Geometric Programming with Applications in Power Control

Daniel P. Palomar
Hong Kong University of Science and Technolgy (HKUST)

ELEC5470 - Convex Optimization

Fall 2018-19, HKUST, Hong Kong

Outline of Lecture

- GP
- Generalized GP
- Examples
- More transformations for GP
- Power control via GP
- Complementary GP and Signomial Programming
- Summary

GP

- A *geometric program* (GP) is a type of mathematical optimization problem with objective and constraint functions with a special form.

- Importance of GP:
  - new numerical methods can solve large-scale GPs extremely efficiently and reliably
  - a number of practical problems have recently been found to be equivalent to (or well approximated) by GPs (e.g., circuit design, geometric placement problems, power control in wireless communications, duality in information theory).

- GP modeling is not a brainless employment of a software package; it involves some knowledge, as well as creativity, to be done effectively.

Monomials

- A *monomial* is a real-valued function f of x of the form[1]

$$f(x) = cx_1^{a_1} x_2^{a_2} \cdots x_n^{a_n}$$

  where $c > 0$ and $a_i \in \mathrm{R}$.

- Calculus for monomials:
  - if f and g are both monomials then so are fg and f/g.
  - a monomial raised to any power is also a monomial:

$$f^{\gamma}(x) = \left(cx_1^{a_1} x_2^{a_2} \cdots x_n^{a_n}\right)^{\gamma} = c^{\gamma} x_1^{\gamma a_1} x_2^{\gamma a_2} \cdots x_n^{\gamma a_n}.$$

[1] The term monomial used in the context of GP differs from the standard definition in algebra, where a monomial is similarly defined but with nonnegative integer exponents a_i and $c = 1$.

Posynomials

- A *posynomial* is a sum of monomials:

$$f(x) = \sum_{k=1}^{K} c_k x_1^{a_{1k}} x_2^{a_{2k}} \cdots x_n^{a_{nk}}.$$

- Calculus for posynomials:
  - posynomials are closed under addition, multiplication, and positive scaling
  - if f is a posynomial and g is a monomial, then f/g is a posynomial
  - if γ is a nonnegative integer and f is a posynomial, then f^γ always makes sense and is a posynomial.

Examples

- Monomials (assume x, y, and z positive variables):

$$2x, \quad 0.23, \quad 2z\sqrt{x/y}, \quad 3x^2y^{-0.12}z.$$

- Posynomials:

$$0.23 + x/y, \quad 2\left(1 + xy\right)^3, \quad 2x + 3y + 2z.$$

- What about these?:

$$-1.1, \quad 2\left(1 + xy\right)^{3.1}, \quad 2x + 3y - 2z, \quad x^2 + \tan x.$$

GP in Standard Form

- A *GP in standard form* is a problem of the form

$$\begin{array}{lll} \underset{x}{\text{minimize}} & f_0(x) & \\ \text{subject to} & f_i(x) \leq 1 & i = 1, \cdots, m \\ & g_i(x) = 1 & i = 1, \cdots, p. \end{array}$$

  where f_i are posynomials, g_i are monomials, and x_i are the optimization variables (with the implicit assumption that the variables are positive, i.e., $x_i > 0$).

- Observe that posynomials and monomials in the definition of a GP seem to play the role of convex functions and affine functions, respectively, in the definition of a convex problem.

- Note that a GP in standard form is nonconvex!
  - Ex.: $\sqrt{x}$ or x^3.
- A simple example of a GP in standard form is

$$\begin{array}{ll} \text{minimize} & x^{-1}y^{-1/2}z^{-1} + 2.3xz + 4xyz \\ \text{subject to} & (1/3)\, x^{-2}y^{-2} + (4/3)\, y^{1/2}z^{-1} \leq 1 \\ & x + 2y + 3z \leq 1 \\ & (1/2)\, xy = 1 \end{array}$$

A Simple Extension of GP

- If f is a posynomial and g is a monomial, then the constraint $f(x) \leq g(x)$ can be handled by expressing it as $f(x)/g(x) \leq 1$ which is a valid posynomial inequality constraint.

- Example:

$$\begin{array}{ll} \text{maximize} & x/y \\ \text{subject to} & 2 \leq x \leq 3 \\ & x^2 + 3y/z \leq \sqrt{y} \\ & x/y = z^2 \end{array}$$

can be rewritten as the GP

$$\begin{array}{ll} \text{minimize} & x^{-1}y \\ \text{subject to} & 2x^{-1} \leq 1, \qquad (1/3)\,x \leq 1 \\ & x^2y^{-1/2} + 3y^{1/2}z^{-1} \leq 1 \\ & xy^{-1}z^{-2} = 1. \end{array}$$

Simple Application

- Optimize the shape of a box with height h, width w, and depth d:
  - objective: maximize the volume
  - limit on the total wall area of A_{wall}
  - limit on the floor area of A_{floor}
  - lower and upper bounds on the aspect ratios h/w and w/d.

- Problem formulation:

$$\begin{array}{ll} \text{maximize} & hwd \\ \text{subject to} & 2\left(hw+hd\right) \leq A_{\text{wall}}, \qquad wd \leq A_{\text{floor}} \\ & \alpha \leq h/w \leq \beta, \qquad \gamma \leq w/d \leq \delta \end{array}$$

which can be rewritten as a GP in standard form.

How GPs Are Solved

- Standard interior-point algorithms can solve a GP with 1,000 variables and 10,000 constraints in under a minute, on a small desktop computer.

