Production-Function Approach to Portfolio Evaluation

Version 1.5 Draft

26 June 2020

Concept

We separate the financial and conversion-efficiency aspects of the production process, which are generic across all technologies, from the physical and technical aspects, which are necessarily specific to the particular process. The motivation for this is that the financial and waste computations can be done uniformly for any technology (even for disparate ones such as PV cells and biofuels) and that different experts may be required to assess the cost, waste, and techno-physical aspects of technological progress.

Formulation

Sets

Set	Description	Examples
$c \in \mathcal{C}$	capital	equipment
$f \in \mathcal{F}$	fixed cost	rent, insurance
$i \in \mathcal{I}$	input	feedstock, labor
$o \in \mathcal{O}$	output	product, co-product, waste
$m \in \mathcal{M}$	metric	cost, jobs, carbon footprint, efficiency, lifetime
$p \in \mathcal{P}$	technical parameter	temperature, pressure
$\nu \in N$	technology type	electrolysis, PV cell
$\theta \in \Theta$	scenario	the result of a particular investment
$\chi \in X$	investment category	investment alternatives
$\phi \in \Phi_\chi$	investment	a particular investment
$\omega \in \Omega$	portfolio	a basket of investments

Variables

Variable	Type	Description	Units
K	calculated	unit cost	USD/unit
C_c	function	capital cost	USD
τ_c	cost	lifetime of capital	year
S	cost	scale of operation	unit/year

F_f	function	fixed cost	USD/year
I_i	input	input quantity	input/unit
I_i^*	calculated	ideal input quantity	input/unit
η_i	waste	input efficiency	input/input
p_i	cost	input price	USD/input
O_o	calculated	ideal output quantity	output/unit
O_o^*	calculated	output quantity	output/unit
η'_o	waste	output efficiency	output/output
p'_o	cost	output price (+/-)	USD/output
μ_m	calculated	metric	metric/unit
P_o	function	production function	output/unit
M_m	function	metric function	metric/unit
α_p	parameter	technical parameter	(mixed)
ξ_θ	variable	scenario inputs	(mixed)
ζ_θ	variable	scenario outputs	(mixed)
ψ	function	scenario evaluation	(mixed)
σ_ϕ	function	scenario probability	1
q_ϕ	variable	investment cost	USD
$\boldsymbol{\zeta}_\phi$	random variable	investment outcome	(mixed)
$\mathbf{Z}(\omega)$	random variable	portfolio outcome	(mixed)
$Q(\omega)$	calculated	portfolio cost	USD
$Q^{\min}$	parameter	minimum portfolio cost	USD
$Q^{\max}$	parameter	maximum portfolio cost	USD

Cost

The cost characterizations (capital and fixed costs) are represented as functions of the scale of operations and of the technical parameters in the design:

- Capital cost: $C_c(S, \alpha_p)$.
- Fixed cost: $F_f(S, \alpha_p)$.

The per-unit cost is computed using a simple levelization formula:

$$K = \left(\sum_c C_c / \tau_c + \sum_f F_f \right) / S + \sum_i p_i \cdot I_i - \sum_o p'_o \cdot O_o$$

Waste

The waste relative to the idealized production process is captured by the η parameters. Expert elicitation might estimate how the ηs would change in response to R&D investment.

- Waste of input: $I_i^* = \eta_i I_i$.
- Waste of output: $O_o = \eta'_o O_o^*$.

Production

The production function idealizes production by ignoring waste, but accounting for physical and technical processes (e.g., stoichiometry). This requires a technical model or a tabulation/fit of the results of technical modeling.

$$O_o^* = P_o(S, C_c, \tau_c, F_f, I_i^*, \alpha_p)$$

Metrics

Metrics such as efficiency, lifetime, or carbon footprint are also compute based on the physical and technical characteristics of the process. This requires a technical model or a tabulation/fit of the results of technical modeling. We use the convention that higher values are worse and lower values are better.

$$\mu_m = M_m(S, C_c, \tau_c, F_f, I_i, I_i^*, O_o^*, O_o, K, \alpha_p)$$

Scenarios

A *scenario* represents a state of affairs for a technology ν. If we denote the scenario as θ, we have the input variables

$$\xi_\theta = (C_c, F_f, I_i, \alpha_p)\ |_\theta$$

and the output variables

$$\zeta_\theta = (K, \mu_m)\ |_\theta$$

and their relationship

$$\zeta_\theta = \psi_\nu(\xi_\theta)\ |_{\nu=\nu(\theta)}$$

where

$$\psi_\nu = (P_o, M_m)\ |_\nu$$

for the technology of the scenario.

