Fast Portfolio re-Balancing as a Fractional Linear Program

John Mount*

August 12, 2010

1 Introduction

Here we write out a concrete and complete implementation of a fast risk re-balance system. That is a system that takes a current portfolio (set of holdings) plus estimates of future returns and quickly solves for a trade that moves to an optimal new portfolio. We will use the simple mean-variance Sharpe ratio criteria to balance return and risk and we further incorporate trade costs. These methods are similar to those of [RV02, VR07].[1] The emphasis is on approximations that allow the use of a very fast optimizer. The use of an optimizer allows the requirement of an intelligent decision to be encoded declaratively and then mechanically solved for.

2 Model

This model is deliberately simplified to allow quick implementation, execution and estimation.

Assume we are trading a number of core securities (say stocks) and shorting a hedging instrument (say a future in an index representing a basket of the core securities). Further assume an Arbitrage Pricing Theory[RR84, HW05] type formulation where for each security we have a factor structure for near-term rate of return:

$$r_i \approx \alpha_i + \beta_{i,1} F_1 + \cdots + \beta_{i,n} F_n + \eta_i = \alpha_i + \beta_i^\top F + \eta_i$$

(where n is small compared to m: the total number of securities being tracked).

That is we are writing the near-term return rate as a linear function of the following definitions, coefficients and random variables:

*web: http://www.win-vector.com/ email: mailto:jmount@win-vector.com

[1] But we differ in sticking with a mean/variance characterization of risk. Whereas the references work more directly with an efficient frontier of non-dominated portfolios.

- α_i: The unique to the security return (this term should be zero for un-modeled securities and is non-zero for securities that we wish to trade in or out of)

- β_i: A vector of loadings into shared factors F (to be precise β is a vector of loading coefficients and F is a vector of random variables).

- $\mathrm{E}[x]$: denotes expected value of the random variable x.

- $\mathrm{var}[x]$: denotes the variance of the random variable x (or equivalently the expected squared deviation from the mean: $\mathrm{E}[(x - \mathrm{E}(x))^2]$).

- F_i: The random variables representing different market factors (like banking, tech, future currency prices). These represent possible drivers of price change. We will allow these to have non-zero expectations (or estimated expectations) which $\mathrm{E}[F_i] = f_i$.

- η_i: The un-diversifiable risk (a mean-zero random variable) unique to this security.

- λ_i: The fraction of the hedging instrument that is invested in the i-th security (i.e. the composition of the index or future we use for hedging).

- $\mathrm{cost}(y, h, x, g)$: The cost of trading from a portfolio with y_i dollars allocated in the i-th security and h dollars short in "the hedging instrument" (a broad market future or index representing a basket of securities) to a portfolio with x_i dollars in the i-th security and g short in the hedging instrument (we discuss cash later).

The goal is to use our estimates of $\alpha, \beta, F, f, \eta$ to find a trade that moves us from our current holdings to a new set of holdings that have a higher expected risk-adjusted payoff than our current position. If we represent our current portfolio as a vector x representing the number of dollars invested in each security and a scalar g representing the number of dollars short in the hedging instrument then our net exposure to the i-th security is $x_i - \lambda_i g$ where λ is the vector that represents what fraction of the hedging instrument is allocated to the ith security.

We can then try to find a new allocation y, h (y again a vector representing the total dollars in each security and h number of dollars short in the hedging instrument). For simplicity we will assume there is one security that represents cash. In a mean-variance formulation we would want to maximize:

$$\frac{(\mathrm{E}[\mathrm{return}(y, h)] - \mathrm{cost}(y, h, x, g))^2}{\mathrm{var}[\mathrm{return}(y, h)]}$$

(while also avoiding some trivial cases like negative return and $\mathrm{var}[\mathrm{return}(y, h)] = 0$).

To do this we will break these terms into small pieces and re-write this as a tractable optimization problem.

