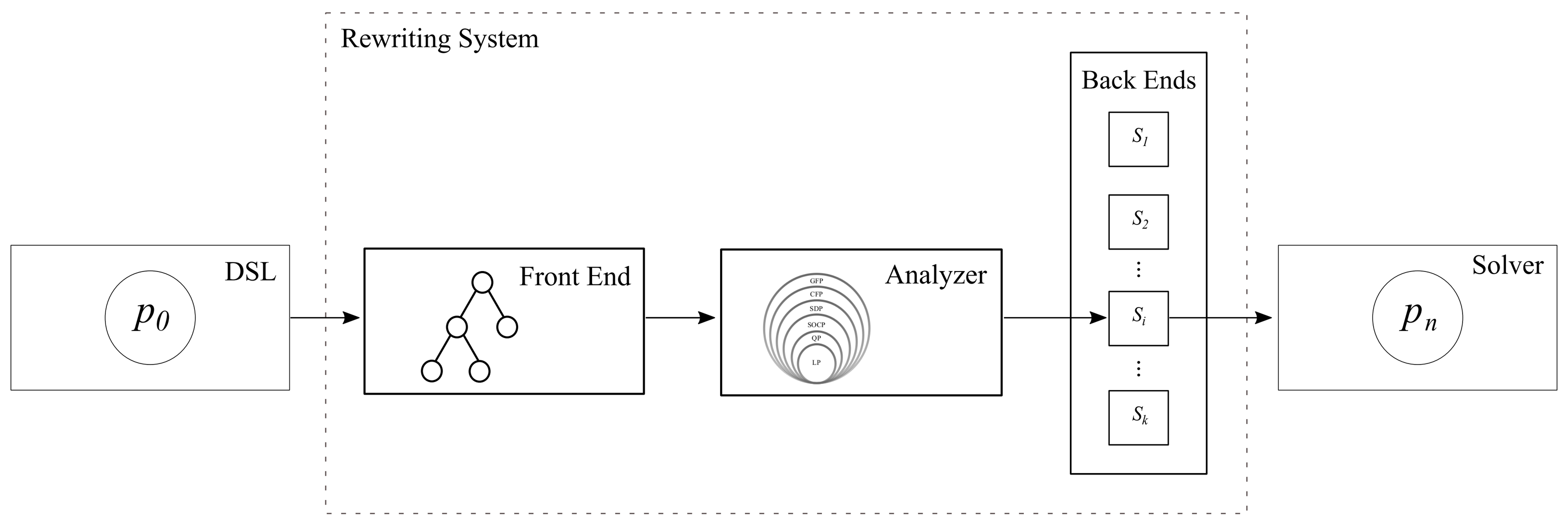
Disciplined Geometric Programming

Akshay Agrawal [1] Steven Diamond [2] Stephen Boyd [1]

[1]Stanford University, Electrical Engineering [2]Stanford University, Computer Science Department

Domain-specific languages (DSLs) for convex optimization

- DSLs for convex optimization make it easy to specify and solve convex problems
- Modern DSLs (CVXPY, CVXR, Convex.jl, CVX) based on disciplined convex programming (DCP) [6]
- DCP is a library of functions (atoms) with known curvature and monotonicity, and a composition rule for combining them.



Geometric programming

A geometric program (GP) [5] is an optimization problem

$$\begin{array}{ll} \text{minimize} & f_0(x) \\ \text{subject to} & f_i(x) \leq 1, \quad i = 1, \ldots, m \\ & g_i(x) = 1, \quad i = 1, \ldots, p, \end{array}$$

- $g_i : \mathbf{R}^n_{++} \to \mathbf{R}$ are monomials: $(x_1, \ldots, x_n) \mapsto c x_1^{a_1} \cdots x_n^{a_n}$, $c > 0$.
- $f_i : \mathbf{R}^n_{++} \to \mathbf{R}$ are posynomials: sums of monomials

Solving a GP reduces to solving a convex optimization problem, so GPs can be solved reliably and efficiently. Applications include [3]:

- chemical engineering
- circuit design
- transformer design
- aircraft design
- mechanical engineering
- communications

Log-log convex programs

We introduce log-log convex programs (LLCPs), which generalize GPs

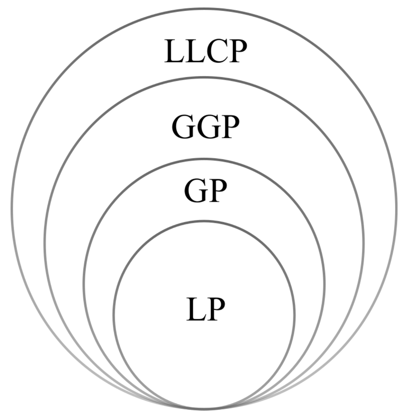
- For $f : D \to \mathbf{R}_{++}$, $D \subseteq \mathbf{R}^n_{++}$, $F(u) = \log f(e^u)$ is its log-log transformation
- f is log-log convex, log-log concave, or log-log affine if F is convex, concave, or affine (resp.)
- A log-log convex program is like a GP but with f_i log-log convex, g_i log-log affine

Direct analogue of the composition rule for convex functions holds for log-log convex functions

- For example, if $f : \mathbf{R}^k_{++} \to \mathbf{R}_{++}$ is log-log convex and increasing in each argument, and if $f_1, \ldots, f_n$ are log-log convex, then $f(f_1(x), \ldots, f_k(x))$ is log-log convex

Hierarchy of optimization problems.

