Optimization 101

Pamela H. Vance, Ph.D.

Managing Director, Product Management

August 15, 2018

Axioma

Flexible is better.®

Optimization 101

What is optimization?

What is the difference between objectives and constraints?

Which objectives and constraints make the optimization hard?

Why are combinatorial constraints so difficult?

When should I use soft constraints or the constraint hierarchy?

What is Optimization?

Optimization problems are **decision-making** problems

- We seek the decision that will maximize or minimize a business goal, called the **objective**, while satisfying business rules or **constraints**

Portfolio Optimization

- **Decision:** how much to hold (or trade) in each individual asset in the investable universe
- **Objective:**
  - Maximize return OR Minimize tracking error OR Maximize return minus risk aversion * variance (MVO)
- **Constraints:**
  - Long only
  - Portfolio budget
  - Limit on turnover
  - Sector/industry/country/currency limits

Should I use an Objective or a Constraint?

Think about the words you use to describe the portfolio characteristics you want

“Must have” or “Most important” characteristics usually belong in constraints.

If you use “as high as possible” or “as low as possible”, this should be your objective.

Should I use an Objective or a Constraint?

An optimization problem has a **single** objective

- The objective can be simple
  - Ex: max return
- Or, it can be a weighted combination of several quantities
  - Ex: max return – λ_1* variance
  - Ex: max return – λ_2* transaction costs
  - Ex: max return – λ_1* variance – λ_2* transaction costs

If more than one quantity is included in the objective, you need to supply ***multipliers*** (λ's) to capture the tradeoffs

Sometimes these tradeoffs are well-understood, but more frequently they are not!

When in doubt, keep the objective simple.

Should I use an Objective or a Constraint?

The objective should measure the quality/desirability of a potential decision (portfolio)

If you are presented with several portfolios that satisfy all your constraints**, how would you choose among them?**

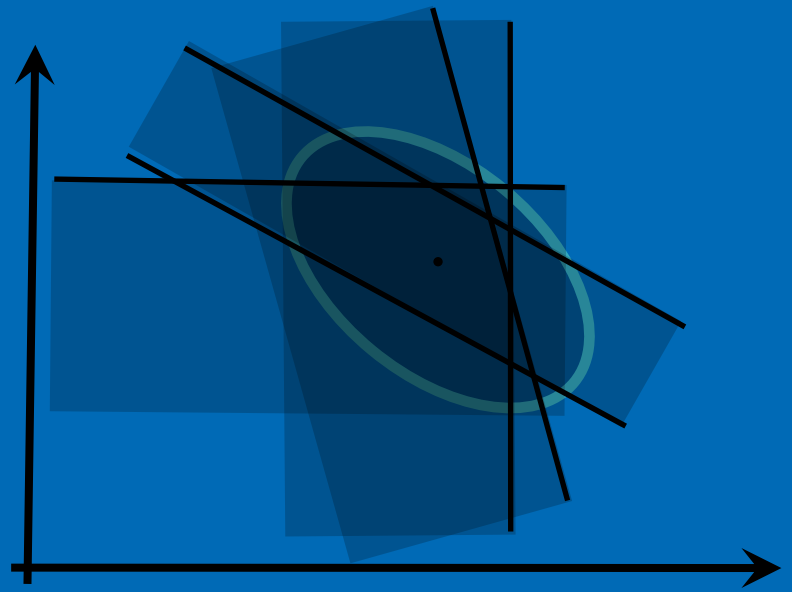
If you can write down a quantitative expression that completely captures the desirability of a portfolio, this should be your objective

- This is rare, but get as close as you can
- We will talk later about handling less-quantitative considerations and “fuzzy” tradeoffs

Constraints: The good news and the bad news

Good News

- An optimization problem can have as many constraints as you like
- Axioma enables you to use virtually any combination of the many constraint types we support



Bad News

- Adding constraints will never improve the objective function value
  - At best they will have no impact on the quality (objective) of the solution
  - At worst they may be impeding the transfer of your alpha signals to your portfolio
- Some types of constraints can make the optimization problem very hard to solve
  - You can see significant increase in solution time after adding “just one more” constraint

Axioma
Flexible is better.®

Should I use an Objective or a Constraint? What should I capture as Constraints?