Investments

An *investment* ϕ assigns a probability distribution to scenarios:

$\sigma_\phi(\theta) = P(\theta \mid \phi)$.

such that

$\int d\theta \sigma_\phi(\theta) = 1$ or $\sum_\theta \sigma_\phi(\theta) = 1$,

depending upon whether one is performing the computations discretely or continuously. Expectations and other measures on probability distributions can be computed from the $\sigma_\phi(\theta)$. We treat the outcome $\boldsymbol{\zeta}_\phi$ as a random variable for the outcomes ζ_θ according to the distribution $\sigma_\phi(\theta)$.

Because investment options may be mutually exclusive, as is the case for investing in the same R&D at different funding levels, we say Φ_χ is the set of mutually exclusive investments (i.e., only one can ocurr) in investment category χ: investments in different categories χ can be combined arbitrarily, but just one investment from each Φ_χ may be chosen.

Thus the universe of all portfolios is $\Omega = \prod_\chi \Phi_\chi$, so a particular portfolio $\omega \in \Omega$ has components $\phi = \omega_\chi \in \Phi_\chi$. The overall outcome of a portfolio is a random variable:

$$\mathbf{Z}(\omega) = \sum_\chi \boldsymbol{\zeta}_\phi \mid_{\phi=\omega_\chi}$$

The cost of an investment q_ϕ, so the cost of a porfolio is:

$$Q(\omega) = \sum_\chi q_\phi \mid_{\phi=\omega_\chi}$$

Decision problem

The multi-objective decision problem is

$\min_{\omega\in\Omega} \mathbb{F}\, \mathbf{Z}(\omega)$

such that

$Q^{\min} \leq Q(\omega) \leq Q^{\max}$,

where $\mathbb{F}$ is the expectation operator $\mathbb{E}$, value-at-risk, or another operator on probability spaces. Recall that $\mathbf{Z}$ is a vector with components for cost K and each metric μ_m, so this is a multi-objective problem.

The two-stage decision problem is a special case of the general problem outlined here: Each scenario θ can be considers as a composite of one or more stages.

Experts

Each expert elicitation takes the form of an assessment of the probability and range (e.g., 10th to 90th percentile) of change in the cost or waste parameters or the production or metric functions. In essence, the expert elicitation defines $\sigma_\phi(\theta)$ for each potential scenario θ of each investment ϕ.

Examples

Idealized electrolysis of water

Here is a very simple model for electrolysis of water. We just have water, electricity, a catalyst, and some lab space. We choose the fundamental unit of operation to be moles of H_2:

$H_2O \rightarrow H_2 + ½\ O_2$

Experts could assess how much R&D to increase the various efficiencies η would cost. They could also suggest different catalysts, adding alkali, or replacing the process with PEM.

Tracked quantities.

$\mathcal{C} = \{\text{catalyst}\}$

$\mathcal{F} = \{\text{rent}\}$

$\mathcal{I} = \{\text{water}, \text{electricity}\}$

$\mathcal{O} = \{\text{oxygen}, \text{hydrogen}\}$

$\mathcal{M} = \{\text{jobs}\}$

Current design.

$I_{\text{water}} = 19.04\ \text{g/mole}$

$\eta_{\text{water}} = 0.95$ (due to mass transport loss on input)

$I_{\text{electricity}} = 279\ \text{kJ/mole}$

$\eta_{\text{electricity}} = 0.85$ (due to ohmic losses on input)

$\eta_{\text{oxygen}} = 0.90$ (due to mass transport loss on output)

$\eta_{\text{hydrogen}} = 0.90$ (due to mass transport loss on output)

Current costs.

$C_{\text{catalyst}} = (0.63\ \text{USD}) \cdot \frac{s}{6650\ \text{mole/yr}}$ (cost of Al-Ni catalyst)

$\tau_{\text{catalyst}} = 3\ \text{yr}$ (effective lifetime of Al-Ni catalyst)

$$F_{\mathrm{rent}} = (1000\ \mathrm{USD/yr}) \cdot \frac{S}{6650\ \mathrm{mole/yr}}$$

$S = 6650$ mole/yr (rough estimate for a 50W setup)

Current prices.

$$p_{\mathrm{water}} = 4.8 \cdot 10^{-3}\ \mathrm{USD/mole}$$

$$p_{\mathrm{electricity}} = 3.33 \cdot 10^{-5}\ \mathrm{USD/kJ}$$

$$p_{\mathrm{oxygen}} = 3.0 \cdot 10^{-3}\ \mathrm{USD/g}$$

$$p_{\mathrm{hydrogen}} = 1.0 \cdot 10^{-2}\ \mathrm{USD/g}$$

Production function (à la Leontief)

$$P_{\mathrm{oxygen}} = (16.00\ \mathrm{g}) \cdot \min\left\{\frac{I^*_{\mathrm{water}}}{18.08\ \mathrm{g}}, \frac{I^*_{\mathrm{electricity}}}{237\ \mathrm{kJ}}\right\}$$

$$P_{\mathrm{hydrogen}} = (2.00\ \mathrm{g}) \cdot \min\left\{\frac{I^*_{\mathrm{water}}}{18.08\ \mathrm{g}}, \frac{I^*_{\mathrm{electricity}}}{237\ \mathrm{kJ}}\right\}$$