LLCPs generalize GPs and so-called generalized geometric programs (GGPs)



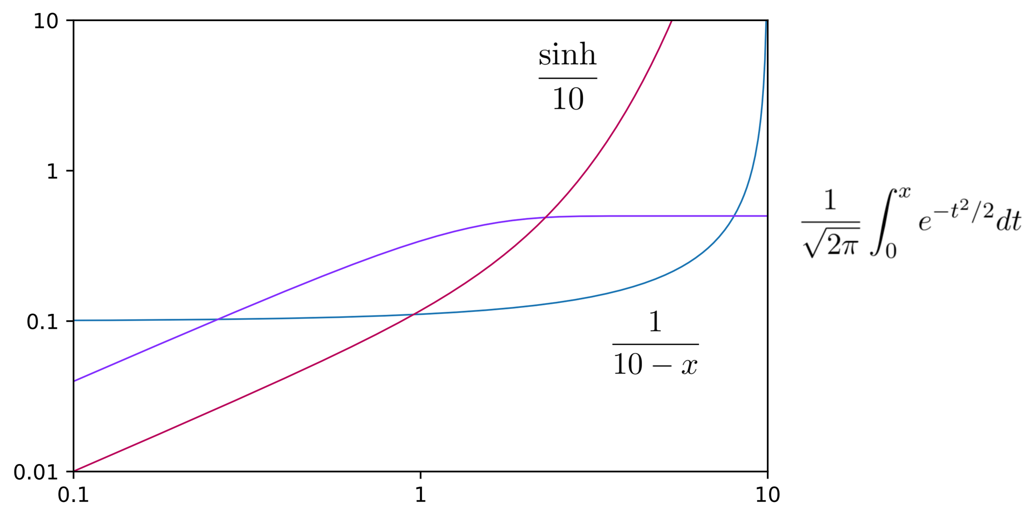
Properties

Convexity with respect to the geometric mean. f is log-log convex if (and only if) it is convex with respect to the geometric mean, *i.e.*

$$f(x^\theta \circ y^{1-\theta}) \leq f(x)^\theta f(y)^{1-\theta},$$

for $\theta \in [0, 1]$ and $x, y \in \mathbf{dom}\, f$ ($\circ$ is the elementwise product, and the powers are meant elementwise)

Scalar functions. Scalar log-log convex functions are convex on a log-log plot



Examples

Log-log affine functions

- products
- ratios
- powers

Log-log convex functions

- $x_1 + x_2$, $\max(x_1, x_2)$
- posynomials
- ℓ_p norms
- $(I - X)^{-1}$, (spectral radius of X less than 1)

Log-log concave functions

- $x_1 - x_2$, with $x_1 > x_2 > 0$
- $-x \log x$, $x \in (0, 1)$
- complementary CDF of a log-concave density, *e.g.* $\frac{1}{\sqrt{2\pi}} \int_x^\infty e^{-t^2/2} dt$

Disciplined geometric programming

Analogue of DCP, but for LLCPs

- Library of atoms with known log-log curvature (sum, product, ratio, exp, ...)
- Atoms may be combined using the composition rule
- Can express LLCPs of the form

$$\begin{array}{ll} \text{minimize} & f_0(x) \\ \text{subject to} & f_i(x) \leq \tilde{f}_i(x), \quad i = 1, \ldots, m \\ & g_i(x) = \tilde{g}_i(x), \quad i = 1, \ldots, p, \end{array} \qquad (1)$$

with f_i log-log convex, $\tilde{f}_i$ log-log concave, g_i and $\tilde{g}_i$ log-log affine (must be verifiable by the composition rule)

Implementation

DGP implemented as a reduction in CVXPY 1.0:

`https://www.cvxpy.org/tutorial/dgp/index.html`

- User types in a DGP-compliant LLCP and calls a single method to solve it
- CVXPY reduces the LLCP to a (disciplined) convex program, solves it, and returns a solution to the original problem

```
import cvxpy as cp
x = cp.Variable(pos=True)
y = cp.Variable(pos=True)
z = cp.Variable(pos=True)
objective_fn = x*y*z
constraints = [4*x*y*z + 2*x*z <= 10, x <= 2*y, y <= 2*x, z >= 1]
problem = cp.Problem(cp.Maximize(objective_fn), constraints)
problem.solve(gp=True)
print(problem.value)
print(x.value, y.value, z.value)
```

References

[1] Akshay Agrawal, Steven Diamond, and Stephen Boyd. Disciplined geometric programming. *Optimization Letters*, 2019.

[2] Akshay Agrawal, Robin Verschueren, Steven Diamond, and Stephen Boyd. A rewriting system for convex optimization problems. *Journal of Control and Decision*, 5(1):42–60, 2018.

[3] Stephen Boyd, Seung-Jean Kim, Lieven Vandenberghe, and Arash Hassibi. A tutorial on geometric programming. *Optimization and engineering*, 8(1):67, 2007.

[4] Edward Burnell and Warren Hoburg. GPkit software for geometric programming. `https://github.com/convexengineering/gpkit`, 2018. Version 0.7.0.

[5] Richard Duffin, Elmor Peterson, and Clarence Zener. Geometric programming — theory and application, 1967.

[6] Michael Grant, Stephen Boyd, and Yinyu Ye. Disciplined convex programming. In *Global optimization*, pages 155–210. Springer, 2006.

[7] Constantin Niculescu. Convexity according to the geometric mean. *Mathematical Inequalities and Applications*, 3(2):155–167, 04 2000.