- For sparse problems, far larger problems can be readily solved, e.g., a problem with 10,000 variables and 1,000,000 constraints can be solved in minutes.

- It is also possible to optimize a GP solver for a particular application, exploiting special structure to gain in efficiency or in size of manageable problems.

- Interior-point methods for GPs are also very robust; they require no algorithm parameter tuning and no starting point or initial guess of the optimal solution.

- They always find the (global) optimal solution and, when the problem is infeasible, they provide a certificate of infeasibility.

- The main trick to solving a GP efficiently is to convert it to a *convex optimization problem*.

GP in Convex Form

- Start with the GP in standard form

$$\begin{array}{lll} \underset{x}{\text{minimize}} & f_0(x) & \\ \text{subject to} & f_i(x) \leq 1 & i = 1, \cdots, m \\ & g_i(x) = 1 & i = 1, \cdots, p. \end{array}$$

- Then, use a logarithmic change of variables and a logarithmic transformation of the objective and constraint functions:
  - in place of the original variables x_i we use their logarithms $\tilde{x}_i = \log x_i$ (so $x_i = e^{\tilde{x}_i}$)
  - instead of minimizing the objective f_0, we minimize its logarithm $\log f_0$

– we replace the inequality constraints $f_i \leq 1$ with $\log f_i \leq 0$ and the equality constraints $g_i = 1$ with $\log g_i = 0$

to obtain the convex problem

$$\begin{array}{lll} \underset{\tilde{x}}{\text{minimize}} & \log f_0\left(e^{\tilde{x}}\right) & \\ \text{subject to} & \log f_i\left(e^{\tilde{x}}\right) \leq 0 & i = 1, \cdots, m \\ & \log g_i\left(e^{\tilde{x}}\right) = 0 & i = 1, \cdots, p. \end{array}$$

Proof. (of convexity)

The monomial $g(x) = c x_1^{a_1} x_2^{a_2} \cdots x_n^{a_n}$ under the transformation becomes

$$\begin{aligned} \log g\left(e^{\tilde{x}}\right) &= \log c + a_1 \log x_1 + a_2 \log x_2 + \cdots + a_n \log x_n \\ &= \tilde{c} + a_1 \tilde{x}_1 + a_2 \tilde{x}_2 + \cdots + a_n \tilde{x}_n \end{aligned}$$

which is an *affine* function of the variables $\tilde{x}_i$.

The posynomial $f(x) = \sum_k c_k x_1^{a_{1k}} x_2^{a_{2k}} \cdots x_n^{a_{nk}}$ under the transformation becomes

$$
\begin{aligned}
\log f\left(e^{\widetilde{x}}\right) &= \log \sum_k c_k e^{a_{1k}\tilde{x}_1} e^{a_{2k}\tilde{x}_2} \cdots e^{a_{nk}\tilde{x}_n} \\
&= \log \sum_k e^{\tilde{c}_k + a_{1k}\tilde{x}_1 + a_{2k}\tilde{x}_2 + \cdots + a_{nk}\tilde{x}_n}
\end{aligned}
$$

which is the convex log-sum-exp function.

Therefore, the monomial equality constraints become affine constrains and the posynomial inequality constraints become convex constraints. $\square$

Generalized GP: Fractional Powers of Posynomials

- We know that posynomials are preserved under positive integer powers, e.g., the constraint

$$f_1(x)^2 + f_2(x)^3 \leq 1$$

is a standard posynomial inequality.

- However,

$$f_1(x)^{2.2} + f_2(x)^{3.1} \leq 1$$

is not a valid posynomial constraint.

- Nevertheless, we can handle it using a trick:

- introduce new variables t_1 and t_2, along with the inequality constraints

$$f_1(x) \le t_1, \qquad f_2(x) \le t_2$$

  which are both valid posynomial inequality constraints, and
- replace the original nonposynomial inequality with

$$t_1^{2.2} + t_2^{3.1} \le 1$$

  which is a valid posynomial inequality.

- This relies on the fact that the posynomial $t_1^{2.2} + t_2^{3.1}$ is an *increasing* function of t_1 and t_2.

- This method can be used to handle any number of *positive fractional powers*.

- Example: the problem

$$
\begin{array}{ll}
\text{minimize} & \sqrt{1+x^2} + (1+y/z)^{3.1} \\
\text{subject to} & 1/x + z/y \leq 1 \\
& (x/y + y/z)^{2.2} + x + y \leq 1
\end{array}
$$

is not a GP because neither the objective nor the second constraint function are posynomials, but can be rewritten as the GP

$$
\begin{array}{ll}
\text{minimize} & t_1^{0.5} + t_2^{3.1} \\
\text{subject to} & 1 + x^2 \leq t_1 \\
& 1 + y/z \leq t_2 \\
& 1/x + z/y \leq 1 \\
& t_3^{2.2} + x + y \leq 1 \\
& x/y + y/z \leq t_3.
\end{array}
$$

Generalized GP: Composite Functions of Posynomials

- Same idea also applies to other composite functions of posynomials.

- If f_0 is a posynomial of k variables, with all its exponents positive (or zero), and $f_1, \cdots, f_k$ are posynomials, then the inequality

$$f_0\left(f_1\left(x\right), \cdots, f_k\left(x\right)\right) \leq 1$$

can be handled by replacing it with

$$f_0\left(t_1, \cdots, t_k\right) \leq 1$$

and

$$f_1\left(x\right) \leq t_1, \quad \cdots, \quad f_k\left(x\right) \leq t_k.$$

- This shows that products, as well as sums, of fractional positive powers of posynomials can be handled.

