Computational Method of Optimization

Convex optimization problems - Lecture VII

Convex optimization problems

- Agenda
- generalized inequality constraints
- semidefinite programming
- vector optimization

Source: Convex Optimization — Boyd & Vandenberghe

Generalized inequality constraints

convex problem with generalized inequality constraints

$$\begin{array}{ll} \text{minimize} & f_0(x) \\ \text{subject to} & f_i(x) \preceq_{K_i} 0, \quad i = 1, \ldots, m \\ & Ax = b \end{array}$$

- $f_0 : \mathbf{R}^n \to \mathbf{R}$ convex; $f_i : \mathbf{R}^n \to \mathbf{R}^{k_i}$ K_i-convex w.r.t. proper cone K_i
- same properties as standard convex problem (convex feasible set, local optimum is global, etc.)

conic form problem: special case with affine objective and constraints

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & Fx + g \preceq_K 0 \\ & Ax = b \end{array}$$

extends linear programming ($K = \mathbf{R}^m_+$) to nonpolyhedral cones

Here we're to generalize the objective.
But right now, we're going to generalize the inequalities to be vector inequalities. So here it was $f_i(x)$ is less than or equal to zero, we have f_i of x return a vector, it's defined by this cone K_i. So in this case if f_i is K_i convex so it's a convex function with respect to this cone, this generalize and equality, then this is called convex problem with generalize and equality constraints.

It's a conic form problem and a special case. Everything is affine so it's equivalent of a linear program basically. A normal optimization problem, if every function is an affine you have a linear program because you minimize and affine function, subject to affine functions less than zero because you're conventionally called linear and inequalities. Affine equality, that's conventionally called linear equality constraints.

Semidefinite program (SDP)

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & x_1 F_1 + x_2 F_2 + \cdots + x_n F_n + G \preceq 0 \\ & Ax = b \end{array}$$

with $F_i, G \in \mathbf{S}^k$

- inequality constraint is called linear matrix inequality (LMI)
- includes problems with multiple LMI constraints: for example,

$$x_1 \hat{F}_1 + \cdots + x_n \hat{F}_n + \hat{G} \preceq 0, \qquad x_1 \tilde{F}_1 + \cdots + x_n \tilde{F}_n + \tilde{G} \preceq 0$$

is equivalent to single LMI

$$x_1 \begin{bmatrix} \hat{F}_1 & 0 \\ 0 & \tilde{F}_1 \end{bmatrix} + x_2 \begin{bmatrix} \hat{F}_2 & 0 \\ 0 & \tilde{F}_2 \end{bmatrix} + \cdots + x_n \begin{bmatrix} \hat{F}_n & 0 \\ 0 & \tilde{F}_n \end{bmatrix} + \begin{bmatrix} \hat{G} & 0 \\ 0 & \tilde{G} \end{bmatrix} \preceq 0$$

Semi-definite programming or SDP, this is relatively new. It looks like a linear program except for one generalization. So we minimize linear function, linear equality constraints – that's so far linear programming. Here's the difference. There's an inequality and it's a matrix inequality. These are symmetric matrix's here, called an LMI, or linear matrix inequality.

So this is an LMI. Here as drawn there's only one LMI, but of course, you could have multiple LMIs, if you have a bunch of LMIs like this, if you have two, you can lump it together into one and you lump it together into one by simply making a block diagonal matrix here with the LMIs because if I have a block matrix and block matrix is negative semi-definite. So that's why we can, without lose of generality, work with just one LMI in general.

