Multi-Objective Optimization

Optimal Taxation

Kenneth Judd
Philipp Müller

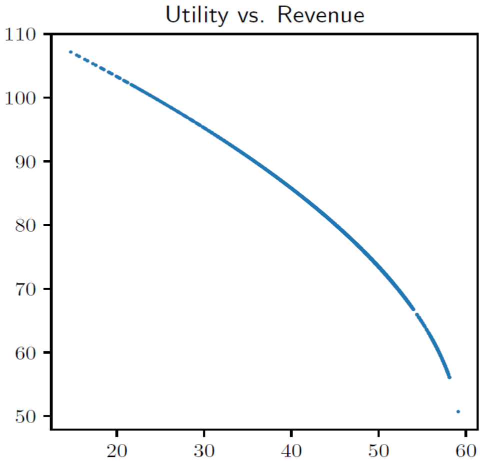
April 27, 2020

Motivation

Motivation

- Optimal taxation faces different objectives. The social planner has to balance
  - revenue
  - average utility of tax payers
  - impact on distribution
  - utility of important people
- Many other problems face similar tradeoffs between conflicting objectives. In engineering, e.g., this includes airplane wing design where we optimize the
  - subsonic performance and
  - transsonic performance.

Motivation



Multi-Objective Optimization

We consider a multi-objective optimization problem of the form

$$\begin{aligned} &\text{"}\min_{Y}\text{"}\{f_1(Y), f_2(Y), \dots, f_k(Y)\} \\ &\text{subject to } Y \in S \end{aligned} \tag{1}$$

with k objective functions $f_i : \mathbb{R}^N \to \mathbb{R}$ and the *design vector* Y.
S denotes the feasible set with

$$S \equiv \{Y \in \mathbb{R}^N : g(Y) \leq 0, h(Y) = 0\}, \tag{2}$$

with $h(\cdot)$ denoting the equality and $g(\cdot)$ denoting the inequality constraints.

Pareto Defintions

Definition (Pareto Optimal Points)

A design-point Y_1 dominates the design-point Y_2, if

$$f_i(Y_1) \leq f_i(Y_2), \forall i = 1, \ldots, k,$$

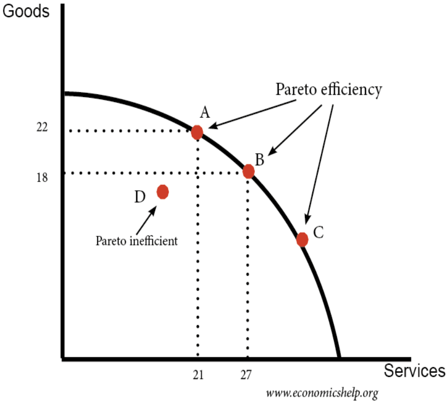
with at least one inequality being strict.

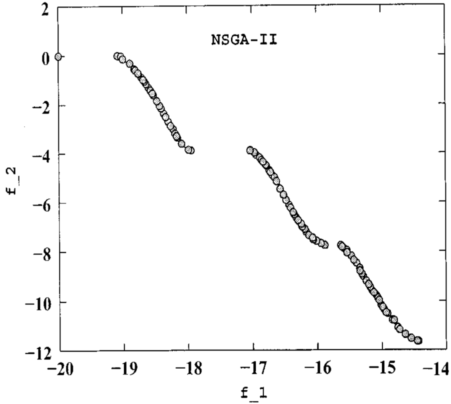
I.e., a design point Y_1 is pareto optimal if there exist no feasible point Y_2 which improves any objective.

Definition (Pareto frontier)

Set of non-dominated design points.

Pareto Front





Recall: Gradient Descent for Single-Objective Optimization

Suppose we minimize the unconstrained single-objective problem

$$\min f(x)$$

with $f : \mathbb{R}^n \to \mathbb{R}^1$.

1. Calculate the steepest descent direction
$$d = -\frac{\nabla f(x)}{||\nabla f(x)||}.$$
2. Find the optimal steplength λ by applying a (inexact) line-search.
3. Update the point $x^{i+1} = x + \lambda d$; restart from step 1 until the optimum is reached, i.e., $\nabla f(x) \approx 0$.