3 Solution

3.1 Without Costs

We first examine a simpler version of the problem that ignores costs (but does include risk). The Sharpe ratio[Lo02, Mou08] formulation is maximize:

$$\frac{(\mathrm{E}[\mathrm{return}(y,h)])^2}{\mathrm{var}[\mathrm{return}(y,h)]}.$$

We will instead solve

$$\begin{aligned}
\text{Maximize:} \quad & \mathrm{E}[\mathrm{return}(y,h)] \\
\text{Where:} \quad & \\
\mathrm{var}[\mathrm{return}(y,h)] \quad = \quad & 1.
\end{aligned}$$

As long as we keep the problem formulation homogeneous (that is invariant under simultaneous scaling of all variables) the two problems share optimal solutions (modulo the fact that the second form is doing a better job of keeping track of the sign of the return).

Next we expand out the expected return $\mathrm{E}[\mathrm{return}(y,h)]$ and risk $\mathrm{var}[\mathrm{return}(y,h)]$ terms.

We say the expected return is equal the sum of α_i and f_i times our net exposure to the i-th security:

$$\mathrm{E}[\mathrm{return}(y,h)] = (\alpha + f)^\top (y - h\lambda).$$

Our vector of net exposures to each of the factors F is given by:

$$\beta^\top (y - h\lambda)$$

where β is the matrix whose i-th row is the factor loadings for the i-th security. Then if C is the n by n covariance matrix between the n factors F we have:

$$\begin{aligned}
\mathrm{var}[\mathrm{return}(y,h)] &= (y - h\lambda)^\top \beta C \beta^\top (y - h\lambda) + \sum_i \eta_i^2 (y_i - h\lambda_i)^2 \\
&= (y - h\lambda)^\top (\beta C \beta^\top + E)(y - h\lambda)
\end{aligned}$$

where E is the diagonal matrix such that $E_{i,i} = \eta_i^2$.

This is just trying to say the variance is a sum of contributions from the net factor exposures plus contributions from the un-diversifiable risks.

So our optimization problem (in variables y, h) is now:

$$\begin{aligned}
\text{Maximize:} \quad & (\alpha + f)^\top (y - h\lambda) \\
\text{Where:} \quad & \\
(y - h\lambda)^\top (\beta C \beta^\top + E)(y - h\lambda) \quad = \quad & 1.
\end{aligned}$$

Let us change variables work with a new vector z such that

$$z = y - h\lambda.$$

In this perspective we are trying to find a z solving:

$$\begin{aligned} \text{Maximize:} & \quad (\alpha + f)^\top z \\ \text{Where:} & \\ z^\top(\beta C \beta^\top + E) z & = 1. \end{aligned}$$

The Karush-Kuhn-Tucker conditions of optimality[Kuh06] say that at the optimum we must have:

$$\nabla_z (\alpha + f)^\top z = \gamma \nabla_z z^\top (\beta C \beta^\top + E) z$$

where γ is an unknown scalar constant. It is not hard to see what the relation between the unknown γ should and the unknown z is (if you know one the other can be had by inspection). Passing the gradient operator through the last equation we have:

$$(\alpha + f)^\top = 2\gamma z^\top (\beta C \beta^\top + E)$$

So:

$$z \propto ((\beta C \beta^\top + E)^\top)^{-1} (\alpha + f)$$

Again, this is the usual statement that the gradient of the objective function is some scaling (γ) of the gradient of the constraint. Since we have a homogeneous problem and do not care about scaling we see that our solution is any multiple of:

$$\widehat{z} = ((\beta C \beta^\top + E)^\top)^{-1} (\alpha + f)$$

with the sign of the scaling determining if we are maximizing loss or gain. This is the same solution we get if we had (equivalently) formulated this problem as a Rayleigh quotient. In solving for z (instead of solving for y, h) we have solved for the optimal net-exposure (between direct ownership and the hedge) for each of the core securities. We can inspect to a reasonable solution by setting $h = \max_{i : z_i < 0} z_i \lambda_i$ and $y = z + h\lambda$.

Notice that this is very fast in that the solution is simple linear algebra (and we may even use the previous portfolio as a starting point in solution).

