# ORI 391Q Final Examination

## The Rules

- A signed copy of this page is to be used as the cover page for your submission.
- This exam consists of four questions, each presented on a separate page. The problems are weighted equally, but they are not equally difficult nor equally long.
- Your completed examination is due by **5pm *sharp* on Thursday, May 13**. You must deliver it to me in ETC 5.128A during my office hours listed below. Alternative arrangements of any kind require compelling reasons and must be agreed upon by me in advance. Do *not* leave the exam with anyone else, or leave it in my office unattended; you *will* lose points.
- All work must be *entirely* your own. There is to be absolutely no collaboration on this exam. No matter how trivial the issue, you may communicate *only* with me about it. If you turn in the exam early, you may still not discuss it with others until after everyone else has submitted it.
- You may use the textbook, any materials I have distributed to you, and any notes you have taken. Use of other resources is strongly discouraged. At this point especially, if you feel the need to consult other sources, you're probably missing something. Regardless, my standard citation rules apply.
- Here are my office hours during the final exam period:
  - Monday, May 3rd: 2-3PM
  - Wednesday, May 5th; Monday, May 10th; Wednesday, May 12th: 12:30-3PM
  - Thursday, May 13th: 3PM-5PM

  Feel free to ask questions by email as well. Likewise, make sure to read *your* email regularly in case I send out clarifications.
- If you are stuck on a problem, I will consider offering you a hint in exchange for a deduction, though I may do so only near the end of the exam period.
- I will deduct points from long, needlessly complex solutions, even if they are correct. If your solution to a problem goes on for pages and pages, you should try to figure out a simpler one.
- Your work *must* be clear and legible. Make sure that your final answer to each subproblem is collected in a single place and clearly marked as such. If I cannot easily find the final answer and read the supporting work, you will lose points.

I acknowledge that I have read this page and have abided by the rules set forth in it.

Sign here:______________________________

Print your name:______________________________

1. Consider the Markowitz portfolio optimization problem described in §4.7.6 on pages 185-187. We shall use the following data:

```
pbar = [.12 .10 .07  .03]';
S = [0.0064 0.0008 -0.0011 0;
     0.0008 0.0025  0      0;
    -0.0011 0       0.0004 0;
     0      0       0      0];
```

(a) Consider the standard risk/return multicriterion model

$$\begin{array}{ll} \text{maximize} & (\bar{p}^T x, -\sqrt{x^T \Sigma x}) \\ \text{subject to} & \mathbf{1}^T x = 1, \ x \succeq 0 \end{array}$$

Generate the Pareto optimal tradeoff curve for this model, with the standard deviation on the $x$ axis. Determine the *exact* locations of both endpoints. Also create an area plot of the optimal portfolios $x$ as a function of standard deviation.

Your plots should look very similar to those found in Figure 4.12, but not precisely the same. Yours don't need to be as pretty; in fact, to generate the area plot, you can just do this:

```
plot( std, xx(1,:), std, xx(1,:)+xx(2,:), std, xx(1,:)+xx(2,:)+xx(3,:) );
```

(b) Assume that the price change vector $p$ is a Gaussian random variable, with mean $\bar{p}$ and covariance $\Sigma$, and consider the multicriterion model

$$\begin{array}{ll} \text{maximize} & (\bar{p}^T x, -\,\mathbf{prob}(p^T x \le 0)) \\ \text{subject to} & \mathbf{1}^T x = 1, \ x \succeq 0 \end{array}$$

The quantity $\mathbf{prob}(p^T x \le 0)$ is the *total loss risk*—the probability that the portfolio loses its entire value (or worse). Generate the Pareto optimal tradeoff curve for this model, with $\eta = \mathbf{prob}(p^T x \le 0)$ on the $x$ axis. Determine the *exact* locations of both endpoints. Use a logarithmic scale for $\eta$, and include $\eta \ge 10^{-4}$ on your plots. Also make an area plot of the optimal portfolios $x$ versus $\eta$, again with a logarithmic scale for $\eta$.

*Hint.* The Matlab functions `erfc` and `erfcinv` can be used to evaluate the Gaussian CDF $\Phi(x) = (1/\sqrt{2\pi}) \int_{-\infty}^{x} e^{-t^2/2} dt$ and its inverse. Specifically, $\Phi(u) = \frac{1}{2}\mathbf{erfc}(-u/\sqrt{2})$.

(c) Consider the composite model

$$\begin{array}{ll} \text{maximize} & (\bar{p}^T x, -\sqrt{x^T \Sigma x}) \\ \text{subject to} & \mathbf{prob}(p^T x \le 0) \le 0.005 \\ & \mathbf{1}^T x = 1, \ x \succeq 0 \end{array}$$

This is the standard risk/return model from part (a), modified to bound the total loss risk by 0.5%. Generate a Pareto optimal tradeoff and an area plot as well. In this case, you may determine the endpoints numerically.