Multi-objective Optimization

We exemplary consider the unconstrained multi-objective optimization problem

$$\min_{Y} \{f_1(Y), f_2(Y), \ldots, f_k(Y)\} \tag{3}$$

with k objective functions $f_i : \mathbb{R}^N \to \mathbb{R}$ and the *design vector* Y.

Multi-objective Optimization

We exemplary consider the unconstrained multi-objective optimization problem

$$\min_{Y}\{f_1(Y), f_2(Y), \ldots, f_k(Y)\} \tag{3}$$

with k objective functions $f_i : \mathbb{R}^N \to \mathbb{R}$ and the *design vector* Y.

(In-)equality constraints can be incorporated into the approach, but we focus on unconstrained optimization for simplicity.

Multiple-Gradient Descent Algorithm

1. Calculate the **common descent direction** ω,
2. Find the optimal steplength λ by applying a (inexact) line-search.
3. Update the point $Y^{i+1} = Y^0 + h\omega$; restart from step 1 until *the optimum is reached*.

However, the calculation of the **common descent direction** $\boldsymbol{\omega}$ and the **stopping condition** require more careful attention as in the 1D case.

Note: This requires the objectives to be differentiable!

Common Descent Direction ω

- To determine the common descent direction ω, we calculate the objective functions gradients

$$u_i = \nabla_x f_i(Y) \quad i = 1, \ldots, k \tag{4}$$

- You can calculate the gradients by finite differences, however, by now we know “better”.

Goal Starting from design point Y^i, does there exist a nonzero descent direction vector $\omega \in \mathbb{R}^N$ which improves **at least one objective** while **not worsening the other objectives**.

Calculate Common Descent Direction

- We define ω to be the minimal norm element solution in the convex hull to

$$\min_{\alpha \in \mathbb{R}^k} \|\sum_{i=1}^{n} \alpha_i u_i\|$$
$$\text{s.t. } \alpha_i \geq 0, \sum \alpha_i = 1,$$

i.e.,

$$\omega = \sum_i \alpha_i u_i.$$

We are Lucky!

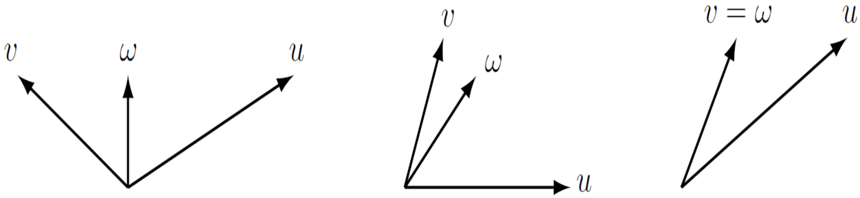
- For the 2 objective case, there exists a closed-form solution.
- The common descent direction reads

$$\omega = \alpha u_1 + (1 - \alpha) u_2$$

with

$$\alpha = \begin{cases} \frac{u_2 \cdot (u_2 - u_1)}{||u_1 - u_2||^2} & \text{if } u_1 \cdot u_2 < \min\left(||u||, ||v||\right)^2 \\ 0 & \text{ortherwise.} \end{cases} \tag{5}$$

Common Descent Direction



Optimality Condition

Definition (Pareto stationarity)

A design point is called pareto stationary, if the common descent direction $\omega = 0$.

- If Y is Pareto stationary, stop.

Note: Remember finite precision arithmetic! We have to loosen the stop condition $\omega < \text{tol}$. Common choice: $\text{tol} = 1E - 08$.

Application: Life-cycle Savings

- We consider a **social planner** aiming at maximizing the *agents' utility*, while maximizing the *goverment's revenue*
- Formalized, these objectives read

$$f_1(c, l, \boldsymbol{\tau}) = \sum_{t=1}^{T} (1 - r)^t u(c_t, l_t, \boldsymbol{\tau}) \tag{6}$$

$$f_2(a, c, l, \boldsymbol{\tau}) = \sum_{t=1}^{T} \beta^t \mathrm{R}(a_t, c_t, l_t, \boldsymbol{\tau}), \tag{7}$$

  with f_1 denoting the present value of utility u, and f_2 the present value of revenue R
- For simplicity, the literature often only considers one representative agent.