LP and SOCP as SDP

LP and equivalent SDP

LP: minimize $c^T x$
subject to $Ax \preceq b$

SDP: minimize $c^T x$
subject to $\mathbf{diag}(Ax - b) \preceq 0$

(note different interpretation of generalized inequality $\preceq$)

SOCP and equivalent SDP

SOCP: minimize $f^T x$
subject to $\|A_i x + b_i\|_2 \leq c_i^T x + d_i, \quad i = 1, \ldots, m$

SDP: minimize $f^T x$
subject to $\begin{bmatrix} (c_i^T x + d_i)I & A_i x + b_i \\ (A_i x + b_i)^T & c_i^T x + d_i \end{bmatrix} \succeq 0, \quad i = 1, \ldots, m$

Semi-definite programming or SDP looks like a linear program except for one generalization. So we minimize linear function, linear equality constraints – that's so far linear programming. Here's the difference. There's an inequality and it's a matrix inequality. These are symmetric matrix's here. So this thing here is called an LMI. A matrix inequality is very complicated. To say that a matrix is positive semi-definite or negative semi-definite.

Here's a more interesting embedding. SOCP, so second order cone programming, the question is how do you represent this as LMI? this is non linear and it's non differentiable and this later one. This is a matrix here, which has a block – this one one block of the matrix is a scaler times the identity. It's positive definite if and only if the diagonal entities are positive.

LP and SOCP as SDP

LP and equivalent SDP

LP: minimize $c^T x$
subject to $Ax \preceq b$

SDP: minimize $c^T x$
subject to $\mathbf{diag}(Ax - b) \preceq 0$

(note different interpretation of generalized inequality $\preceq$)

SOCP and equivalent SDP

SOCP: minimize $f^T x$
subject to $\|A_i x + b_i\|_2 \leq c_i^T x + d_i, \quad i = 1, \ldots, m$

SDP: minimize $f^T x$
subject to $\begin{bmatrix} (c_i^T x + d_i)I & A_i x + b_i \\ (A_i x + b_i)^T & c_i^T x + d_i \end{bmatrix} \succeq 0, \quad i = 1, \ldots, m$

Now, for this matrix, that says $c_i^T x + d_i$ like this – let me put these together. Maybe if you do it on the pad here and capture both the big LM on this one and that one. There we go. If you can just capture this and what I'm writing here. It's this must be bigger than or equal to this thing $(A_i x + b)^T$.

It's positive definite if and only if the diagonal entities are positive. We write

$c_i^T x + d_i \geq (A_i x + b)^T \frac{1}{c_i^T x + di} (A_i x + b)$

$\Rightarrow (c_i^T x + d_i)^2 \geq (A_i x + b)^T (A_i x + b)$

We'll get exactly this second order cone constrained

Eigenvalue minimization

$$\text{minimize} \quad \lambda_{\max}(A(x))$$

where $A(x) = A_0 + x_1 A_1 + \cdots + x_n A_n$ (with given $A_i \in \mathbf{S}^k$)

equivalent SDP

$$\begin{array}{ll} \text{minimize} & t \\ \text{subject to} & A(x) \preceq tI \end{array}$$

- variables $x \in \mathbf{R}^n$, $t \in \mathbf{R}$
- follows from

$$\lambda_{\max}(A) \leq t \quad \Longleftrightarrow \quad A \preceq tI$$

It's eigenvalue minimizations. So here, I have a matrix which is an affine function of a vector x, so an affine matrix valued function looks like this. It's a constant matrix, these are all symmetric, plus and then a linear combination of symmetric matrix's here and I want to minimize the maximum eigenvalue of that matrix.

That's a complicated problem, because you take this matrix – so far, it's fine, every entry of this matrix is an affine function of x, but now to get the maximum eigenvalue value you form the characteristic polynomial.

Now, on the other hand, we can optimize the maximum eigenvalue very easily as an SDP. You minimize t subject to A of x is less than or equal to tI.. And this is affine in the variables s and t. So that's eigenvalue. So eigenvalue minimization is quite straight forward.

