\|\Delta w\| \leq \rho \|w\|$. By the Cauchy-Schwarz inequality, we have

$$w^{\top} \Delta w \leqslant \|w\| \|\Delta w\| \leqslant \rho w^{\top} w.$$

Finally, for any $w \neq 0$, we can construct a positive semi-definite matrix Δ which is ρ times the projection operator onto the subspace spanned by w. This matrix has $\|\Delta\| = \rho$ and clearly satisfies $w^{\top}\Delta w = \rho w^{\top} w$.

Therefore, we conclude that $\max_{\|\Delta\| \leq \rho} w^{\top} \Delta w = \rho w^{\top} w$.

Using this result, we can simplify the problem above into a problem from the previous subsection.

Theorem 5. Adding uncertainty to asset correlations in the form of a positive semi-definite matrix Δ with largest singular value norm $\|\Delta\| \leq \rho$ is equivalent to adding a constant ρ to the variance of each asset.

Proof. To prove this, we simply consider the $D(\lambda)$ from above. Because the maximization over Δ comes first, we can immediately use the lemma from above.

$$D(\lambda) = \min_{w \in \mathcal{W}} \max_{\hat{r} \in \mathcal{R}} \max_{\|\Delta\| \leq \rho} \left\{ \frac{1}{2} w^{\top} \left(D + u u^{\top} + \Delta \right) w - \hat{r}^{\top} w + \lambda g(w) - \lambda \tau \right\}$$
$$= \min_{w \in \mathcal{W}} \max_{\hat{r} \in \mathcal{R}} \left\{ \frac{1}{2} w^{\top} \left(D + \rho + u u^{\top} \right) w - \hat{r}^{\top} w + \lambda g(w) - \lambda \tau \right\}.$$

Then we have shifted the uncertainty Δ onto the diagonal of the covariance matrix. This corresponds to an increase in asset variances in D.

This result shows that really the most general problem we are considering in this paper is that from the previous subsection. For an implementation of the correlation model in MAT-LAB, see the general algorithm in Appendix A.

6 Conclusion

In this paper, we have examined the general principle of duality and its applications to portfolio optimization with constraints in the form of transaction costs. We considered three main models of increasing complexity and generality, including the binary model, the ternary model, and the correlation model. The overarching theme is that decomposability of objective functions and transaction costs allows for simplifications of the problem. In the correlation model, the separability of the problem was violated, but we used a branch-and-bound algorithm to dramatically cut down on the overall run-time.

Future work could focus on improving the implementation of these algorithms, and in particular taking advantage of obvious parallelism opportunities. For example, the binary and ternary models both require sequential optimization of individual assets after decomposing the objective function, which could easily be done in parallel. Furthermore, the linear programming algorithm used in the ternary model and quadratic programming algorithm used in the correlation model can both be sped up using parallelism. Finally, and most importantly, the branch-and-bound algorithm used in Section 5 performs several independent minimization problems on different sub-trees of the combinatorial search space. These could each be done in parallel to boost run-time of the algorithm.

Another avenue of future work would be to develop further approximation methods of the dual problem in the correlation method. For example, if it were not for the appearance of the non-polynomial transaction cost terms, one promising strategy would be to change to a basis of eigenvectors of the covariance matrix Σ . One could hope to approximate the transaction costs with functions that allow this change of basis. This would lead to a variant of the branch-and-bound algorithm where we instead perform relaxation on the cost function. This could substantially increase the speed of obtaining an ϵ -suboptimal solution to the dual problem in practice.

References

- [1] S. Boyd and L. Vandenberghe. Convex Optimization. Cambridge University Press, 2004.
- [2] M.J. Brusco and S. Stahl. Branch-and-Bound Applications in Combinatorial Data Analysis. Springer, 2005.
- [3] V. Chvatal. *Linear Programming*. W. H. Freeman, 1983.
- [4] T.H. Cormen, C.E. Leiserson, R.L. Rivest, and Stein C. Introduction to Algorithms, Second Edition. The MIT Press, 2001.
- [5] Signal Processing Microelectronics. Qpc quadratic programming in c, 2013.

Appendix A. Numerical Implementation

In this Appendix, we describe a general solver for all three models which was implemented in MATLAB. In this section we add the extra constraint on overall net wealth being invested,

$$\sum_{i=1}^{n} w_i \leqslant 1$$

It is worth noting that this condition breaks the separability that allowed for general polynomialtime results in the binary and ternary models.

In the attached MATLAB codes, we provide one general program which computes allocations corresponding to optimizers (w^*, λ^*) of the dual problem in each model, subject to constraints on transaction costs and the overall net investment constraint above. Because the correlation model generalizes both the ternary and binary model, we use the super-gradient algorithm in Algorithm 1 with the Lagrangian dual optimized via the branch-and-bound algorithm in Algorithm 4. To solve quadratic programming problems, we use the a C-based Quadratic Programming MATLAB interface known as QPC[5]. In particular, we use a primaldual predictor-corrector algorithm for quadratic programming problems constrained by linear inequalities. We use sparse matrices in the binary and ternary models and dense matrices in the correlation model.

To test this algorithm in general cases, we construct random inputs. In order to make sure that the transaction cost and allocation constraints actually bind in general, we generated inputs as follows. Returns are drawn uniformly from [-1, 1], standard deviations are set to 1, upper bounds to 1, lower bounds to -1, buy costs to 0.01 and sell costs to 0.02. The transaction cost limit τ is set to one-half the sum of the buy costs plus a uniform random number on [0, 0.001] so that the inputs are generic in terms of transaction costs. Finally, as applicable, each component of the vector u which controls correlations is drawn uniformly from [-0.05, 0.05].

In Figure 6, we show a graph of the average run-time versus number of assets and desired accuracy ϵ in the full correlation model with a branch-and-bound algorithm. Each line corresponds to a different cut-off for the accuracy of the solution to the dual problem, as described

Figure 6: Timings of full correlation branch-and-bound algorithm with varying number of assets and desired accuracy, ϵ .



Figure 7: Timings of binary, ternary, and full correlation models in the general purpose algorithm with varying number of assets and desired accuracy $\epsilon = 0.01$.



in Algorithm 1. In particular, the increase in computation time appears to be linear in ϵ , but exponential in the number of assets. In Figure 7, we show a comparison of the scaling of the binary model, the ternary model, and the full correlation model in this general purpose implementation. In particular, while the correlation model appears to scale exponentially with respect to the number of assets, even with the branch-and-bound algorithm, the binary and ternary models exhibit superior scaling.