3.2 With Costs

Once we introduce costs we are at the very least attempting to maximize a slightly more complicated form like:

$$\frac{\mathrm{E}[\mathrm{return}(y, h)] - \mathrm{cost}(y, h, x, g)}{\sqrt{\mathrm{var}[\mathrm{return}(y, h)]}}.$$

A candidate for the new term $\mathrm{cost}(y, h, x, g)$ is:

$$\mathrm{cost}(y, h, x, g) = \sum_i w_i |y_i - x_i| + v|h - g|.$$

where the vector w and scalar v are some stand-in for per-dollar trade costs plus costs of uniformed trading (such as one half of the bid/ask gap over share price).[2]

The standard technique for dealing with the absolute value operator is to introduce a few more variables and constraints as follows:

$$\begin{aligned} d_i &\geq y_i - x_i \\ d_i &\geq x_i - y_i \\ c &\geq h - g \\ c &\geq g - h \end{aligned}$$

and re-write the cost function as:

$$\text{cost}(y, h, x, g) = \sum_i w_i d_i + vc.$$

Since we are maximizing return we are also minimizing costs which pushes these new variables into the relevant constraints.

Now that we have introduced a number of linear inequality constraints we might as well bring in the sign constraints $y, h \geq 0$ and a conservation constraint $1^\top x + g = 1^\top y + h$. If we don't want to trade the hedge we can enforce that with an extra constraint of the form $g = h$, similarly we can add in any desired trades by introducing constraints on targeted y_i.

Now that we have a number of linear inequalities the complicated form of the denominator starts to be a problem. If our entire objective were a mere linear function we could apply linear programming (which in this day and age is very very fast). But we are held up in that we have both a square root and a ratio in our formulation (which we will deal with next).

3.3 Naive Linear Constraints

So let us try to simplify into a pure linear programming formulation. We want to maximize $\text{E}[\text{return}(y, h)] - \text{cost}(y, h, x, g)$ and again treat the value of $\text{var}[\text{return}(y, h)]$ as a constraint.

If we had, as in the earlier cost-free subsection, no cost factors (and no additional constraints) we would be maximizing $\frac{\text{E}[\text{return}(y,h)]}{\sqrt{\text{var}[\text{return}(y,h)]}}$. As we said before: at the optimal point we expect the gradient of the objective function (with respect to our variables) to be zero.[3] A solution to the unconstrained problem should obey the equalities:

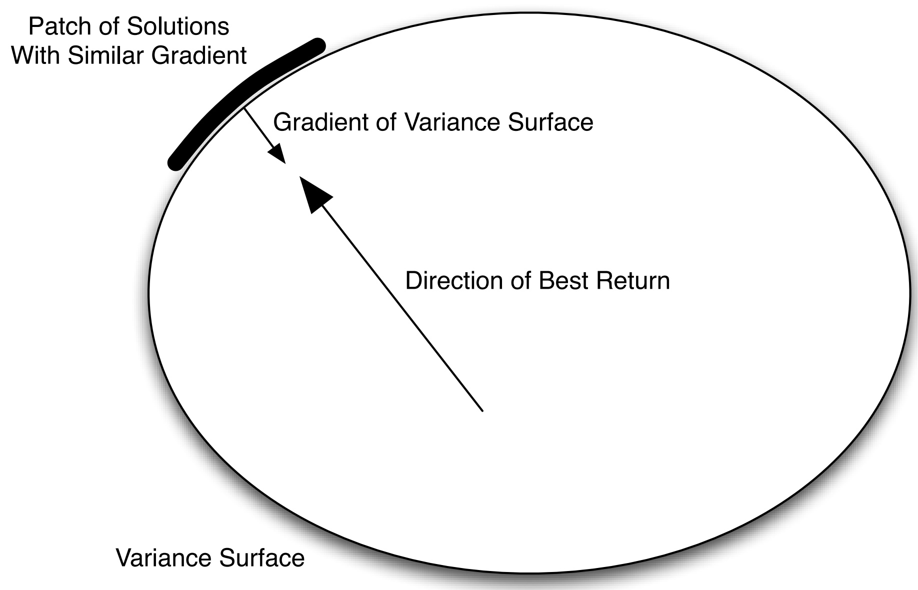
$$\nabla_{y,h} \text{E}[\text{return}(y, h)] = \gamma \nabla_{y,h} \text{var}[\text{return}(y, h)]$$

[2] It would be nice to introduce a penalty depending on the number of securities traded at all, but this kind of constraint does not have such a nice optimization formulation until we push into the more expensive world of search or mixed integer programming.