Generalized GP: Maximum of Posynomials

- Suppose f_1, f_2, and f_3 are posynomials. The inequality constraint

$$\max\{f_1(x), f_2(x)\} + f_3(x) \leq 1$$

is certainly not a posynomial inequality.

- We can introduce a new variable t and two new inequalities to obtain:

$$t + f_3(x) \leq 1$$

and

$$f_1(x) \leq t, \qquad f_2(x) \leq t$$

which are valid posynomial inequality constraints.

Generalized Posynomials

- A function f of positive variables $x_1, \cdots, x_n$ is a *generalized posynomial* if it can be formed from posynomials using the operations of addition, multiplication, positive (fractional) powers, and maximum.

- Examples of generalized posynomials:

$$\max\left\{1 + x_1, 2x_1 + x_2^{0.2} x_3^{-3.9}\right\}$$

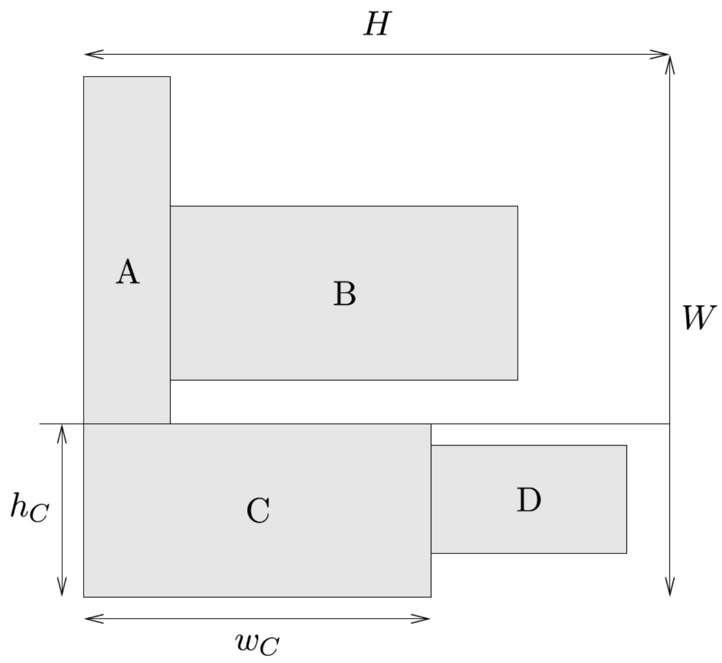
$$(1 + \max\{x_1, x_2\}) \left(\begin{array}{c} \max\left\{1 + x_1, 2x_1 + x_2^{0.2} x_3^{-3.9}\right\} \\ + \left(0.1 x_1 x_3^{-0.5} + x_2^{1.7} x_3^{0.7}\right)^{1.5} \end{array} \right)^{1.7}$$

- What about the following inequality?:

$$x + y + z - \min\left\{\sqrt{xy}, (1 + xy)^{-0.3}\right\} \leq 0$$

GGP Example: Floor Planning

- The problem is to configure and place some rectangles in such a way they do not overlap:
  - objective: minimize the area of the bounding box
  - fixed areas for the four rectangles
  - upper and lower bounds on the aspect ratio
  - placement of rectangles as indicated in the figure



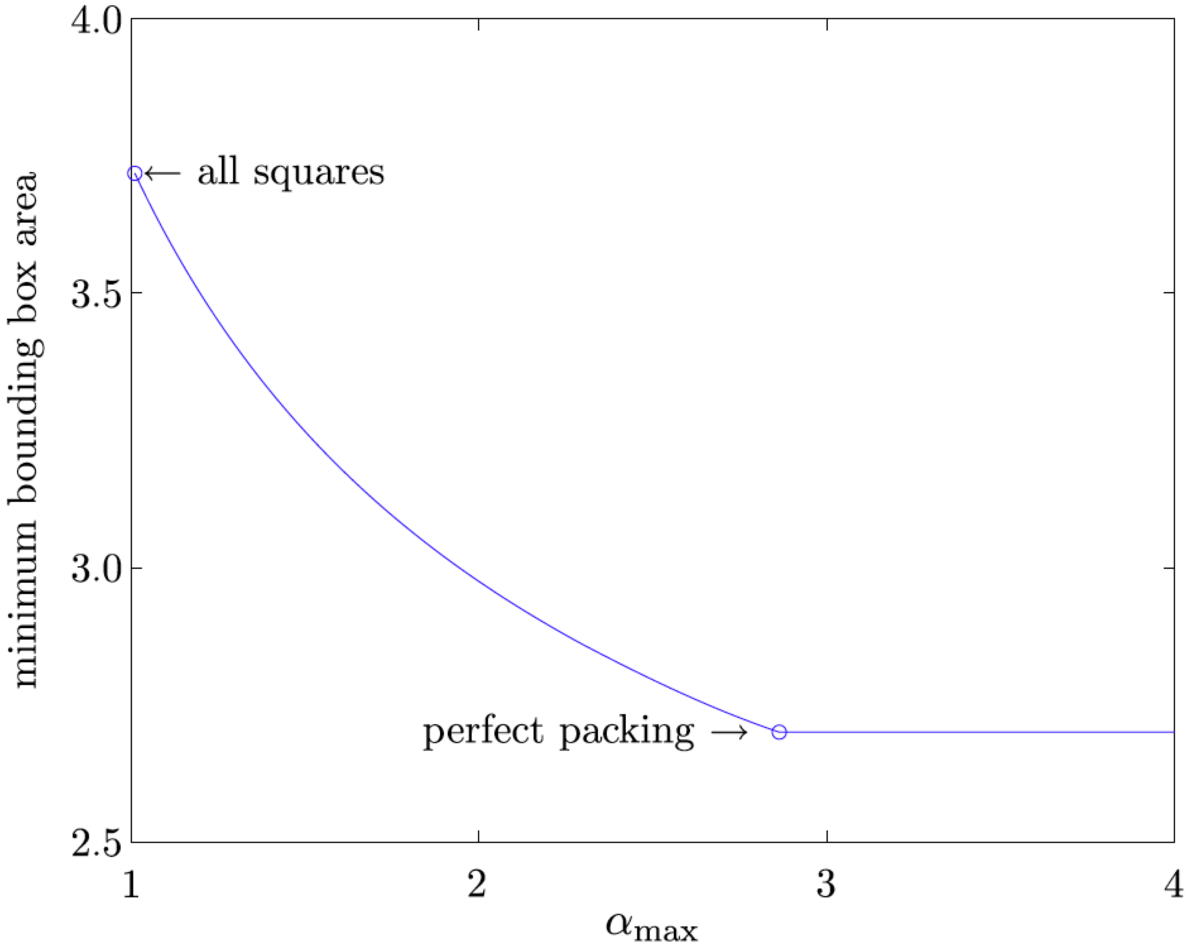
- We can identify the following terms:
  - width of the bounding box: $W = \max\{w_A + w_B, w_C + w_D\}$
  - height of the bounding box: $H = \max\{h_A, h_B\} + \max\{h_C, h_D\}$
  - area of the bounding box: WH
  - areas of rectangles: $h_i w_i$
  - aspect rations of rectangles: h_i/w_i