Metric function.

$$M_{\mathrm{cost}} = K / O_{\mathrm{hydrogen}}$$

$$M_{\mathrm{GHG}} = \Big((0.00108\ \mathrm{gCO2e/gH20}) I_{\mathrm{water}} + (0.138\ \mathrm{gCO2e/kJ}) I_{\mathrm{electricity}}\Big) / O_{\mathrm{hydrogen}}$$

$$M_{\mathrm{jobs}} = (0.00015\ \mathrm{job/mole}) / O_{\mathrm{hydrogen}}$$

Performance of current design.

$K = 0.18$ USD/mole (i.e., not profitable since it is positive)

$$O_{\mathrm{oxygen}} = 14\ \mathrm{g/mole}$$

$$O_{\mathrm{hydrogen}} = 1.8\ \mathrm{g/mole}$$

$$\mu_{\mathrm{cost}} = 0.102\ \mathrm{USD/gH2}$$

$$\mu_{\mathrm{GHG}} = 21.4\ \mathrm{gCO2e/gH2}$$

$$\mu_{\mathrm{jobs}} = 0.000083\ \mathrm{job/gH2}$$

Implementation

Database tables (one per set) hold all of the variables and the expert assessments. These tables are augmented by concise code with mathematical representations of the production and metric functions.

The Monte-Carlo computations are amenable to fast tensor-based implementation in Python.

See <https://github.com/NREL/portfolio/tree/master/production-function/framework/code/tyche/> for the `tyche` package that computes cost, production, and metrics from a technology design.

Database tables

Each analysis case is represented by a `Technology` and a `Scenario` within that technology.

Metadata about indices

The `indices` table simply describes the various indices available for the variables. The `Offset` column specifies the memory location in the argument for the production and metric functions.

Technology	Type	Index	Offset	Description	Notes
Simple electrolysis	Capital	Catalyst	0	Catalyst	
Simple electrolysis	Fixed	Rent	0	Rent	
Simple electrolysis	Input	Water	0	Water	
Simple electrolysis	Input	Electricity	1	Electricity	
Simple electrolysis	Output	Oxygen	0	Oxygen	
Simple electrolysis	Output	Hydrogen	1	Hydrogen	
Simple electrolysis	Metric	Cost	0	Cost	
Simple electrolysis	Metric	Jobs	1	Jobs	
Simple electrolysis	Metric	GHG	2	GHGs	

Design variables

The `design` table specifies the values of all of the variables in the mathematical formulation of the design.

Technology	Scenario	Variable	Index	Value	Units	Notes
Simple	Base	Input	Water	19.04	g/mole	I_{water}

electrolysis						
Simple electrolysis	Base	Input Efficiency	Water	0.95	1	η_{water}
Simple electrolysis	Base	Input	Electricity	279	kJ/mole	$I_{\text{electricity}}$
Simple electrolysis	Base	Input Efficiency	Electricity	0.85	1	$\eta_{\text{electricity}}$
Simple electrolysis	Base	Output Efficiency	Oxygen	0.90	1	η_{oxygen}
Simple electrolysis	Base	Output Efficiency	Hydrogen	0.90	1	η_{hydrogen}
Simple electrolysis	Base	Lifetime	Catalyst	3	yr	τ_{catalyst}
Simple electrolysis	Base	Scale		6650	mole/yr	S
Simple electrolysis	Base	Input price	Water	4.8e-3	USD/mole	p_{water}
Simple electrolysis	Base	Input price	Electricity	3.33e-5	USD/kJ	$p_{\text{electricity}}$
Simple electrolysis	Base	Output price	Oxygen	3.0e-3	USD/g	p_{oxygen}
Simple electrolysis	Base	Output price	Hydrogen	1.0e-2	USD/g	p_{hydrogen}

Note that the `Value` column can either contain numeric literals or Python expressions specifying probability distribution functions. For example, a normal distribution with mean of five and standard deviation of two would be written `st.norm(5, 2)`. All of the Scipy probability distribution functions are available for use, as are two special functions, `constant` and `mixture`. The `constant` distribution is just a single constant value; the `mixture` distribution is the mixture of a list of distributions, with specified relative weights. The `mixture` function is particularly important because it allows one to specify a first distribution in the case of an R&D breakthrough, but a second distribution if no breakthrough occurs.

Metadata for functions

The `functions` table simply documents which Python module and functions to use for the technology and scenario.