Hard mandates your portfolio must satisfy (definitely)

- Long-only
- Tracking error limits
- Required portfolio characteristics: beta, dividend yield, …

Not-quite-so-hard rules you want to satisfy (usually)

- Industry/sector/country/ currency exposure limits
- Style factor exposures (tilts)
- Turnover limits
- Limits on the number of names held

Preferences (as soft constraints)

- Lower turnover
- Lower taxes

Linear, Convex, and Combinatorial Elements

Axioma's Optimization Engine can solve optimization problems with linear and non-linear convex elements very efficiently

Linear:

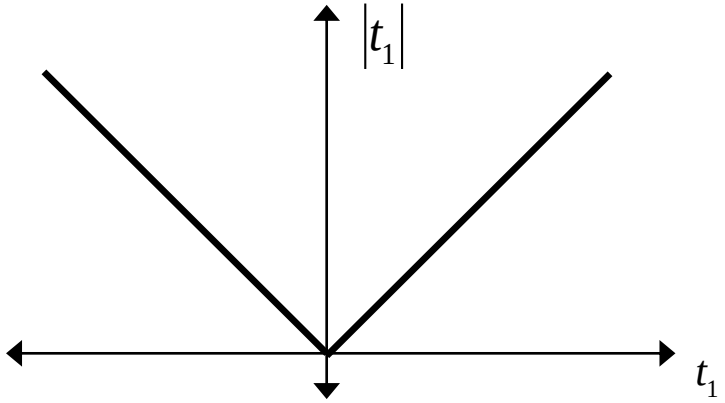
- Final holdings in asset i — h_i
- Expected Return (objective or constr.) — $\alpha_1 h_1 + \alpha_2 h_2 + \alpha_3 h_3 + \ldots = \sum_i \alpha_i h_i = \alpha^T h$
- Limits on individual holdings: — $l_i \le h_i \le u_i$
- Limits on Industry/Sector/Country: — $l \le \sum_{i \in S} h_i \le u$
- Active Holdings: — $l_i \le h_i - b_i \le u_i$
- Industry/Sector/etc Active Holdings: — $l \le \sum_{i \in S} (h_i - b_i) \le u$
- Beta Constraint: — $\beta_l \le \sum_i \beta_i h_i \le \beta_u$
- Limits on Active Duration by Bucket — $l \le \sum_{i \in S} \delta_i (h_i - b_i) \le u$

Linear elements are easy. You can add as many as you like with relatively little impact on solution speed.

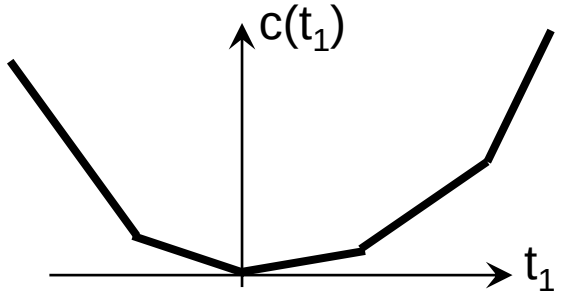
Linear, Convex, and Combinatorial Elements

Non-linear Convex:

- Absolute Values
  - Linear transaction costs $\sum_i c_i|t_i|$
  - Limits on Turnover $\sum_i |t_i| \leq max$
  - Limits on Absolute Holdings $\sum_i |h_i| \leq max$



- Piecewise linear
  - Limits on Transaction Costs



- It is easy to minimize or place a max limit, not easy to maximize or place a minimum limit