- Problem formulation:

$$\begin{array}{ll} \text{minimize} & \begin{array}{l} \max\{w_A + w_B, w_C + w_D\} \\ \qquad \times \left(\max\{h_A, h_B\} + \max\{h_C, h_D\}\right) \end{array} \\ \text{subject to} & h_A w_A = a, \quad h_B w_B = b, \quad h_C w_C = c, \quad h_D w_D = d \\ & 1/\alpha_{\max} \leq h_A/w_A \leq \alpha_{\max}, \quad \cdots \end{array}$$

which is a GGP.

- Numerical results: Optimal trade-off curves of minimum bounding box area versus maximum aspect ratio $\alpha_{\max}$:



GGP Example: Digital Circuit Gate Sizing

- Consider a digital circuit consisting of a set of *gates* (that perform logic functions) interconnected by wires.

- Each gate is to be sized; in the simplest case, each gate has a variable $x_i \geq 1$ associated with it, which gives a scale factor for the size of the transistors it is made from.

- The scale factors of the gates, which are the optimization variables, affect the total circuit area, the power consumed by the circuit, and the speed of the circuit.

- We can identify the following terms:
  - total circuit area:

$$A = \sum_{i=1}^{n} a_i x_i$$

    where a_i is the area of gate i with unit scaling
  - total power consumed by the circuit:

$$P = \sum_{i=1}^{n} f_i e_i x_i$$

    where f_i is the frequency of transition of the gate and e_i is the energy lost when the gate transitions
  - each gate has an input capacitance:

$$C_i = \alpha_i + \beta_i x_i$$

and driving resistance:

$$R_i = \gamma_i / x_i$$

– delay of a gate is the product of its driving resistance and the sum of the input capacitances:

$$D_i = \begin{cases} R_i \sum_j C_j & \text{for } i \text{ not an output gate} \\ R_i C_i^{\text{out}} & \text{for } i \text{ an output gate} \end{cases}$$

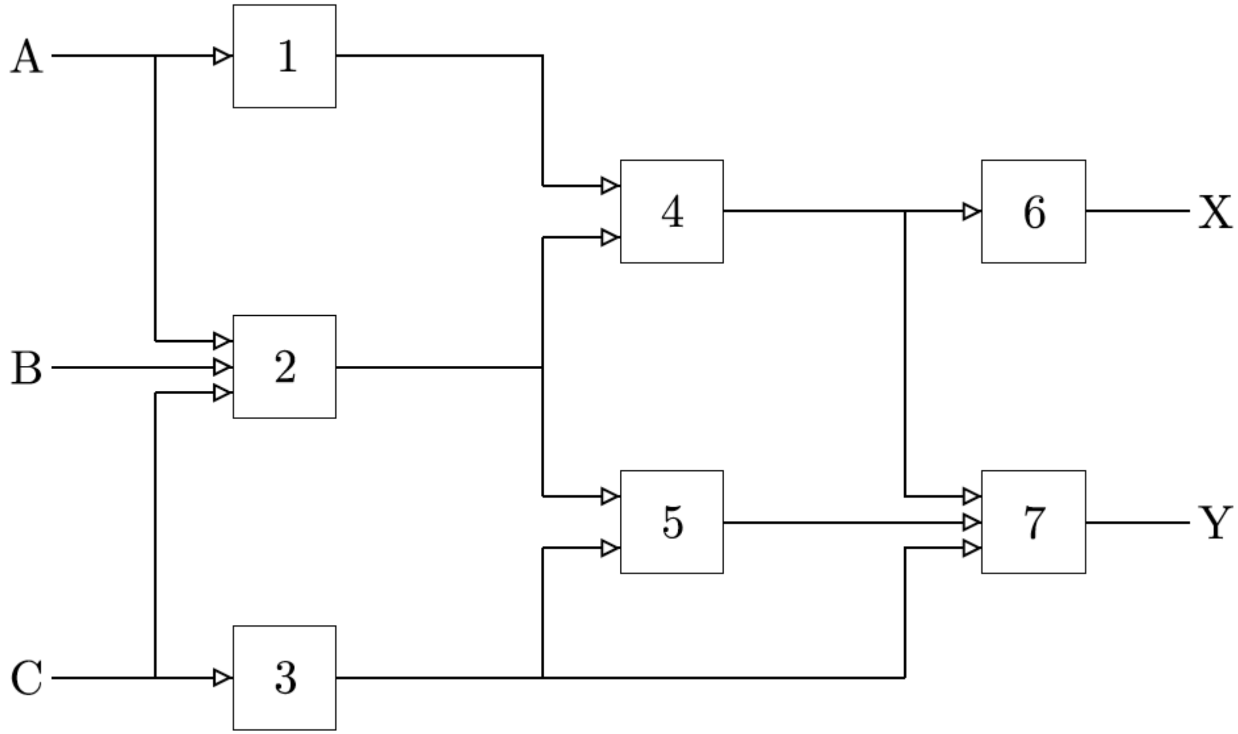
which is a posynomial function of the scale factors