Matrix norm minimization

$$\text{minimize} \quad \|A(x)\|_2 = \left(\lambda_{\max}(A(x)^T A(x))\right)^{1/2}$$

where $A(x) = A_0 + x_1 A_1 + \cdots + x_n A_n$ (with given $A_i \in \mathbf{R}^{p\times q}$)

equivalent SDP

$$\begin{array}{ll} \text{minimize} & t \\ \text{subject to} & \begin{bmatrix} tI & A(x) \\ A(x)^T & tI \end{bmatrix} \succeq 0 \end{array}$$

- variables $x \in \mathbf{R}^n$, $t \in \mathbf{R}$
- constraint follows from

$$\begin{aligned} \|A\|_2 \leq t \quad &\iff \quad A^T A \preceq t^2 I, \quad t \geq 0 \\ &\iff \quad \begin{bmatrix} tI & A \\ A^T & tI \end{bmatrix} \succeq 0 \end{aligned}$$

Matrix norm, is a complicated function. It's the square root of the largest eigenvalue of A^TA. We know it's sort of SDP representable. This is quadratic and in an SDP, which after all has the constraint of a linear matrix inequality.

$tI \geq 0$ and $tI \geq \frac{A(x)^T A(x)}{tI} \Rightarrow t^2 I \geq A(x)^T A(x)$

That means the maximum eigenvalue of this matrix is less than t^2, but the maximum eigenvalue of $A(x)^T A(x)$ is the maximum singular value of A^2 and you take square roots and you're back to this thing. So that's the embedding.

Vector optimization

general vector optimization problem

$$\begin{array}{ll} \text{minimize (w.r.t. } K) & f_0(x) \\ \text{subject to} & f_i(x) \leq 0, \quad i = 1, \ldots, m \\ & h_i(x) = 0, \quad i = 1, \ldots, p \end{array}$$

vector objective $f_0 : \mathbf{R}^n \to \mathbf{R}^q$, minimized w.r.t. proper cone $K \in \mathbf{R}^q$

convex vector optimization problem

$$\begin{array}{ll} \text{minimize (w.r.t. } K) & f_0(x) \\ \text{subject to} & f_i(x) \leq 0, \quad i = 1, \ldots, m \\ & Ax = b \end{array}$$

with f_0 K-convex, $f_1, \ldots, f_m$ convex

We are going to generalize the objective to be a vector. So, so far, we left the objective as a scaler, and we generalized the inequalities to be vectors and you get things like cone programming.

What is f_0 of x is a vector with two things?
What does it mean to say minimize a vector?
it's important to get all the semantics right.
So, a convex vector optimization is just like this except f_0 of x is a vector and there's a cone associated with it. Let's do the scaler case, the semantics of scaler objective optimization. If the semantics are extremely simple, and it all comes down to the following

Vector optimization

general vector optimization problem

$$\begin{array}{ll} \text{minimize (w.r.t. } K) & f_0(x) \\ \text{subject to} & f_i(x) \leq 0, \quad i = 1, \ldots, m \\ & h_i(x) = 0, \quad i = 1, \ldots, p \end{array}$$

vector objective $f_0 : \mathbf{R}^n \to \mathbf{R}^q$, minimized w.r.t. proper cone $K \in \mathbf{R}^q$

convex vector optimization problem

$$\begin{array}{ll} \text{minimize (w.r.t. } K) & f_0(x) \\ \text{subject to} & f_i(x) \leq 0, \quad i = 1, \ldots, m \\ & Ax = b \end{array}$$

with f_0 K-convex, $f_1, \ldots, f_m$ convex

If you have three people come up with a proposed x–you need to know about the semantics of scaler optimization. I look at the first x and I check the constraints. If any constraint is violated in the slightest I dismiss that x and it means it's not feasible. Now, the remaining two people both pass the feasibility test. They're x's are both feasible. They need the constraints.

I evaluate the objective function for those two points. If one is less than the other, that's better and I dismiss the other one because here I have two potential points, each of which satisfies the hard constraints. They're not both optimal. The answer is as far as that problem is concerned, you don't even have a right to have a preference between one or the other.