[3] In this case we introduce symbols $a = \text{E}[\text{return}(y, h)]$ and $b = \text{var}[\text{return}(y, h)]$ and recall from calculus the derivative of the ratio: $(ab^{-1/2})' = (a'b^{-1/2} - \frac{1}{2}ab^{-3/2}b')$. At a maximum this latest quantity would be zero. This is true when: $a' = \frac{1}{2}(a/b)b'$, which is the same conditions as in the previous section.

(we say equalities because this is a vector equation and γ is again the Lagrangian coefficient representing the correct relative pricing of risk versus return near the optimum).

This gives a plausible (but not good) idea for a large set of additional linear inequalities to add to our constrained system. The idea is to copy over some approximation of the above system to ask that the cost-sensitive solution fall near the original unconstrained system. This makes sense and does indeed force a new solution to be near the unconstrained solution. Schematically this is because the set of solutions is nicely parametrized by its gradients (just as you can tell where you are on a circle by being told which direction is pointing out):



Unfortunately this works "too well" as we have seen in earlier sections that $\nabla_{y,h}\text{var}[\text{return}(y, h)]$ essentially reads off y, h so constraint directly involving this term just copy solutions from the cost-insensitive system into the cost-sensitive system without giving enough freedom to actually reduce costs.

3.4 Better Linear Constraints

To get a better set of constraints we need to step back from the math a bit and look at what we expect from the problem itself. What we would like is to find a solution that has:

- similar expected return (exposure to α)
- similar expected factor risk (exposure to β)
- similar total risk
- reduced or minimal trade costs.

The most plausible conditions that this would be possible is that there are securities (or combinations of securities) that have profit and risk exposure similar to what is seen in the cost-insensitive solution that have lesser trade-costs. In this form it means there is money to be saved in not incurring expensive trade-costs without a reason (reasons being large α or unique utility in hedging).

We list out our wants as new constraints. Let $\widehat{y}, \widehat{h}$ be optimal solutions inspected from the earlier cost-free problem and let ϵ be a small constant (used to introduce some "slop" or "play" into the problem).

Then from an original problem point of view it is reasonable to want:

- Similar expected return

$$(\alpha + f)^\top (y - h\lambda) \geq (1 - \epsilon)(\alpha + f)^\top (\widehat{y} - \widehat{h}\lambda)$$

- Similar expected component-wise factor risk

$$\begin{array}{rcl} C\beta^\top (y - h\lambda) & \geq & (1 - \epsilon) C\beta^\top (\widehat{y} - \widehat{h}\lambda) \\ C\beta^\top (y - h\lambda) & \leq & (1 + \epsilon) C\beta^\top (\widehat{y} - \widehat{h}\lambda) \end{array}$$

  Here we are using the fact that the number of factors (n) is smaller than the number of securities (m) so that we only get $2n$ inequalities from this step and thus do not over-constrain the solution.

- Similar total risk

  The problem here is the un-diversifiable risk. At this point we have to change our formulation to make the problem tractable. Up until now we have been modeling risk/variance as $(y - h\lambda)^\top (\beta C \beta^\top + E)(y - h\lambda)$, that is as an additive combination of $(y - h\lambda)^\top \beta C \beta^\top (y - h\lambda)$ and $(y - h\lambda)^\top E (y - h\lambda)$. The previous bullet-point models the $(y - h\lambda)^\top \beta C \beta^\top (y - h\lambda)$, or factor-exposure portion. What remains is what to do with the un-diversifiable risk.