– speed of the circuit is its maximum or worst-case delay D which is a generalized posynomial.

- Problem formulation:

$$
\begin{array}{lll}
\text{minimize} & D & \\
\text{subject to} & P \leq P^{\max}, & A \leq A^{\max} \\
 & x_i \geq 1, & i = 1, \cdots, n
\end{array}
$$

which is a GGP.



- The worst-case delay for the circuit in the figure is

$$D = \max \left\{ \begin{array}{c} D_1 + D_4 + D_6,\ D_1 + D_4 + D_7,\ D_2 + D_4 + D_6, \\ D_2 + D_4 + D_7,\ D_2 + D_5 + D_7,\ D_3 + D_5 + D_7 \end{array} \right\}$$

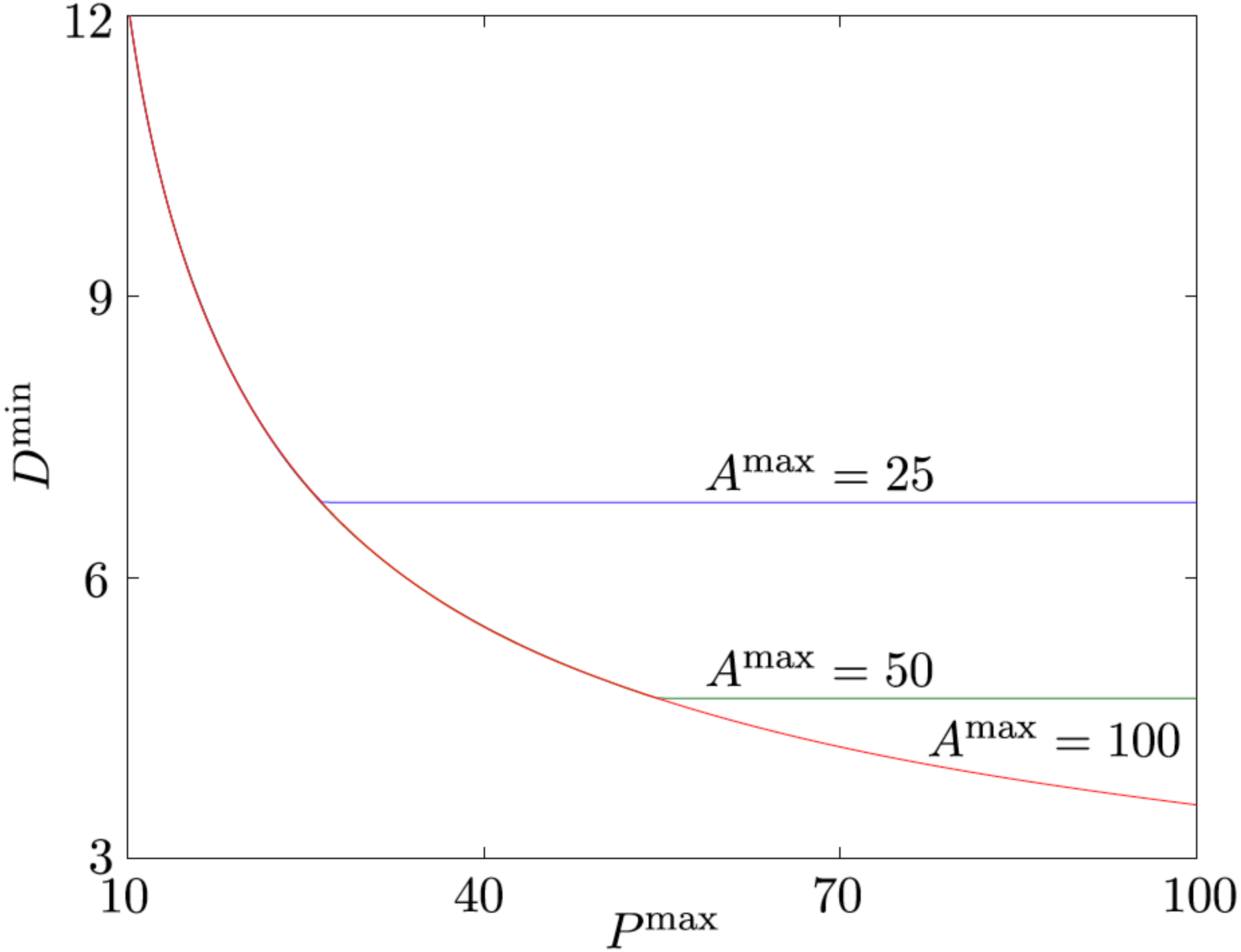
- In larger circuits, the number of paths can be very large and it is not practical to list all the possible paths like before.