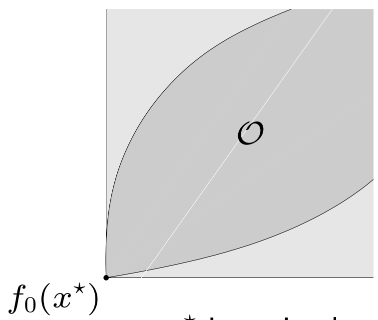
So, the point is that they're all comparable. Now, suppose they're vectors. Now they're vectors and as you know with vector inequalities they're not linear orders. Linear ordering is one in which any two elements can be compared. There are linear vector orderings.

Optimal and Pareto optimal points

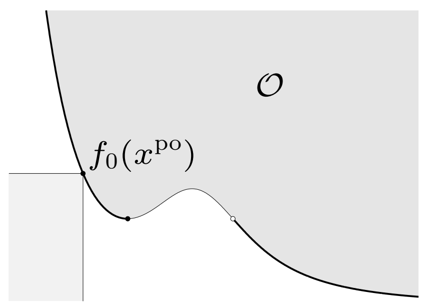
set of achievable objective values

$$\mathcal{O} = \{f_0(x) \mid x \text{ feasible}\}$$

- feasible x is **optimal** if $f_0(x)$ is the minimum value of $\mathcal{O}$
- feasible x is **Pareto optimal** if $f_0(x)$ is a minimal value of $\mathcal{O}$



$x^\star$ is optimal



x^{po} is Pareto optimal

You look at the set of achievable values. That's the set of objective values you can possibly get with a feasible x. You say the point is optimal if that achieved objective value is minimal in O. You say it's pareto optimal if it's a minimal value.

A minimum point of a set is one where every other point in the set is comparable to it and larger. So this is a minimum point. If this is the set of achievable objectives, this is a minimum point. This is the only case, essentially, when you can say unambiguously this point is optimal. It's the only time you can say minimize something and say x solves this because no one could possible argument with it here in this case. Minimum means you're better than every other point. Minimal means no point is better than this

Multi-criterion optimization

vector optimization problem with $K = \mathbf{R}^q_+$

$$f_0(x) = (F_1(x), \ldots, F_q(x))$$

- q different objectives F_i; roughly speaking we want all F_i's to be small
- feasible $x^\star$ is optimal if

$$y \text{ feasible} \quad \Longrightarrow \quad f_0(x^\star) \preceq f_0(y)$$

  if there exists an optimal point, the objectives are noncompeting
- feasible $x^{\rm po}$ is Pareto optimal if

$$y \text{ feasible}, \quad f_0(y) \preceq f_0(x^{\rm po}) \quad \Longrightarrow \quad f_0(x^{\rm po}) = f_0(y)$$

  if there are multiple Pareto optimal values, there is a trade-off between the objectives

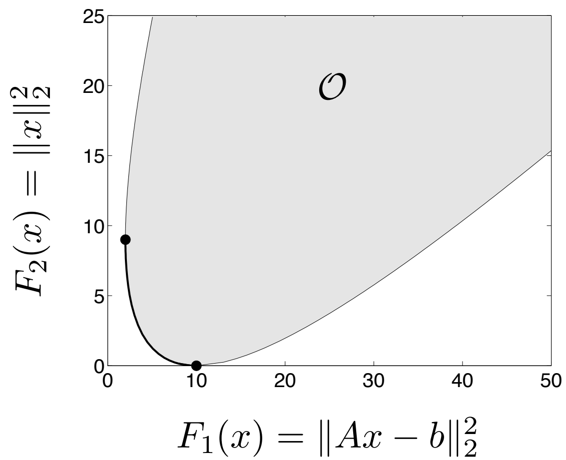
In a multi- criterion problem, you have multiple objectives and if you want to think about something specific here you might think of an engineering design, let's say a circuit and one of these would be something like power, one would be delay, one would be area, just for example.