One Optimization Step

- The solcial planner chooses the agent's consumption c_t, and labour l_t such that her utility is maximized given a **fixed** tax schedule $\boldsymbol{\tau}$.

$$f_1(c, l, \boldsymbol{\tau}) = \sum_{t=1}^{T} (1 - r)^t u(c_t, l_t, \boldsymbol{\tau})$$

- The social planner chooses the tax schedule $\boldsymbol{\tau}$ given **fixed** assets a_t, consumption c_t, and labour l_t

$$f_2(a, c, l, \boldsymbol{\tau}) = \sum_{t=1}^{T} \beta^t \mathrm{R}(a_t, c_t, l_t, \boldsymbol{\tau}),$$

Social Planner Optimization Problem

$$
\begin{aligned}
&\max_{\boldsymbol{\tau}} \left\{ f_1(\hat{c}_t, \hat{l}_t, \boldsymbol{\tau}), f_2(\hat{a}_t, \hat{c}_t, \hat{l}_t, \boldsymbol{\tau}) \right\} \\
&\quad \text{s.t. } \hat{a}_t, \hat{c}_t, \hat{l}_t = \arg \max_{a_t, c_t, l_t} f_2(a_t, c_t, l_t, \boldsymbol{\tau}) \\
&\qquad (1+r)a_t + w_t l_t - c_t - \mathrm{T}(a_t, l_t, c_t, \boldsymbol{\tau}) + b_t - a_{t+1} = 0 \\
&\qquad (1+r_d)d_t + b_t - d_{t+1} = 0
\end{aligned}
\tag{8}
$$

Social Planner Optimization Problem

$$
\begin{aligned}
&\max_{\boldsymbol{\tau}} \left\{ f_1(\hat{c}_t, \hat{l}_t, \boldsymbol{\tau}), f_2(\hat{a}_t, \hat{c}_t, \hat{l}_t, \boldsymbol{\tau}) \right\} \\
&\quad \text{s.t. } \hat{a}_t, \hat{c}_t, \hat{l}_t = \arg\max_{a_t, c_t, l_t} f_2(a_t, c_t, l_t, \boldsymbol{\tau}) \\
&\qquad (1+r)a_t + w_t l_t - c_t - \mathrm{T}(a_t, l_t, c_t, \boldsymbol{\tau}) + b_t - a_{t+1} = 0 \\
&\qquad (1+r_d)d_t + b_t - d_{t+1} = 0
\end{aligned}
\tag{8}
$$

Social Planner Optimization Problem

$$
\begin{aligned}
&\max_{\boldsymbol{\tau}} \left\{ f_1(\hat{c}_t, \hat{l}_t, \boldsymbol{\tau}), f_2(\hat{a}_t, \hat{c}_t, \hat{l}_t, \boldsymbol{\tau}) \right\} \\
&\quad \text{s.t. } \hat{a}_t, \hat{c}_t, \hat{l}_t = \arg \max_{a_t, c_t, l_t} f_2(a_t, c_t, l_t, \boldsymbol{\tau}) \\
&\qquad (1 + r)a_t + w_t l_t - c_t - \mathrm{T}(a_t, l_t, c_t, \boldsymbol{\tau}) + b_t - a_{t+1} = 0 \\
&\qquad (1 + r_d)d_t + b_t - d_{t+1} = 0
\end{aligned}
\tag{8}
$$

The tax vector τ denotes our design vector Y.

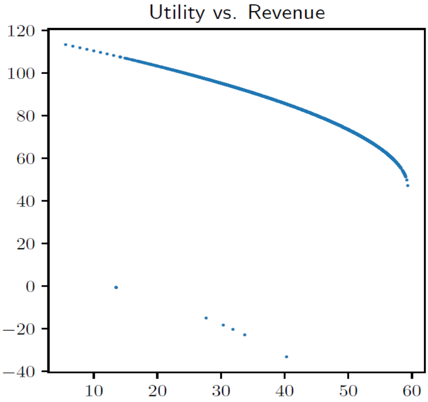
Sensitivity Analysis

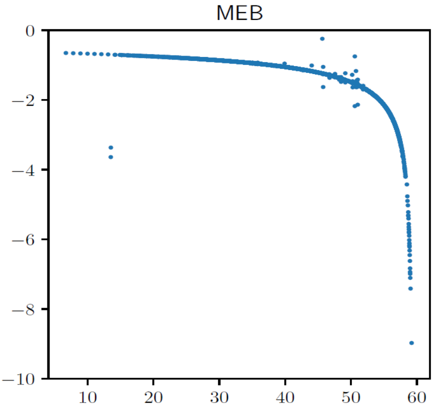
- The objectives do depend on the **solutions of an inner optimization problem**.
- Thus, the derivatives of the objective functions do depend on the derivatives of the optimal solution w.r.t. to $\boldsymbol{\tau}$ as