- A simple recursion for D can be used to avoid enumerating all pahts: define T_i as the maximum delay over all paths that end with gate i and compute it via recursion.

- For the previous example, the recursion is

$$
\begin{aligned}
T_i &= D_i, \quad i = 1, 2, 3 \\
T_4 &= \max\{T_1, T_2\} + D_4 \\
T_5 &= \max\{T_2, T_3\} + D_5 \\
T_6 &= T_4 + D_6 \\
T_7 &= \max\{T_3, T_4, T_5\} + D_7 \\
D &= \max\{T_6, T_7\}.
\end{aligned}
$$

- Numerical results: Optimal trade-off curves of minimum delay $D^{\min}$ versus maximum power $P^{\max}$, for three values of maximum area $A^{\max}$:



More Transformations in GP: Function Composition

- We have already seen composition with the positive power function or the maximum function in generalized GP.

- It is possible to handle composition with many other functions.

- Composition with function $1/(1-z)$: the constraint

$$\frac{1}{1-q(x)} + f(x) \leq 1,$$

where f and q generalized posynomials (with the implicit constraint $q(x) < 1$), can be expressed as the two generalized posynomial constraints

$$t + f(x) \leq 1, \qquad q(x) + 1/t \leq 1.$$

- A more general variation: the constraint

$$\frac{p\left(x\right)}{r\left(x\right)-q\left(x\right)}+f\left(x\right)\leq 1,$$

where r is a monomial, p, q, and f are generalized posynomials (with the implicit constraint $q\left(x\right)<r\left(x\right)$), can be expressed as

$$t+f\left(x\right)\leq 1,\qquad q\left(x\right)+p\left(x\right)/t\leq r\left(x\right).$$

- Another example is the exponential function; one good approximation is

$$e^{f(x)}\approx\left(1+f\left(x\right)/a\right)^{a}$$

where a is large. If f is a generalized posynomial, then the right-hand side is a generalized posynomial.

More Transformations in GP: Additive log Terms

- Suppose we have a constraint of the form:

$$f(x) + \log q(x) \leq 1$$

  where f and q are generalized posynomials. (Note that the left-hand side can be negative.)

- We can use the approximation (valid for large a)

$$\log u \approx a\left(u^{1/a} - 1\right)$$

  to obtain

$$f(x) + a\left(q(x)^{1/a} - 1\right) \leq 1$$

  and then the generalized posynomial inequality

$$f(x) + aq(x)^{1/a} \leq 1 + a.$$

Power Control via GP

- Consider a wireless network with n logical transmitter/receiver pairs.
- The power received from transmitter j at receiver i is given by $G_{ij}p_j$ where G_{ij} includes the path gain and fading.
- The SINR for the receiver on the ith link is

$$\mathsf{sinr}_i = \frac{p_i G_{ii}}{\sum_{j \neq i} p_j G_{ij} + \sigma_i^2}.$$

- Constraints:
  - we require the SINR for all links to be larger than a threshold: $\mathsf{sinr}_i \geq \mathsf{sinr}_i^{\min}$
  - we impose limits on the transmitted powers: $p_i^{\min} \leq p_i \leq p_i^{\max}$.

- Problem formulation (to minimize the total power):

$$\begin{array}{lll} \underset{\mathbf{p}}{\text{minimize}} & p_1 + \cdots + p_n & \\ \text{subject to} & \frac{p_i G_{ii}}{\sum_{j \neq i} p_j G_{ij} + \sigma_i^2} \geq \mathsf{sinr}_i^{\min} & i = 1, \cdots, n \\ & p_i^{\min} \leq p_i \leq p_i^{\max}. & \end{array}$$

- This problem is not a GP, but can be rewritten as a GP taking the inverse of the SINR constraints:

$$\frac{\sum_{j \neq i} p_j G_{ij} + \sigma_i^2}{p_i G_{ii}} \leq 1/\mathsf{sinr}_i^{\min}$$

which are valid posynomial constraints.

- However, we are "killing flies with canons" here as this problem is in fact an LP: linear objective, linear power bounds, and the SINR constraints can be written as:

$$\sum_{j \neq i} p_j G_{ij} + \sigma_i^2 \leq p_i G_{ii} \left(1/\mathsf{sinr}_i^{\min}\right).$$

- We can add constraints to control the near-far problem:

$$p_{i_1} G_{i_1} = p_{i_2} G_{i_2}$$

  which are also linear.