One could be how close you're tracking. The point is there are multiple objectives all of which you want small. Once you think about it you realize everything has this form.

You want all of these small. If someone comes up with a point and if you meet or beat them on all objectives and beat them on one, you're better. So pareto optimal means no one beats you. That's pareto optimal. It means that if you have two pareto optimal things, you have to have traded something off.

Regularized least-squares

minimize (w.r.t. $\mathbf{R}^2_+$) $\quad (\|Ax - b\|_2^2, \|x\|_2^2)$



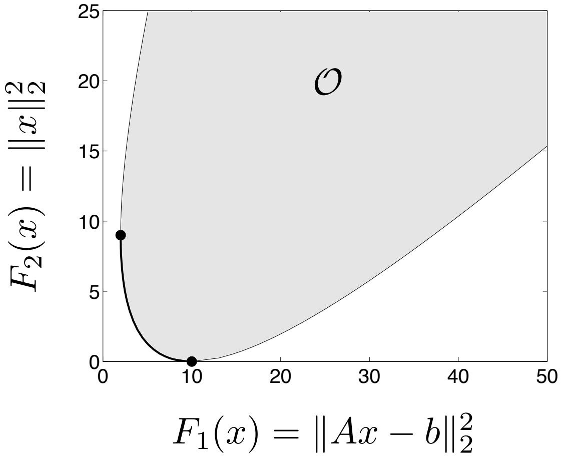
example for $A \in \mathbf{R}^{100\times 10}$; heavy line is formed by Pareto optimal points

Minimize with respect to R_+^2. You minimize with respect to the second order quadratic the following: two objects, both by the way convex here. Norm Ax-b squared, that's like a matching criterion. That's how well do you map something. But at the same time, please do it with a small x.

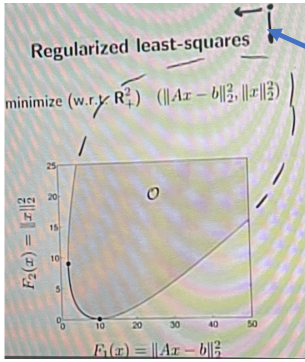
x by the way could be a vector in 10,000-dimensional space, but there's only two objectives, what you do is you check every vector in 10,000 so you plug in every possible x and you get a score, which is a pair of numbers, non-negative numbers and you put a dot at every single point. This is conceptual. You now do this for all 10,000 points and you get this shaded plot that looks like that. In fact, it's a hyperbole or something.

Regularized least-squares

minimize (w.r.t. $\mathbf{R}^2_+$) $\quad (\|Ax - b\|_2^2, \|x\|_2^2)$



example for $A \in \mathbf{R}^{100\times 10}$; heavy line is formed by Pareto optimal points



Now, also I should add that O has some interesting structure out here. What's interesting for us is down and to the left. That's the only thing interesting to us.

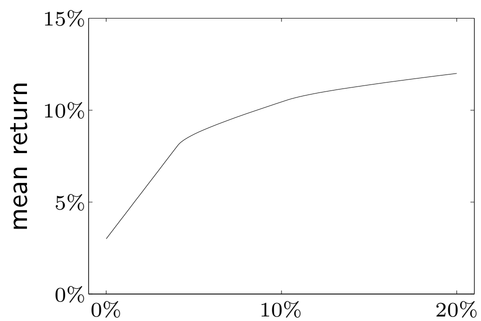
So we're only interested in what's down and left. So in actuality the only thing that's interesting here is what's shaded here is the pareto optimal curve. This is called the regulization curve.