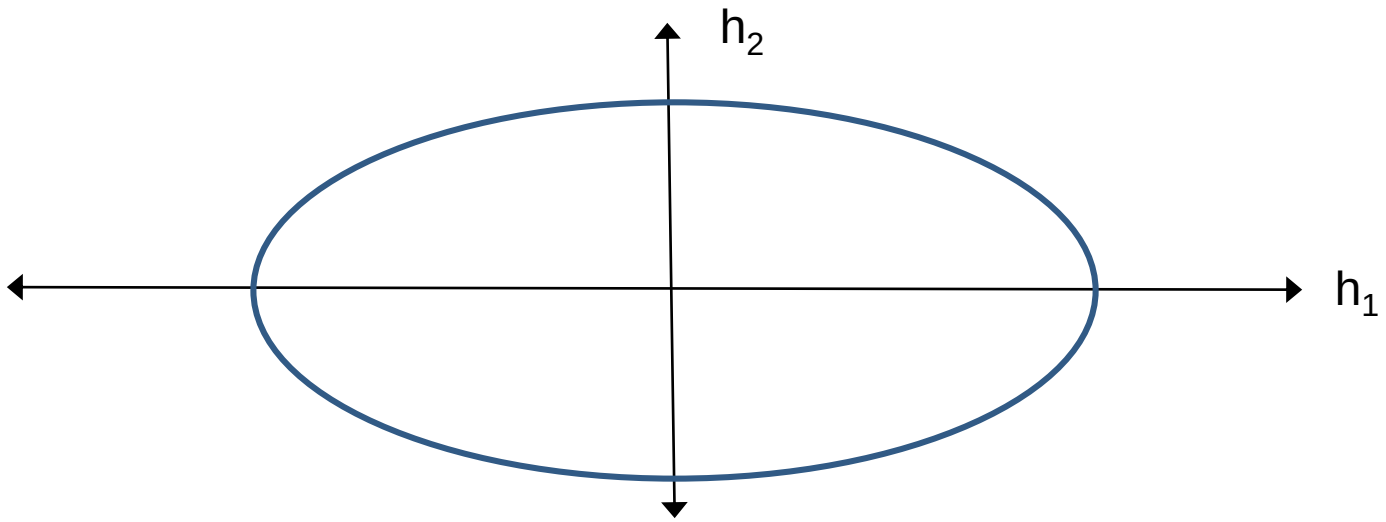
Linear, Convex, and Combinatorial Elements

Non-linear Convex:

- Risk constraints:

$$\sqrt{Var_1 * h_1^2 + Var_2 * h_2^2 + Var_3 * h_3^2 + 2 * Cov_{1,2} * h_1 * h_2 + 2 * Cov_{2,3} * h_2 * h_3 + 2 * Cov_{1,3} * h_1 * h_3} = (h^T Q\, h)^{1/2} \leq c$$

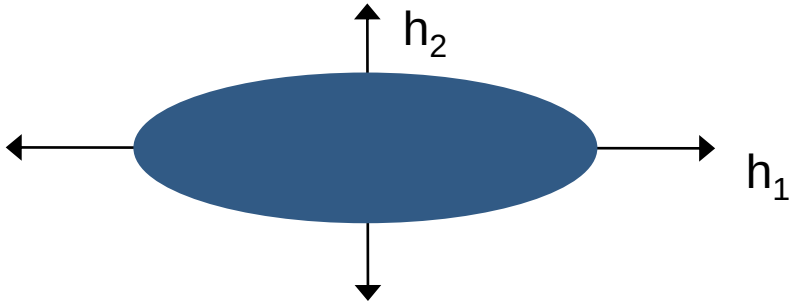
- Active Risk constraints : $((h - b)^T Q\, (h - b))^{1/2} \leq c$



Linear, Convex, and Combinatorial Elements

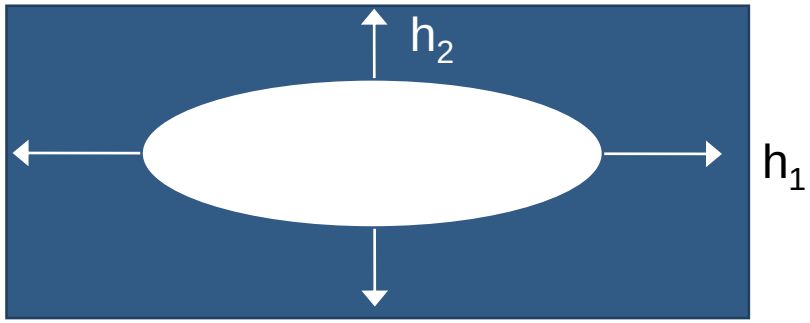
Upper Limit on risk (convex)