- Alternatively, we could maximize the SINR of a particular user i^* as a GP:

$$\begin{array}{ll} \underset{\mathbf{p}}{\text{minimize}} & \frac{\sum_{j \neq i^*} p_j G_{i^* j} + \sigma_{i^*}^2}{p_{i^*} G_{i^* i^*}} \\ \text{subject to} & \text{(other constraints)}. \end{array}$$

- Another interesting formulation related to fairness is to maximize the minimum SINR:

$$\begin{array}{ll} \underset{\mathbf{p}}{\text{maximize}} & \min_i \frac{p_i G_{ii}}{\sum_{j \neq i} p_j G_{ij} + \sigma_i^2} \\ \text{subject to} & \text{(other constraints)} \end{array}$$

or, in epigraph form,

$$\begin{array}{lll} \underset{t,\mathbf{p}}{\text{maximize}} & t & \\ \text{subject to} & t \leq \frac{p_i G_{ii}}{\sum_{j \neq i} p_j G_{ij} + \sigma_i^2} & \forall i \\ & \text{(other constraints).} & \end{array}$$

- It can be rewritten as a GP:

$$\begin{array}{lll} \underset{t,\mathbf{p}}{\text{minimize}} & t^{-1} & \\ \text{subject to} & \frac{\sum_{j \neq i} t p_j G_{ij} + t \sigma_i^2}{p_i G_{ii}} \leq 1 & \forall i \\ & \text{(other constraints).} & \end{array}$$

- Yet another important parameter is the rate:

$$R_i = \log\left(1 + \mathsf{sinr}_i\right).$$

- Maximizing the rate of the whole system $R = \sum_i R_i = \sum_i \log\left(1 + \mathsf{sinr}_i\right)$ is equivalent to

$$\begin{array}{ll} \underset{\mathbf{p}}{\text{maximize}} & \prod_i \left(1 + \mathsf{sinr}_i\right) \\ \text{subject to} & \text{(other constraints)}. \end{array}$$

- The difficulty with the maximization of the rate is that $1 + \mathsf{sinr}_i$ is a quotient of posynomials ...

- An engineering solution is to make the low-SNR approximation:

$$\log\left(1 + \mathsf{sinr}_i\right) \approx \log\left(\mathsf{sinr}_i\right).$$

- The problem then becomes

$$\begin{array}{ll} \underset{\mathbf{p}}{\text{minimize}} & \prod_i \left(1/\mathsf{sinr}_i\right) \\ \text{subject to} & \text{(other constraints)} \end{array}$$

which a GP because we know that the inverse of the SINR is a posynomial and so it the profuct of inverses of SINRs.

- We can also include individual rate constraints of the form $R_i \geq R_i^{\min}$; these constraints are easy as they can be rewritten as

$$1/\mathsf{sinr}_i \leq \frac{1}{e^{R_i^{\min}} - 1}$$

which is a valid posynomial inequality constraint.

- Yet another interesting type of constraints are outage probability constraints (modeling the channel gain as $G_{ij}F_{ij}$, where F_{ij} is a random variable with exponential distribution that models Rayleigh fading, and ignoring the noise):

$$\begin{aligned} P_{\mathsf{out},i} &= \mathsf{Pr}\left[\mathsf{sinr}_i \leq \mathsf{sinr}_i^{\mathsf{th}}\right] = \mathsf{Pr}\left[p_i G_{ii} \leq \mathsf{sinr}_i^{\mathsf{th}} \sum_{j \neq i} p_j G_{ij}\right] \\ &= 1 - \prod_{j \neq i} \frac{1}{1 + \frac{\mathsf{sinr}_i^{\mathsf{th}} p_j G_{ij}}{p_i G_{ii}}} \end{aligned}$$

and then we can write constraints like $P_{\mathsf{out},i} \leq P_{\mathsf{out},i}^{\max}$ as

$$\prod_{j \neq i} \left(1 + \frac{\mathsf{sinr}_i^{\mathsf{th}} p_j G_{ij}}{p_i G_{ii}}\right) \leq \frac{1}{1 - P_{\mathsf{out},i}^{\max}}$$

which is a valid posynomial inequality constraint.

Complementary GP and Signomial Programming

- So far, all the considered constraints and objectives can be formulated as a GP except for the maximization of the total rate that requires the approximation $\log(1 + \mathsf{sinr}_i) \approx \log(\mathsf{sinr}_i)$.

- However, for medium-to-low SINR, the approximation does not hold and we have to deal with a ratio of posynomials:

$$\begin{array}{ll} \underset{\mathbf{p}}{\text{minimize}} & \prod_i \frac{1}{1+\mathsf{sinr}_i} = \prod_i \frac{\sum_{j \neq i} p_j G_{ij} + \sigma_i^2}{\sum_j p_j G_{ij} + \sigma_i^2} \\ \text{subject to} & \text{(other constraints)}. \end{array}$$

- Minimizing or upper bounding a ratio of posynomials is a truly nonconvex problem that is intrinsically intractable.

- This leads to
  - *complementary GP*: allows upper bound constraints on the ratio between posynomials.
  - *signomial programming*: involving signomials, which are like posynomials but possibly with negative coefficients.

- A signomial can be expressed as the difference of two posynomials, $f(x) - g(x) \leq 1$, which can always be rewritten as a ratio of two posynomials:
$$\frac{f(x)}{1+g(x)} \leq 1.$$

- Similarly, a ratio between posynomials $f(x)/g(x) \leq 1$ can always be rewritten as a signomial:
$$1 + f(x) - g(x) \leq 1.$$

- One way to deal with these problems is with a successive convex approximation, known as *condensation method* in operations research:
  - single condensation method: approximate the denominator of a ratio between two posynomials as a monomial (it becomes then a posynomial)
  - double condensation method: approximate both numerator and denominator (the constraint becomes linear!).