  One thing we note at this point is that our current model is modeling the un-diversifiable risks as independent (this is implicit in the fact that E is a diagonal matrix) which means that while they can not cancel each other it is still considered less risky to have three different \$1 net-exposures than a single \$3 net-exposure. At this point we are going to switch to a more pessimistic net-exposure risk model and instead of using $(y - h\lambda)^\top E (y - h\lambda) = \sum_i \eta_i^2 (y_i - h\lambda_i)^2$ as our un-diversifiable risk contribution we will use $\text{undiv}(y, h) = \sum_i \eta_i |y_i - h\lambda_i|$.

- Reduced or minimal trade costs.

  To reduce trade costs we will add them to the objective function in the form we gave above: $\text{cost}(y, h, x, g) = \sum_i w_i |y_i - x_i| + v|h - g|$ (where x is the vector that represents our current security holdings and g represents our current short position on the hedge).

Let β be our matrix per-security factor exposures and C our factor to factor correlation matrix, w our per-security trade-cost vector, v our hedge-trade cost constant and η our per-security un-diversifiable risk vector.[4] As before let x be the vector representing the current position in securities, g the current short on the hedge and $\widehat{y}$, $\widehat{h}$ the cost-insensitive optimal solution.

Our new problem is to find a new security vector y and hedge short position h maximizing:

$$\frac{\mathrm{E}[\mathrm{return}(y,h)] - \mathrm{cost}(y,h,x,g)}{F + \mathrm{undiv}(y,h)} \tag{1}$$

subject to the constraints:

$$\begin{aligned}
\mathrm{E}[\mathrm{return}(y,h)] &= (\alpha + f)^\top (y - h\lambda) && (2)\\
d &\geq y - x && (3)\\
d &\geq x - y && (4)\\
c &\geq h - g && (5)\\
c &\geq g - h && (6)\\
\mathrm{cost}(y,h,x,g) &= w^\top d + vc && (7)\\
F &= (\widehat{y} - \widehat{h}\lambda)^\top \beta C \beta^\top (\widehat{y} - \widehat{h}\lambda) && (8)\\
z &\geq y - h\lambda && (9)\\
z &\geq h\lambda - y && (10)\\
\mathrm{undiv}(y,h) &= \eta^\top z && (11)\\
y &\geq 0 && (12)\\
h &\geq 0 && (13)\\
(\alpha + f)^\top (y - h\lambda) &\geq (1-\epsilon)(\alpha + f)^\top (\widehat{y} - \widehat{h}\lambda) && (14)\\
C\beta^\top (y - h\lambda) &\geq (1-\epsilon) C\beta^\top (\widehat{y} - \widehat{h}\lambda) && (15)\\
C\beta^\top (y - h\lambda) &\leq (1+\epsilon) C\beta^\top (\widehat{y} - \widehat{h}\lambda) && (16)
\end{aligned}$$

We now have what is called a *fractional program* (all our constraints are linear equalities and inequalities, but our objective function is unfortunately a ratio of linear expressions).[Sch76a, FS04, Sch76c, AL71, BN73, KS81, Tan07, JSG08, Din67, Sch76b, Swa65, AS66]

3.5 What to do with a Fractional Program

There are many methods to solve fractional programs (in particular Dinkelbach's Algorithm [Din67, Sch76c] which can handle some quadratic terms). We will work

[4] At this point it is worth mentioning that we are treating costs and un-diversifiable risks somewhat similarly. In fact it makes sense to bias the cost-insensitive solution a little bit by replacing the un-diversifiable risks with a term that includes a bit of the trade costs (since costs and risks are similar negative indications).

through how one might solve a problem such as ours (only linear terms) with access to a simplex based solver (which would allow through updates very fast re-solvings). The idea comes from parametric programming[DM96, VR07] and takes a few conceptual steps.

In the abstract you are trying to maximize an objective function of the form:

$$\frac{f(x)}{g(x)}$$

where $f(x)$, $g(x)$ are linear functions and the whole system is supposed to obey some linear equalities and inequalities (which we will ignore for now).