- Also risk minimization
  - $((h - b)^T Q (h - b))^{1/2} \leq c$
  - *Min* $((h - b)^T Q (h - b))^{1/2}$



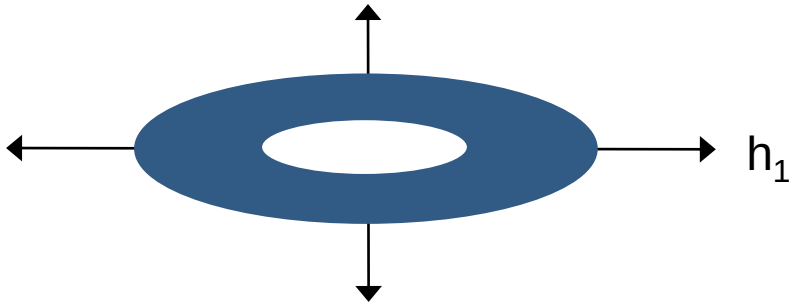
Lower Limit on risk (non-convex)

- Also risk maximization
  - $((h - b)^T Q (h - b))^{1/2} \geq c$
  - *max* $((h - b)^T Q (h - b))^{1/2}$



Upper and lower limits on risk (non-convex)

- $c_1 \leq ((h - b)^T Q (h - b))^{1/2} \leq c_2$



Linear, Convex, and Combinatorial Elements

Non-linear convex:

- Market impact costs
  - Piecewise Linear convex (i.e., transaction cost models) – per unit costs must be non-decreasing with increased volume
    - Quadratic $\sum_{i \in U} \gamma_i t_i^2$
    - Three-halves power $\sum_{i \in U} \gamma_i t_i^{3/2}$
    - Five-thirds power $\sum_{i \in U} \gamma_i t_i^{5/3}$

Non-linear convex elements are also easy. We can solve optimization problems with any combination of non-linear convex and linear elements to optimality.

Linear, Convex, and Combinatorial Elements

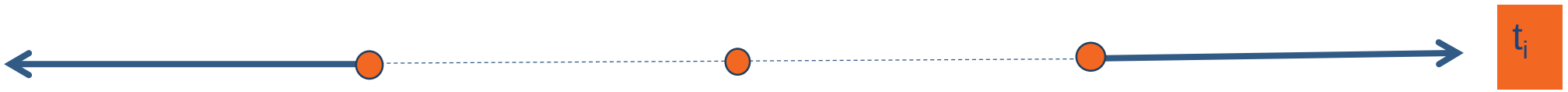
Combinatorial:

- Fixed Charge Transaction Costs (ticket costs)
- Limit number of names held
- Limit number of trades
- Threshold trade & holdings
- Tax Constraints/objectives
- Lower bounds on long and short holdings (under some conditions)

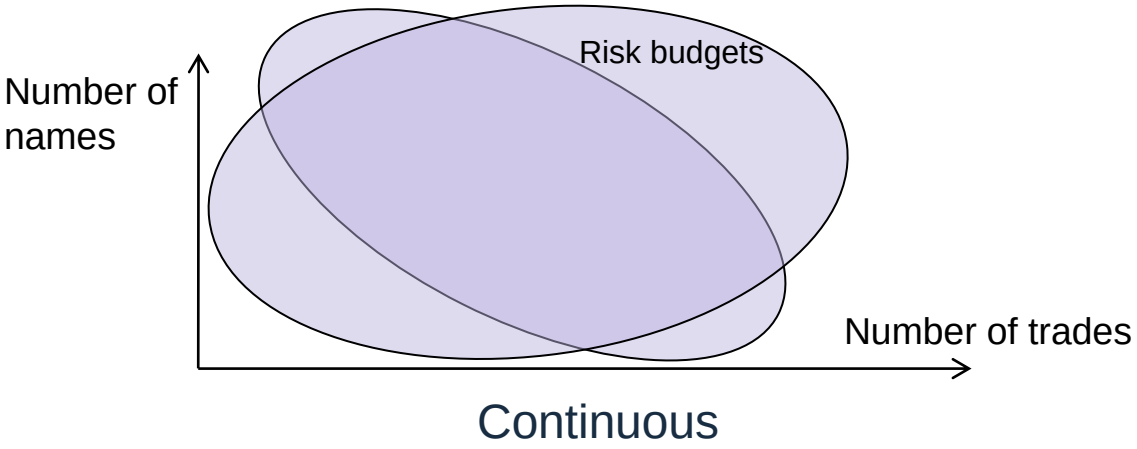
Linear, Convex, and Combinatorial Elements