$$\begin{aligned}\frac{\partial f_1(\hat{c}_t(\boldsymbol{\tau}), \hat{l}_t(\boldsymbol{\tau}), \boldsymbol{\tau})}{\partial \boldsymbol{\tau}} =& \frac{\partial f_1(\hat{c}_t(\boldsymbol{\tau}), \hat{l}_t(\boldsymbol{\tau}), \boldsymbol{\tau})}{\partial c_t} \frac{\partial c_t}{\partial \boldsymbol{\tau}} + \\ & \frac{\partial f_1(\hat{c}_t(\boldsymbol{\tau}), \hat{l}_t(\boldsymbol{\tau}), \boldsymbol{\tau})}{\partial l_t} \frac{\partial l_t}{\partial \boldsymbol{\tau}} + \\ & \frac{\partial f_1(\hat{c}_t(\boldsymbol{\tau}), \hat{l}_t(\boldsymbol{\tau}), \boldsymbol{\tau})}{\partial \boldsymbol{\tau}}\end{aligned}$$

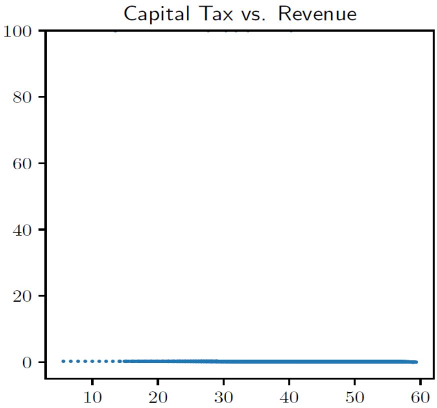
- CasADi provides this up to machine precision by applying either the forward or adjoint mode!

Results

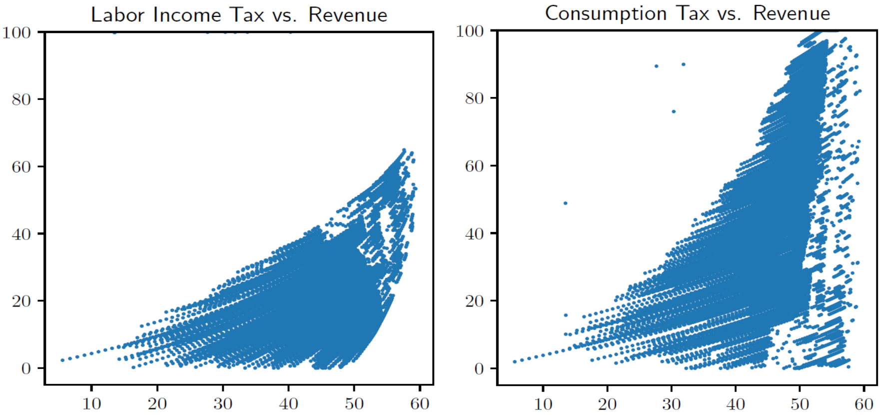

Utility vs. Revenue
120
100
80
60
40
20
0
−20
−40
10
20
30
40
50
60


MEB
0
−2
−4
−6
−8
−10
10
20
30
40
50
60

Results



Results



Labor Income Tax vs. Revenue
100
80
60
40
20
0
10
20
30
40
50
60
Consumption Tax vs. Revenue
100
80
60
40
20
0
10
20
30
40
50
60

Homotopy Methods

- The application of homotopy methods arises naturally as we already exploit the differentiability of our problem.
- Natural application of homotopy are to
  - follow the pareto frontier once starting at a collection of points we found by applying the multiple-gradient descent method, and
  - follow changes in the preference parameter (γ, η) starting from a collection of points we found by applying the multiple-gradient descent method.
- The Judd, Mueller Hompack90 to Python interface is still buggy, thus we cannot present the results yet. First results look promising.