6. Separable Programming

Consider a general *NLP*

min $f(\mathbf{x})$
s.t. $g_j(\mathbf{x}) \le b_j$, $j = 1, \ldots, m$.

Definition 6.1. The *NLP* is a separable program if its objective function and all constraints are consisted of separable functions, i.e.,

$$f(\mathbf{x}) = \sum_{i=1}^{n} f_i(x_i), \quad \text{and} \quad \sum_{i=1}^{n} g_{ji}(x_i) \le b_j, \qquad j = 1, \ldots, m;$$

and all x_i are non-negative variables bounded above, i.e., $0 \le x_i \le \mu_i$ for some μ_i, $i = 1, \ldots, n$. ✧✧

The technique *separable programming* basically replaces all separable functions, in objectives and constraints, by piecewise linear functions.

Definition 6.2. A *convex program* is an *NLP* that minimizes a convex function or maximizes a concave function over a convex set. ✧✧

Fact 6.1. Any (continuous) convex function can be approximated to any degree of accuracy by a piecewise linear convex function. ✧✧

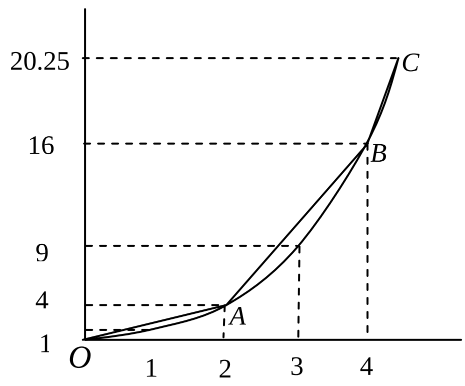
From Fact 6.1, when f_j and g_{ij} are convex functions, they can be approximated to any degree of accuracy by piecewise linear functions. Eventually, such an *NLP* of piecewise convex functions can be represented by a linear program (*LP*). Thus, effectively a separable convex program can be approximated by a sequence of *LP*s to any degree of accuracy.

Note also that when f_j and g_{ij} are convex, an local minimum is in fact a global minimum.

Example 6.1. **NLP1** is:

min $x_1^2 - 2x_1 - x_2$

s.t. $x_1 + 2x_2 \le 5$,
$2x_1 + x_2 \le 9$,
$x_1, x_2 \ge 0$.



The above is a separable program with $f_1(x_1) =$

$x_1^2 - 2x_1$ and $f_2(x_2) = -x_2$. f_2 is linear. The problem is easy to solve if f_1, i.e., x_1^2 is approximated by a linear function.

From the constraints, $x_1 \le 4.5$. Let us approximate the non-linear function $y = x_1^2$ by a piecewise linear function. For simplicity, we take a function of three linear pieces, with break points at 0, 2, 4, and 4.5. The same procedure can be applied to any number of break points at any values.

points	*O*	*A*	*B*	*C*
x_1	0	2	4	4.5
y	0	4	16	20.25

There are two ways, the λ- and the δ-forms, to represent the function in piecewise linear form.

Definition 6.3. **(λ-form)** For any point within a linear segment, its functional value is the convex combination of the values of the two break points of the linear segment. Let $\lambda_i \ge 0$ be the weight of break point i, $i = O, A, B$, and C.

$$\begin{cases} x_1 = 0\lambda_O + 2\lambda_A + 4\lambda_B + 4.5\lambda_C, & (1) \\ y = 0\lambda_O + 4\lambda_A + 16\lambda_B + 20.25\lambda_C, & (2) \\ \lambda_O + \lambda_A + \lambda_B + \lambda_C = 1, & (3) \\ \text{at most two adjacent } \lambda_i \text{ take non-zero values.} & (4) \end{cases}$$

Now reformulate **NLP1** into

NLP2

$$\begin{aligned} \min \quad & y - 2x_1 - x_2 \\ \text{s.t.} \quad & x_1 + 2x_2 \le 5, \\ & 2x_1 + x_2 \le 9, \\ & (1), (2), (3), \text{ and } (4), \\ & x_1, x_2, \lambda_i, i = O, A, B, C \ge 0. \end{aligned}$$

✧✧

Constraints (4) can be omitted if **NLP 1** is a convex program. Check that whenever (4) is violated by a set of λ_i, the value of the corresponding y is above the three-piece linear function, and hence the set of λ_i cannot be a minimum point.

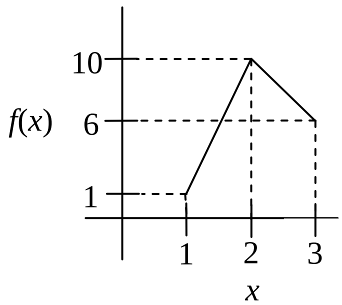
In general, if the NLP is a non-convex program, constraints (4) are needed, though they are implemented implicitly through the separable programming extension of the simplex method. The pivoting step of the separable programming extension simply ensures that at any time at most two adjacent λ_i are in the basis. It can be shown that such a restricted basis rule will lead to the optimum solution.

The idea of separable program is applicable to non-convex programs. Of course, in those cases, the optimum solution can be a local rather than global optimum.

Example 6.2. (Non-Convex problem)

min $f(x)$,

s.t. $1 \le x \le 3$.



$f(x)$ is a piecewise linear function (which is not linear by itself). It is obvious that minimum is $x^* = 1$ with $f(x^*) = 1$. Let us find this by the idea of separable programming.