Combinatorial:

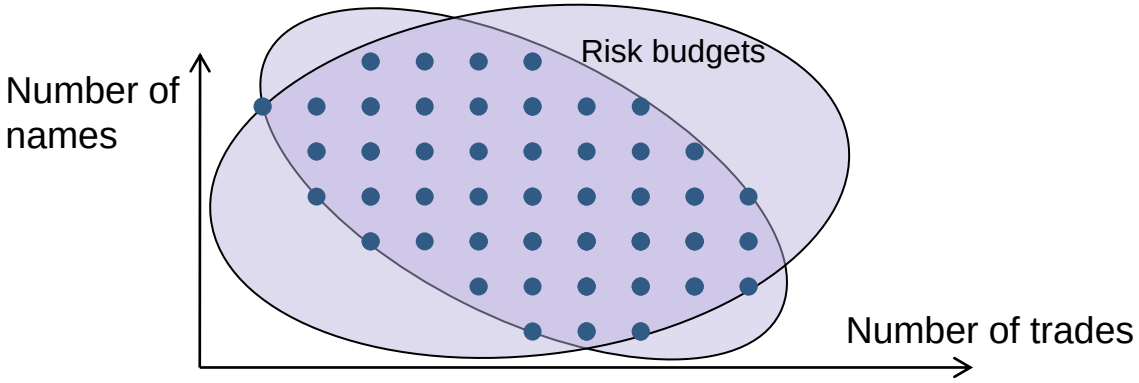
- These restrictions require “0-1” or “discrete” decisions
  - Ex: Limit Names Held
    - Yes/no decision: *Do I hold asset i?*
    - Continuous decision: *How much of asset I do I hold?*
  - Ex: Threshold Trade
    - Yes/no decision: *Do I trade asset i?*
    - Continuous decision: *How much of asset I do I trade?*



Linear, Convex, and Combinatorial Elements



Continuous



Combinatorial

Why do Combinatorial Elements make the optimization problem so difficult?

Optimization problems with linear and non-linear convex elements can be solved to ***optimality*** using a second-order cone programming (**SOCP**) algorithm.

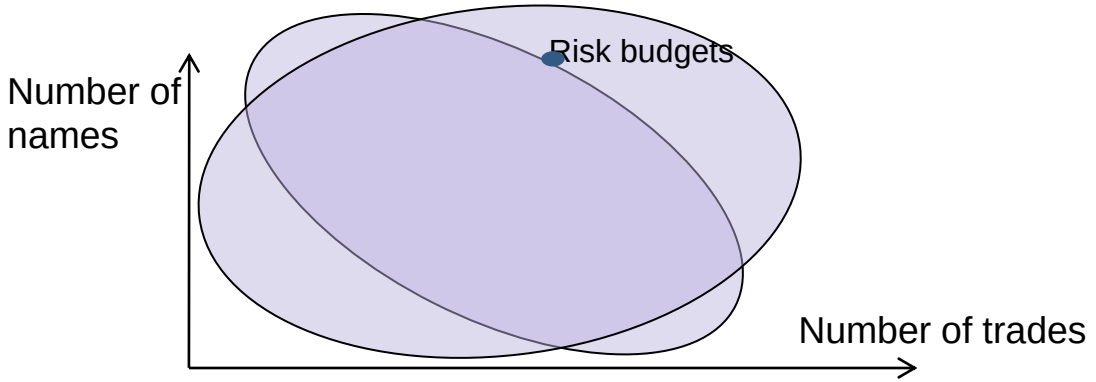
Once combinatorial elements are included, we must resort to a technique called ***branch-and-bound***. In branch-and-bound, we solve ***many*** SOCPs in order to find ***good*** solutions to problems with combinatorial elements.

We cannot generally solve combinatorial problems to true optimality in a practical amount of time.