If we could think of a process that generated a set of vectors $x_1, x_2, \cdots x_k$ (each obeying our constraints) such that the optimal solution is guaranteed to be one of these vectors then we could solve the problem by inspection. We would just plug each of these into our objective function in turn and keep the best.

So it is enough to find a set of vectors to try. To do this notice that (ignoring constraints) that if $\frac{f(x)}{g(x)}$ is maximized then $\nabla_x \frac{f(x)}{g(x)} = 0$. Now

$$\nabla_x \frac{f(x)}{g(x)} = \frac{(\nabla_x f(x))g(x) - f(x)(\nabla_x g(x))}{g(x)^2}.$$

So if we had a complete set of vectors x such that $(\nabla_x f(x))g(x) - f(x)(\nabla_x g(x)) = 0$ then it would be sufficient to inspect just these vectors.

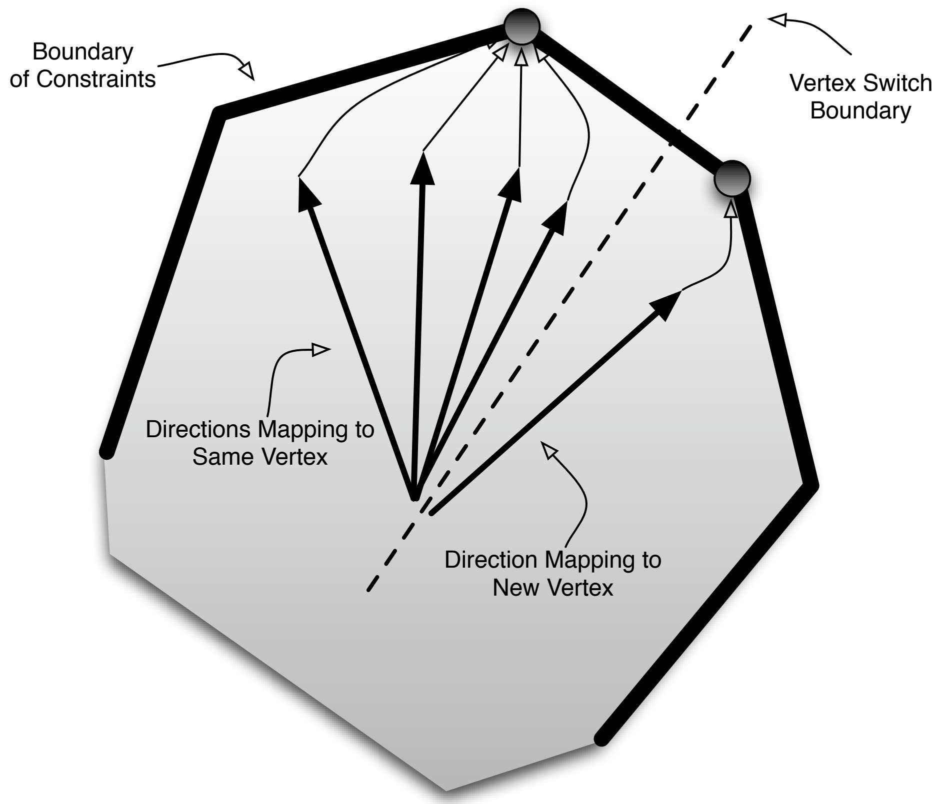
There is another expression that has its maxima in similar circumstances. Consider the expression

$$f(x) - \gamma g(x)$$

where γ is an unknown constant. This expression has its extrema exactly where $(\nabla_x f(x)) - \gamma(\nabla_x g(x)) = 0$ (under reasonable conditions it doesn't have any inflection points, so we can use the extrema to find all the solutions to the above equation). So if we so lucky to pick a γ that was equal to $\frac{f(x)}{g(x)}$ for some optimum we would then recover the vector x.

So it is enough to sweep through all possible values of γ (some of which are sensible, some of which are not) to get a list of candidate x vectors. To do this we want to solve the linear program that maximizes $f(x) - \gamma g(x)$ (subject to our constraints) for many values of our parameter γ. Since γ is a scalar parameter this can be done by a good simplex solver just by updating the objective function and re-solving (finding all solutions should not be too much more expensive than finding a single solution).