Let $x = \lambda_0 + 2\lambda_A + 3\lambda_B$ and $y = \lambda_0 + 10\lambda_A + 6\lambda_B$; and add in the constraint $\lambda_0 + \lambda_A + \lambda_B = 1$. We take the special attention of (4) that at most two λ's can be positive at any time.

The problem becomes

min $\lambda_0 + 10\lambda_A + 6\lambda_B$,

s.t. $\lambda_0 + 2\lambda_A + 3\lambda_B \le 3$,

$\lambda_0 + 2\lambda_A + 3\lambda_B \ge 1$,

$\lambda_0 + \lambda_A + \lambda_B = 1$,

$\lambda_0, \lambda_A, \lambda_B \ge 0$, and at most two adjacent λ's can be positive at the same time.

After adding in slack variable s, surplus variable u, and two artificial variables a_1 and a_2, the problem becomes

min $\lambda_0 + 10\lambda_A + 6\lambda_B + Ma_1 + Ma_2$,

$$\begin{aligned} s.t.\quad & \lambda_0 + 2\lambda_A + 3\lambda_B + s && = 3, \\ & \lambda_0 + 2\lambda_A + 3\lambda_B \quad - u + a_1 && = 1, \\ & \lambda_0 + \lambda_A + \lambda_B \quad + a_2 && = 1, \\ & \lambda_0, \lambda_A, \lambda_B, s, a_1, a_2 \geq 0, \end{aligned}$$

and at most two adjacent λ's can be positive at the same time.

	λ_0	λ_A	λ_B	s	u	a_1	a_2	RHS
	1	10	6	0	0	M	M	0
s	1	2	3	1	0	0	0	3
a_1	1	2	3	0	-1	1	0	1
a_2	1	1	1	0	0	0	1	1

	λ_0	λ_A	λ_B	s	u	a_1	a_2	RHS
	$1-2M$	$10-3M$	$6-4M$	0	M	0	0	$-2M$
s	1	2	3	1	0	0	0	3
a_1	1	2	3	0	-1	1	0	1
a_2	1	1	1	0	0	0	1	1

	λ_0	λ_A	λ_B	s	u	a_1	a_2	RHS
	$\frac{-5-2M}{3}$	$\frac{18-M}{3}$	0	0	2	$\frac{-6+4M}{3}$	0	$-2-\frac{2M}{3}$
s	0	0	0	1	1	-1	0	2
λ_B	$\frac{1}{3}$	$\frac{2}{3}$	1	0	$-\frac{1}{3}$	$\frac{1}{3}$	0	$\frac{1}{3}$
a_2	$\frac{2}{3}$	$\frac{1}{3}$	0	0	$\frac{1}{3}$	$-\frac{1}{3}$	1	$\frac{2}{3}$

Supposedly λ_0 is the most negative. However, since λ_B is in the basis, only λ_A among the λ's is qualified to be in the basis.

	λ_0	λ_A	λ_B	s	u	a_1	a_2	RHS
	$\frac{-8-M}{2}$	0	$\frac{-18+M}{2}$	0	$\frac{10-M}{2}$	$\frac{-10+3M}{2}$	M	$\frac{-10-M}{2}$
s	0	0	0	1	1	-1	0	2
λ_A	$\frac{1}{2}$	1	$\frac{3}{2}$	0	$-\frac{1}{2}$	$\frac{1}{2}$	0	$\frac{1}{2}$
a_2	$\frac{1}{2}$	0	$-\frac{1}{2}$	0	$\frac{1}{2}$	$-\frac{1}{2}$	1	$\frac{1}{2}$

	λ_0	λ_A	λ_B	s	u	a_1	a_2	RHS
	0	8+M	3+2M	0	1–M	-1+2M	0	1
s	0	0	0	1	1	-1	0	2
λ_0	1	2	3	0	-1	1	0	1
a_2	0	-1	-2	0	1	-1	1	0

	λ_0	λ_A	λ_B	s	u	a_1	a_2	RHS
	0	9	5	0	0	M	0	1
s	0	1	2	1	0	0	-1	2
λ_0	1	1	1	0	0	0	1	1
u	0	-1	-2	0	1	-1	1	0

✧✧

Definition 6.4. (**δ-form)** Instead of taking weighted value of break points, we can add up the contribution from each linear segment. Let δ_i be the proportion of the ith segment taken, i = OA, AB, BC.

$$\begin{cases} x_1 = 2\delta_{OA} + 2\delta_{AB} + 0.5\delta_{BC}, & (5) \\ y = 4\delta_{OA} + 12\delta_{AB} + 4.25\delta_{BC}, & (6) \\ 0 \le \delta_{OA}, \delta_{AB}, \delta_{BC} \le 1, & (7) \\ \text{there exists } \delta_j > 0 \text{ such that } \delta_i = 1 \text{ for } i < j \text{ and } \delta_i = 0 \text{ for } i > j. & (8) \end{cases}$$

As for the λ-form, the δ-form may also give a local optimum if the original program is non-convex. When the original program is convex, constraint (8) is not necessary.

Remark 6.1. To get more accurate result, the piecewise linear approximation of f_i can be refined with more linear segments. There are studies on segment refinement to get the best trade off between accuracy and computational effort. ✧✧

Remark 6.2. It is possible to approximate constraints by similar procedure. ✧✧

Remark 6.3. A product term x_1x_2 can be transformed to separable form. Let $s_1 = (x_1+x_2)/2$ and $s_2 = (x_1-x_2)/2$. Then $x_1x_2 = s_1^2 - s_2^2$. ✧✧