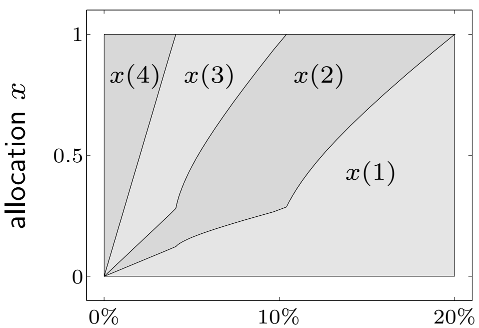
Risk return trade-off in portfolio optimization

$$\begin{array}{ll} \text{minimize (w.r.t. } \mathbf{R}^2_+) & (-\bar{p}^T x, x^T \Sigma x) \\ \text{subject to} & \mathbf{1}^T x = 1, \quad x \succeq 0 \end{array}$$

- $x \in \mathbf{R}^n$ is investment portfolio; x_i is fraction invested in asset i
- $p \in \mathbf{R}^n$ is vector of relative asset price changes; modeled as a random variable with mean $\bar{p}$, covariance Σ
- $\bar{p}^T x = \mathbf{E}\, r$ is expected return; $x^T \Sigma x = \mathbf{var}\, r$ is return variance

example





Here's a generic example. I have two objectives and let me explain the problem here. I have a vector that represents an investment portfolio so x_i, for example, might be the fraction – people set this is up differently, but a classical one is this – x_i represents the fraction invested in asset i. So you have n assets, negative – if you want to have negative, I mean, here we've ruled it out, but the point is if you want to have negative it means you have a sure position.

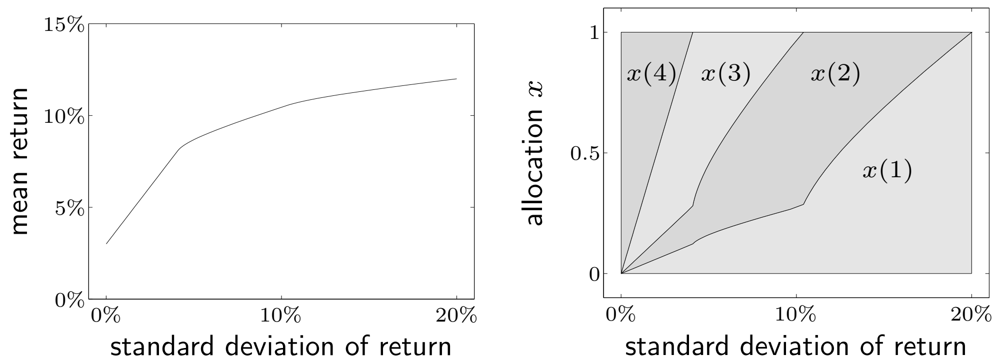
If that asset is cash, it means you're margining. You're borrowing money to finance the purpose of other assets. But simple we just made x a positive and you'd say that there are no short positions. No margining. That's the fraction. And as a fraction, it's got to add up to one.

Risk return trade-off in portfolio optimization

$$\begin{array}{ll} \text{minimize (w.r.t. } \mathbf{R}^2_+) & (-\bar{p}^T x, x^T \Sigma x) \\ \text{subject to} & \mathbf{1}^T x = 1, \quad x \succeq 0 \end{array}$$

- $x \in \mathbf{R}^n$ is investment portfolio; x_i is fraction invested in asset i
- $p \in \mathbf{R}^n$ is vector of relative asset price changes; modeled as a random variable with mean $\bar{p}$, covariance Σ
- $\bar{p}^T x = \mathbf{E}\, r$ is expected return; $x^T \Sigma x = \mathbf{var}\, r$ is return variance

example



Now, what happens here is you invest in these assets as one period and what happens is there's a return or the prices change, so you can call p. So if you form $p^T x$ that gives you the total return on your portfolio. Now, obviously, if the returns were known what's the best possible portfolio?