What we can do is start at a given value of γ. The linear program can then be solved to find an optimal vertex (see diagram below). The nice thing is for a interval of γ values this same vertex will be the optimal solution and identifying the value where we switch to the next vertex is standard (and trivial) task for a simplex solver. We then repeat asking for the next switch point until there are no more and (after sweeping in both increasing and decreasing directions) we have the list of all vertices that represent candidate x vectors. This whole process and list of candidates can be completed in the time equivalent to solving 2 or 3 linear programs (though this technique is just to illustrate the idea, a fractional programming or parametric programming package would solve this problem directly).



4 Conclusion

We have shown how to encode the search for a nearly optimal trade from given portfolio and estimates of future returns into a very efficient form called a fractional linear program. With commercial libraries such a trade could be implemented very quickly allowing its incorporation in high frequency trading systems.

References

[AL71] Y Almogy and O Levin, *A class of fractional programming problems*, Operations Research **19** (1971), no. 1, 57–67.

[AS66] S P Aggarwal and Kanti Swarup, *Fractional functionals programming with a quadratic constraint*, Operations Research **14** (1966), no. 5, 950–956.

[BN73] G R Bitran and A G Novaes, *Linear programming with a fractional objective function*, Operations Research **21** (1973), no. 1, 22–29.

[Din67] Werner Dinkelbach, *On nonlinear fractional programming*, Management Science **13** (1967), no. 7, 492–298.

[DM96] Nguey Dinh Dan and Le Dung Muu, *A parametric simplex method for optimizing a linear function over the efficient set of a bicriteria linear problem*, Acta Mathematica Vietnamica **21** (1996), no. 1, 59–67.

[FS04] J B G Frenk and Siegfried Schaible, *Fractional programming*, Erasmus Research Institute Of Management (2004), 60.

[HW05] Gur Huberman and Zhenyu Wang, *Arbitrage pricing theory*, The New Palgrave Dictionary of Economics (2005), 19.

[JSG08] Vishwas Deep Joshi, Ekta Singh, and Nilama Gupta, *Primal-dual approach to solve linear fractional programming problem*, Journal of the Applied Mathematics, Statistics and Informatics **4** (2008), no. 1, 61–69.

[KS81] Jonathan S H Kornbluth and Ralph E Steuer, *Multiple objective linear fractional programming*, Management Science **27** (1981), no. 9, 1024–1039.

[Kuh06] Moritz Kuhn, *The karush-kuhn-tucker theorem*, 1–14.

[Lo02] Andrew W Lo, *The statistics of sharpe ratios*, Financial Analysts Journal **58** (2002), no. 4, 18.

[Mou08] John Mount, *A quick appreciation of the sharpe ratio*, `http://www.win-vector.com/blog/2008/09/a-quick-appreciation-of-the-sharpe-ratio/`, 2008.

[RR84] Richard Roll and Stephen A Ross, *The arbitrage pricing theory approach to strategic portfolio planning*, Financial Analysts Journal (1984), 14–26.

[RV02] Andrzej Ruszczynski and Robert J Vanderbei, *Frontiers of stochastically nondominated portfolios*, 1–20.

[Sch76a] Siegfried Schaible, *Duality in fractional programming: A unified approach*, Operations Research **24** (1976), no. 3, 452–461.

[Sch76b] ______, *Fractional programming. i. duality*, Management Science **22** (1976), no. 8, 858–867.

[Sch76c] ______, *Fractional programming. ii, on dinkelbach's algorithm*, Management Science **22** (1976), no. 8, 868–873.

[Swa65] Kanti Swarup, *Linear fractional functionals programming*, Operations Research **13** (1965), no. 6, 1029–1036.

[Tan07] S F Tantawy, *A new method for solving linear fractional programming problems*, Australian Journal of Basic and Applied Sciences **1** (2007), no. 2, 105–108.

[VR07] Robert J Vanderbei and A Ruszczynski, *Parametric linear programming and portfolio optimization*, 1–22.