- Successive approximation:
  1. Start with $k = 0$.
  2. Approximate difficult function $f(x)$ with simple function $\tilde{f}(x)$ (typically convex) around some nominal point $x^{(k)}$.
  3. Solve the approximate (and simple) problem with solution $x^{(k+1)}$.
  4. Go to step 2 to refine the approximation around the new point.

Monomial Approximation of a Posynomial

- Recall the arithmetic/geometric mean inequality:

$$\sum_i \alpha_i v_i \geq \prod_i v_i^{\alpha_i}$$

  where $\mathbf{v} > 0$, $\boldsymbol{\alpha} \geq 0$, and $\mathbf{1}^T \boldsymbol{\alpha} = 1$.

- Define $u_i = \alpha_i v_i$ and write

$$\sum_i u_i \geq \prod_i \left(\frac{u_i}{\alpha_i} \right)^{\alpha_i}.$$

- Now, let $\{u_i(x)\}$ be the monomial terms in a posynomial $f(x) = \sum_i u_i(x)$ to finally get the desired result:

$$f(x) = \sum_i u_i(x) \geq \prod_i \left(\frac{u_i(x)}{\alpha_i}\right)^{\alpha_i}$$

with equality for x such that the terms $u_i(x)/\alpha_i$ are equal. One way to force equality at one particular nominal point x_0 is by choosing

$$\alpha_i = \frac{u_i(x_0)}{f(x_0)}.$$

- With the previous result, we can then substitute a lower bound inequality on a posynomial

$$f(x) = \sum_i u_i(x) \geq t$$

by a tightened lower inequality on a monomial

$$\prod_i \left(\frac{u_i(x)}{\alpha_i} \right)^{\alpha_i} \geq t$$

which we can rewrite as a valid monomial inequality

$$\prod_i \left(\frac{u_i(x)}{\alpha_i} \right)^{-\alpha_i} \leq t^{-1}.$$

- As an example, we can now deal with an upper bound inequality constraint of a ratio between posynomials

$$\frac{f(x)}{g(x)} \leq 1$$

by first rewritting it as

$$\frac{f(x)}{t} \leq 1, \qquad t \leq g(x)$$

and then using the monomial approximation for the constraint $t \leq g(x)$.

Approximation by Monomials and Posynomials

- We have seen how to approximate a posynomial by a monomial. We will address now a more general question.

- Can we approximate an arbitrary function $f(x)$ with a monomial or posynomial?

- Let $F(y) = \log f(e^y)$. Then,
  - f can be approximated by a monomial if and only if F can be approximated by an affine function
  - f can be approximated by a posynomial if and only if F can be approximated by a convex function.

Local Monomial Approximation

- Let's find a local monomial approximation of a general function f around the point x.

- In other words, we want to find an affine approximation of $F(y) = \log f(e^y)$ around $y = \log x$.

- This is given by the first-order Taylor approximation of F:

$$F(z) \approx F(y) + \sum_{i=1}^{n} \frac{\partial F}{\partial y_i}(z_i - y_i)$$

which is valid for $z \approx y$.

- Using

$$\frac{\partial F}{\partial y_i} = \frac{1}{f\left(e^y\right)} \frac{\partial f\left(e^y\right)}{\partial y_i} = \frac{1}{f\left(e^y\right)} \frac{\partial f}{\partial x_i} e^{y_i} = \frac{x_i}{f\left(e^y\right)} \frac{\partial f}{\partial x_i}$$

we can write the Taylor approximation (after exponentiating) as

$$e^{F(z)} \approx e^{F(y)} \prod_{i=1}^{n} e^{\frac{\partial F}{\partial y_i}(z_i - y_i)}$$

or

$$f\left(e^z\right) \approx f\left(e^y\right) \prod_{i=1}^{n} e^{\frac{x_i}{f\left(e^y\right)} \frac{\partial f}{\partial x_i}(z_i - y_i)}.$$

Defining $w_i = e^{z_i}$ we finally have

$$f(w) \approx f(x) \prod_{i=1}^{n} \left(\frac{w_i}{x_i} \right)^{a_i}$$

where $a_i = \frac{x_i}{f(e^y)} \frac{\partial f}{\partial x_i}$.

Summary

- We have introduced GP as a class of nonconvex problems that can be transformed into convex form ones and, hence, optimally solved.

- We have seen several applications: floor planning, gate sizing in circuit design, and power control.

- We have also learned variations of GP termed Generalized GP as well as Complementary GP and Signomial Programming.

- Some applications cannot be written as GP but still can be well approximated by GP and then successive approximation methods can be used.

References

- Stephen Boyd, Seung-Jean Kim, Lieven Vandenberghe, and Arash Hassibi, "A tutorial on geometric programming," *Optim. Eng.*, Springer, vol. 8, April 2007.

- Mung Chiang, "Geometric programming for communication systems," *Foundations and Trends in Communications and Information Theory*, Now Publishers, vol. 2, no. 1-2, Aug. 2006.

- Mung Chiang, Chee Wei Tan, Daniel P. Palomar, Daniel O'Neill, and David Julian, "Power Control by Geometric Programming," *IEEE Trans. on Wireless Communications*, vol. 6, no. 7, pp. 2640-2651, July 2007.

- S. Kandukuri and Stephen Boyd, "Optimal power control in interference-limited fading wireless channels with outage-probability specifications," *IEEE Trans. on Wireless Communications*, vol. 1, no. 1, pp. 46–55, 2002.