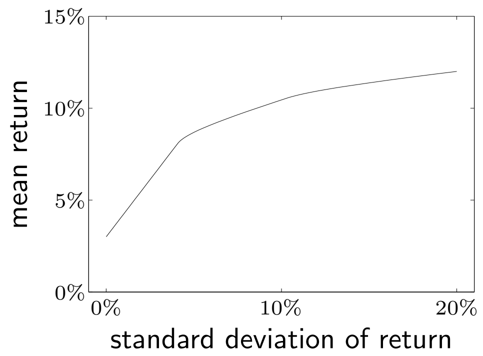
What the returns are, you simply put Er, where r is an index corresponding to whichever one had the highest return. It's a random variable. So the return is a random variable. So p is a random variable and $p^T x$, that's the total investment return, is a random variable.

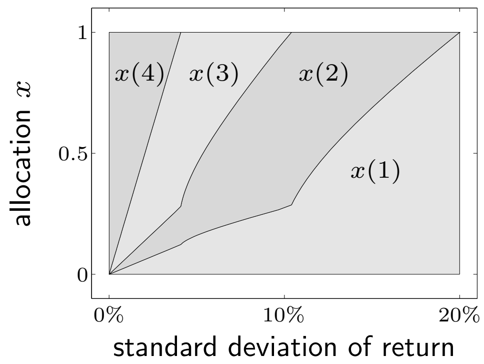
Risk return trade-off in portfolio optimization

$$\begin{array}{ll} \text{minimize (w.r.t. } \mathbf{R}^2_+) & (-\bar{p}^T x, x^T \Sigma x) \\ \text{subject to} & \mathbf{1}^T x = 1, \quad x \succeq 0 \end{array}$$

- $x \in \mathbf{R}^n$ is investment portfolio; x_i is fraction invested in asset i
- $p \in \mathbf{R}^n$ is vector of relative asset price changes; modeled as a random variable with mean $\bar{p}$, covariance Σ
- $\bar{p}^T x = \mathbf{E}\, r$ is expected return; $x^T \Sigma x = \mathbf{var}\, r$ is return variance

example





you evaluate the expected return and the variance, or you could take the square to the variance and get standard deviation. That's sort of the minimum that makes any sense at all would be this pair. And now you have a bi-objective problem because everyone wants high-expected return and everyone will take a lower risk over a higher one.

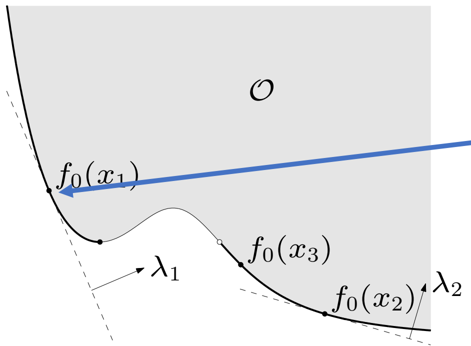
Now is you plot these things by two things; you'd look at the risk as measured by standard deviation of return . The simplest measure is just variance.

Scalarization

to find Pareto optimal points: choose $\lambda \succ_{K^*} 0$ and solve scalar problem

$$\begin{array}{ll} \text{minimize} & \lambda^T f_0(x) \\ \text{subject to} & f_i(x) \le 0, \quad i = 1, \dots, m \\ & h_i(x) = 0, \quad i = 1, \dots, p \end{array}$$

if x is optimal for scalar problem, then it is Pareto-optimal for vector optimization problem



for convex vector optimization problems, can find (almost) all Pareto optimal points by varying $\lambda \succ_{K^*} 0$

$\lambda^T f_0(x)$ is convex and a scaler function. In particular, this is a scaler convex optimization problem. In the case of multiple criterion, we'll see it's something very simple. It's a positive way to sum up the objectives.

So here in this problem if you take a λ like this and you minimize this, it basically says you go as far as you can in this direction, and you might find that point right there. That's pareto optimal

Now, by wiggling λ you're changing the slope and it says for example you can get this point. Now, what's very interesting here is that this point right here is pareto optimal, but you cannot get it by any choice of λ. If you want to wiggle around on the pareto optimal design, you change the weights and it's kind of obvious how you change the weights.