2. Let $\mathcal{E}$ be an ellipsoid in $\mathbf{R}^n$ with center $c$. The *Mahalanobis distance* of a point $z$ to the ellipsoid center $c$ is the factor by which we need to scale that ellipsoid about its center so that $z$ lies on its boundary:

$$M(z, \mathcal{E}) = \inf\{t \geq 0 \mid z \in c + t(\mathcal{E} - c)\}.$$

Note that $z \in \mathcal{E}$ if and only if $M(z, \mathcal{E}) \leq 1$. Using this measure, we can construct a measure of the distance between a point $z$ and an ellipsoid $\mathcal{E}$:

$$\mathbf{dist}_M(z, \mathcal{E}) \triangleq (M(z, \mathcal{E}) - 1)_+ = \max\{M(z, \mathcal{E}) - 1, 0\}$$

This is a potential replacement for the standard distance measure $\mathbf{dist}(z, \mathcal{E}) = \min_{y \in \mathcal{E}} \|z - y\|$; and like this original measure, $\mathbf{dist}_M(z, \mathcal{E}) = 0$ if and only if $z \in \mathcal{E}$.

Now we can describe the problem. We are given $m$ points $x_1, \ldots, x_m \in \mathbf{R}^n$, and we wish to examine the optimal trade-off between the volume of the ellipsoid $\mathcal{E}$ and the total Mahalanobis distance of the points to the ellipsoid:

$$\text{minimize} \quad \left(\mathbf{vol}(\mathcal{E}), \sum_{i=1}^{m} \mathbf{dist}_M(x_i, \mathcal{E})\right).$$

This can be considered a robust version of finding the smallest volume ellipsoid that covers a set of points, since allowing the total Mahalanonis distance to be positive means that one or more of the points are allowed to lie outside of the ellipsoid.

The data for this problem is generated by the file `final_ellipse.m`. Follow these steps:

- Compute the area $\mathbf{vol}(\mathcal{E}_0)$ of the smallest ellipse $\mathcal{E}_0$ that contains all of the points.
- Construct the optimal trade-off between ellipsoid volume and Mahalanobis distance for 10 points in the range $[0.1 \cdot \mathbf{vol}(\mathcal{E}_0), \mathbf{vol}(\mathcal{E}_0)]$.
- On a single figure, plot the data points, the minimum volume covering ellipsoid, and the ellipse corresponding to each of the 10 points in the tradeoff.

You do not need to submit the numeric value of $\mathbf{vol}(\mathcal{E}_0)$ or a tradeoff curve—just the final ellipse figure, and the source code used to generate it.

To simplify your work, I have provided a function `plot_ellipse.m` that plots an ellipse described by $\{x \mid x^T P x + q^T x + r \leq 0\}$. To use it, simply call `plot_ellipse(P,q,r,ltype)`, where `ltype` is a standard `plot` linetype (for instance, `'b-'` would plot a blue ellipse). Use the `hold on` and `hold off` functions to plot multiple ellipses on the same plot. Of course, if you choose a different parametrization for your ellipse, you will need to construct `P`, `q`, and `r` appropriately.

3. A single processor can adjust its speed in each of $T$ time periods, labeled $1, \ldots, T$. Its speed in period $t$ will be denoted $s_t$, $t = 1, \ldots, T$. The speeds are bounded above and below, and in how fast they can change between periods:

$$S^{\min} \leq s_t \leq S^{\max} \qquad |s_{t+1} - s_t| \leq R$$

The energy consumed by the processor in period $t$ is given by $\phi(s_t) = \alpha + \beta s_t + \gamma s_t^2$, where $\alpha$, $\beta$, and $\gamma$ are known and satisfy $\alpha \geq 0$ and $\beta + 2\gamma > 0$.

The processor must handle $n$ jobs, labeled $1, \ldots, n$. Each job has an availability time $A_i$ and a deadline time $D_i$, and a total amount of work required $W_i$:

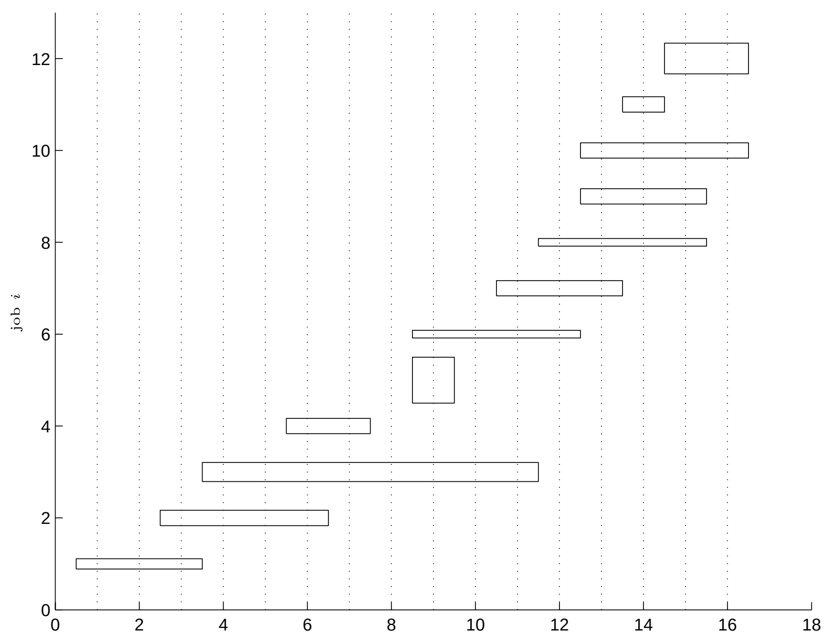
$$(A_i, D_i) \in \{1, \ldots, T\}, \quad D_i \geq A_i, \quad W_i \geq 0.$$

You may assume that in each time period, there is at least one job available; that is, for each $t \in \{1, \ldots, T\}$, there is at least one $i$ for which $A_i \leq t \leq D_i$.

In period $t$, the processor allocates its effort across the $n$ jobs as $\theta_t$, where $\mathbf{1}^T\theta_t = 1$, $\theta_t \succeq 0$. Here $\theta_{ti}$ (the $i$th component of $\theta_t$) gives the fraction of the processor effort devoted to job $i$ in period $t$. Respecting the availability and deadline constraints requires that $\theta_{ti} = 0$ for $t < A_i$ or $t > D_i$. To complete the jobs we must have

$$\sum_{t=A_i}^{D_i} \theta_{ti} s_t \geq W_i, \quad i = 1, \ldots, n.$$

Your task is to minimize the total energy required to complete the scheduled jobs. The data for the problem is generated by the file `final_sched.m`: $S^{\min}$, $S^{\max}$, $R$, $\alpha$, $\beta$, $\gamma$, $A$, $D$, and $W$. In addition, the file will reproduce the figure below—a plot showing the requested times for each job. The width of box $i$ is $D_i - A_i + 1$, and the total area is proportional to the work $W_i$.




Give the energy obtained by your speed profile and allocations. Plot these using the command `bar(theta.*s(:,ones(1,n)),1,'stacked')`, where $s$ is the $T \times 1$ vector of speeds, and $\theta$ is the $T \times n$ matrix of allocations with components $\theta_{ti}$. This will show, at each time period, how much effective speed is allocated to each job. The top of the plot will show the speed $s_t$. You don't need to turn in a color version of this plot; black and white is fine.

4. Let us revisit the educational testing problem given in Homework 11. If $\Sigma$ is positive definite, then the constraint $x \succeq 0$ is redundant. If we remove it, the inequality constraints $Z_{ii} \geq 0$ in the dual are replaced with equality constraints; that is,

$$\begin{array}{ll} \text{maximize} & \mathbf{1}^T x \\ \text{subject to} & \Sigma - \mathbf{diag}(x) \succeq 0 \end{array} \qquad \begin{array}{ll} \text{minimize} & \mathbf{tr}\,\Sigma Z \\ \text{subject to} & Z_{ii} = 1,\ i = 1, 2, \ldots, n \\ & Z \succeq 0 \end{array}$$

Construct a barrier method for solving the *dual* problem. Your method must start at a strictly feasible point and maintain strict feasibility at every iteration. Upon termination, your algorithm must return a feasible primal-dual pair $(x, Z)$ satisfying

$$\mathbf{tr}\,\Sigma Z - \mathbf{1}^T x \leq 0.01\,\mathbf{tr}\,\Sigma Z.$$

Your only other constraint: the cost of each iteration cannot exceed $O(n^3)$ flops.

To test your algorithm, use the same data specified for Homework 11:

```
randn('state',0);
Sigma = randn(30,50);
Sigma = Sigma * Sigma';
```

Plot $\mathbf{tr}\,\Sigma Z - p^*$ versus the number of iterations. Provide your full source code as well.

**Hint**: your may find it helpful, even necessary, to insert statements such as `dZ = 0.5*(dZ+dZ')` or `Z=0.5*(Z+Z')` into your code at key points to ensure that your matrices $Z$ and $\Delta Z$ remain symmetric.