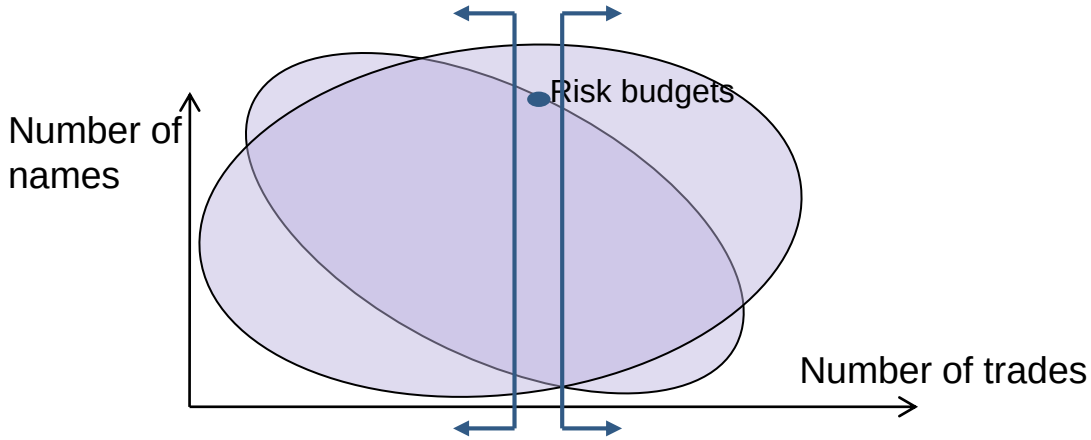
Branch-and-bound

How does branch-and bound work?

- Initially we “relax” the real problem we want to solve by ignoring the combinatorial requirements (i.e., ignore thresholds and numbers of names). We solve the problem with an SOCP solver.
- The objective of the SOCP gives us a “theoretical best bound” on the optimal solution.
- We check to see if the solution to the SOCP satisfies the constraints we ignored. If the restrictions are satisfied we can stop. Otherwise we “branch” and create 2 new SOCP problems.
- The series of SOCP problems we create are represented by a branch-and-bound tree