Technology	Style	Module	Capital	Fixed	Production	Metrics	Notes
Simple electrolysi	numpy	simple_electrolysis	capital_cost	fixed_cost	production	metrics	

s

Currently only the `numpy` style of function is supported, but later `plain` Python functions and `tensorflow` functions will be allowed.

Parameters for functions

The `parameters` table contains ad-hoc parameters specific to the particular production and metrics functions. The `Offset` column specifies the memory location in the argument for the production and metric functions.

Technology	Scenario	Parameter	Offset	Value	Units	Notes
Simple electrolysis	Base	Oxygen production	0	16.00	g	
Simple electrolysis	Base	Hydrogen production	1	2.00	g	
Simple electrolysis	Base	Water consumption	2	18.08	g	
Simple electrolysis	Base	Electricity consumption	3	237	kJ	
Simple electrolysis	Base	Jobs	4	1.5e-4	job/mole	
Simple electrolysis	Base	Reference scale	5	6650	mole/yr	
Simple electrolysis	Base	Reference capital cost for catalyst	6	0.63	USD	
Simple electrolysis	Base	Reference fixed cost for rent	7	1000	USD/yr	
Simple electrolysis	Base	GHG factor for water	8	0.00108	gCO2e/g	based on 244,956 gallons = 1 Mg CO2e
Simple electrolysis	Base	GHG factor for electricity	9	0.138	gCO2e/kJ	based on 1 kWh = 0.5 kg CO2e

Units for results

The `results` table simply specifies the units for the results.

Technology	Variable	Index	Units	Notes
Simple electrolysis	Cost	Cost	USD/mole	
Simple electrolysis	Output	Oxygen	g/mole	

Simple electrolysis	Output	Hydrogen	g/mole
Simple electrolysis	Metric	Cost	job/gH2
Simple electrolysis	Metric	Jobs	job/gH2
Simple electrolysis	Metric	GHG	gCO2e/gH2

Tranches of investments.

In the `tranches` table, each *category* of investment contains a set of mutually exclusive *tranches* that may be associated with one or more *scenarios* defined in the `designs` table. Typically, a category is associated with a technology area and each tranche corresponds to an investment strategy within that category.

Category	Tranche	Scenario	Amount	Notes
Electrolysis R&D	No Electrolysis R&D	Base Electrolysis	0	
Electrolysis R&D	Low Electrolysis R&D	Slow Progress on Electrolysis	1000000	
Electrolysis R&D	Medium Electrolysis R&D	Moderate Progress on Electrolysis	2500000	
Electrolysis R&D	High Electrolysis R&D	Fast Progress on Electrolysis	5000000	

Investments

In the `investments` table, each *investment* is associated with a single *tranche* in one or more *categories*. An investment typically combines tranches from several different investment categories.

Investment	Category	Tranche	Notes
No R&D Spending	Electrolysis R&D	No Electrolysis R&D	
Low R&D Spending	Electrolysis R&D	Low Electrolysis R&D	
Medium R&D Spending	Electrolysis R&D	Medium Electrolysis R&D	
High R&D Spending	Electrolysis R&D	High Electrolysis R&D	

Python module and functions

Each technology design requires a Python module with a production and metrics function.

```
# Simple electrolysis.
```

```python
# All of the computations must be vectorized, so use `numpy`.
import numpy as np


# Capital-cost function.
def capital_cost(scale, parameter):

  # Scale the reference values.
  return np.stack([np.multiply(parameter[6], np.divide(scale,
parameter[5]))])


# Fixed-cost function.
def fixed_cost(scale, parameter):

  # Scale the reference values.
  return np.stack([np.multiply(parameter[7], np.divide(scale,
parameter[5]))])


# Production function.
def production(capital, fixed, input, parameter):

  # Moles of input.
  water       = np.divide(input[0], parameter[2])
  electricity = np.divide(input[1], parameter[3])

  # Moles of output.
  output = np.minimum(water, electricity)

  # Grams of output.
  oxygen   = np.multiply(output, parameter[0])
  hydrogen = np.multiply(output, parameter[1])

  # Package results.
  return np.stack([oxygen, hydrogen])


# Metrics function.
def metrics(capital, fixed, input_raw, input, output_raw, output, cost,
parameter):

  # Hydrogen output.
  hydrogen = output[1]

  # Cost of hydrogen.
  cost1 = np.divide(cost, hydrogen)
```

```python
# Jobs normalized to hydrogen.
jobs = np.divide(parameter[4], hydrogen)

# GHGs associated with water and electricity.
water       = np.multiply(input_raw[0], parameter[8])
electricity = np.multiply(input_raw[1], parameter[9])
co2e = np.divide(np.add(water, electricity), hydrogen)

# Package results.
return np.stack([cost1, jobs, co2e])
```