Relaxed Problem

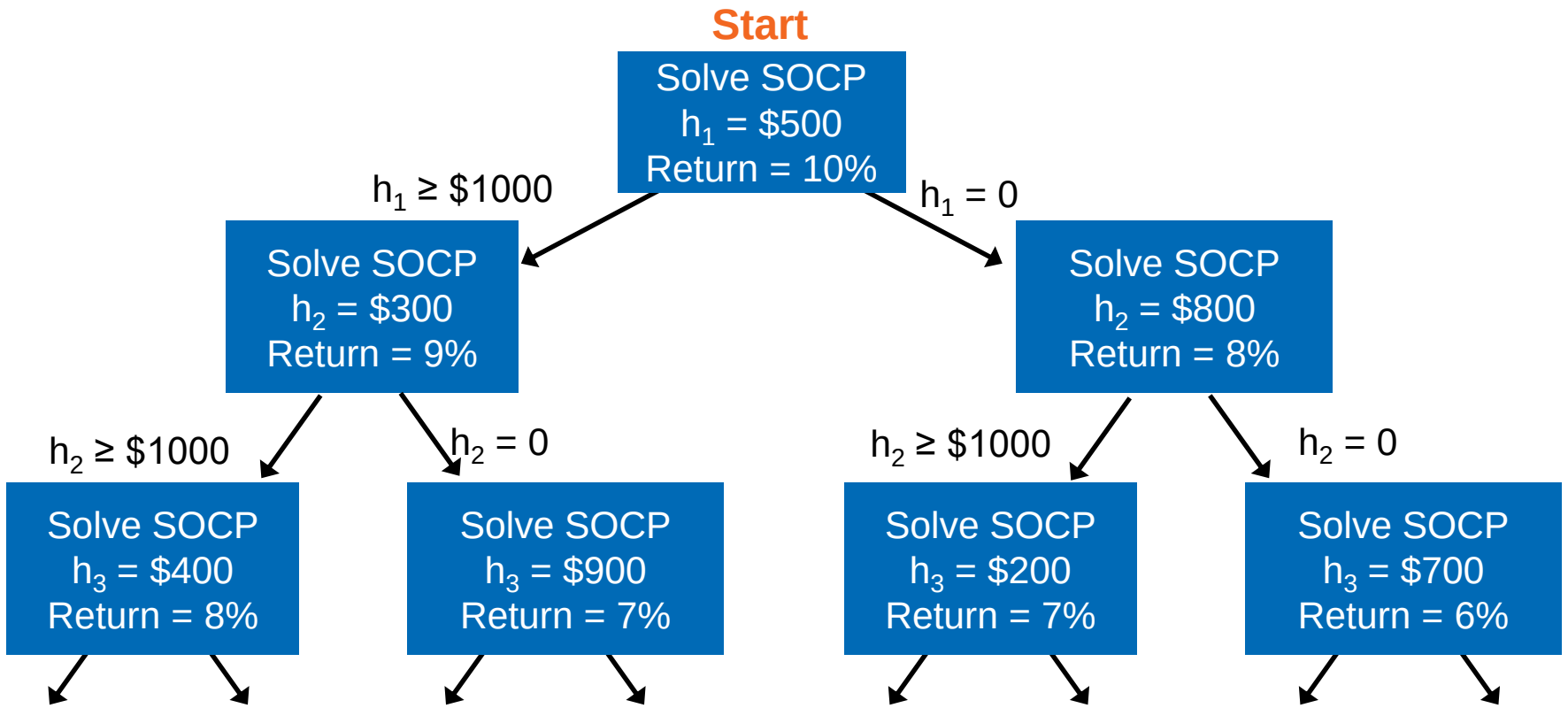


Branching

Branch-and-bound

Ex: Strategy max return with min threshold holding of $1000

First, ignore the threshold constraints, and solve the SOCP



In the worst-case, we may solve 2^n SOCPs (where n is the # of assets in the universe)

Soft Constraints and the Hierarchy

Both can be used to protect against infeasibilities.

- This is particularly important/useful when running backtests.

Soft constraints

- Used extensively with older-generation optimization technology (quadratic programming).
- Can be used to allow violations of the not-so-hard rules you want your portfolio to satisfy
- Can be used to add small incentives to the objective function.
- Can also be used for modeling tricks beyond the scope of Optimization 101
  - Creating a quadratic term in a fixed income optimization problem that lacks a risk model

Soft Constraints and the Hierarchy

Most of the constraints in Axioma Portfolio can be ***softened***. When we soften a constraint, we allow it to be violated by paying a penalty.

Ex: Beta Constraint $\sum_i \beta_i h_i - v \leq \beta_u$

where v is the constraint violation

We add v to the objective function with a penalty value, p

- $p*v$ for linear penalties
- $p*v^2$ for quadratic penalties

Ex: Suppose you were using an MVO objective, now your objective is:

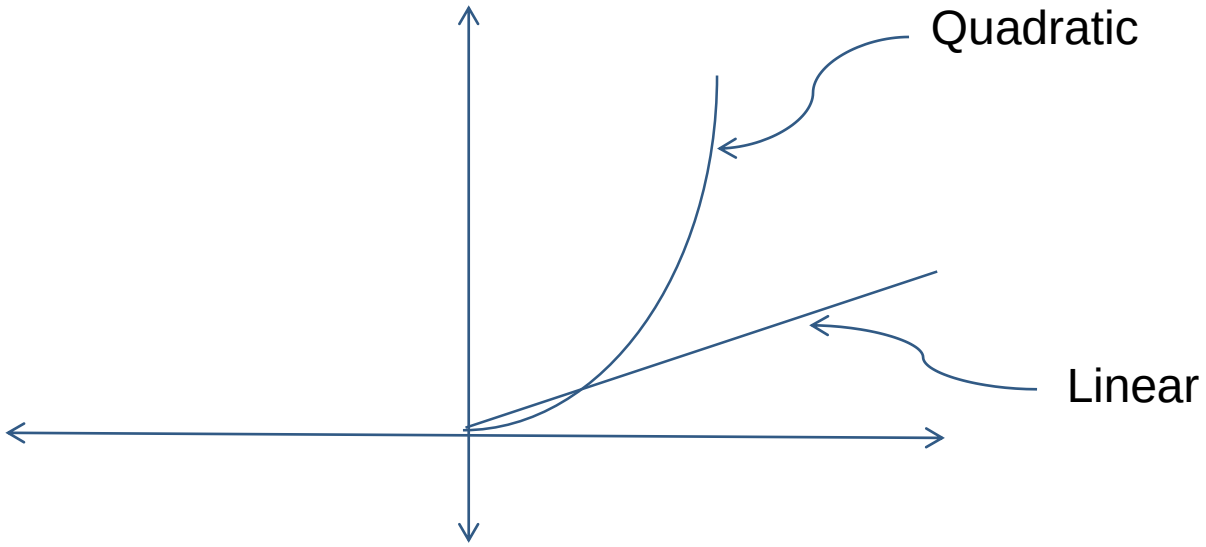
$\alpha h - \lambda * h^T Q\, h - p*v$	*linear penalty*
$\alpha h - \lambda * h^T Q\, h - p*v^2$	*quadratic penalty*

Can you quantify the tradeoffs λ and p among return, risk, and the beta violation? If you are softening a set of constraints (e.g., industries) can you quantify the tradeoffs for each individual industry?

Soft Constraints and the Hierarchy

Linear or Quadratic Penalties?

- The penalty type dictates the form of the additional objective term added by softening the constraint.
- Quadratic penalties escalate more quickly than linear
- When using quadratic penalties in Axioma Portfolio, the **Scaled Quadratic** is recommended. This option automatically scales the penalty using the reference value.



Soft Constraints and the Hierarchy

Creating small incentives with soft constraints

- Most optimization problems have many optimal solutions. If your strategy does not include elements to reflect all your preferences, you may not be finding the solution you most prefer.
  - Ex: Among the solutions with the maximum possible return value, I would prefer the solution with lower turnover.
  - Creating an incentive to select the solution with lower turnover is easily accomplished by including a softened turnover constraint with a low maximum value (0) and a small penalty.
  - The penalty should be small enough so that the turnover penalty contribution is at least a factor of 10 smaller than the main objective term(s) contributions.

Soft Constraints and the Hierarchy

What are the pros and cons of using the Hierarchy rather than soft constraints?

Pro

- The hierarchy will find the solution with the smallest possible violation(s).
- You do not need to calibrate tradeoffs when using the hierarchy. All you need to do is establish the relative importance of your constraints.
- The hierarchy is easier to specify

Con

- The violations created by the hierarchy are not sensitive to the impact on the objective. You have no control over the tradeoff between violation and objective quality.
- Problems with soft constraints can be solved with a single optimization. The hierarchy requires multiple optimizations to solve
- The hierarchy can take longer to solve.

Soft Constraints and the Hierarchy

1

If the optimization cannot be solved with all constraints:

•- Solve the problem with the hard constraints only. If we find a solution:

- Begin adding constraints for each successive level of the hierarchy.

2

When we reach a hierarchy level where the constraints cannot all be satisfied:

- Solve an auxiliary problem to minimize the violation of the constraints at the most-recently-added hierarchy level.

- These are the least important constraints.

3

The auxiliary problem tells us how much we need to relax the newly added constraints.

- Modify the bounds on the newest constraints by the minimum amount required

- Continue to the next level of the hierarchy.

Soft Constraints and the Hierarchy

Some tips on using the hierarchy

- The time it takes to solve the hierarchical problem is proportional to the number of distinct levels in the hierarchy.
- You can assign the same priority (hierarchy level) to more than one constraint. Try to keep the number of hierarchy levels small.
- Combinatorial constraints can complicate the hierarchy because they cannot be softened.

Optimization 101 - Conclusions

When creating your strategy, focus on your primary quality measure to choose your objective. Try to keep your objective simple.

Not all objectives and constraints are created equal!
Use combinatorial elements judiciously.

Soft constraints are very valuable avoiding infeasibilities and creating incentives, but large numbers of soft constraints make calibrating the penalties and interpreting the results difficult.

The constraint hierarchy is an easy-to-specify option for maintaining the feasibility of your strategy. Avoid creating large hierarchies and watch out for constraints that cannot be softened.

Axioma

Flexible is better.®