Operations Research Seminar: Nonlinear Programming
Speaker: Kevin Jurcik

1 Introduction

To this point, we've only considered problems that can be modeled as a linear objective function with linear constraints. While a vast number of problems can be modeled in this way, there are also many problems that cannot. For instance, when minimizing costs for a company, the shipping cost may change when a large enough number of units are being shipped, rather than a single fixed shipping cost. Production rates may have a special learning property, where we can produce an additional good at a variable cost. Indeed, there are many problems for which we cannot assume linearity of our objective function or constraints.

We take a step back from the linear programming standard form to which we have accustomed ourselves and consider a more general optimization problem:

$$\begin{array}{rll} \text{Maximize:} & f(\mathbf{x}) & \\ \text{Subject to:} & g_i(\mathbf{x}) \ \leq b_i & \quad i = 1, 2, ..., m. \end{array}$$

where $\mathbf{x} \in \mathbb{R}^n$, and $f(\mathbf{x}), g_i(\mathbf{x})$ are real valued functions from $\mathbb{R}^n$ to $\mathbb{R}$ for $i = 1, 2, ..., m$.

2 Graphical Illustration of Problems

2.1 Example Problems in 2 Dimensions

Similarly to linear problems, it is quite helpful to look at the graphical representations of some examples in two variables to get an intuition about the problems. We now look at some helpful examples from [1] that shed some light on how a problem is altered by relaxing some of its linearity assumptions. The problems presented here are found in some way by changing the linear program given by

$$\begin{array}{rrrl} \text{Maximize:} & 3x_1 & +5x_2 & = z \\ \text{Subject to:} & x_1 & & \leq 4 \\ & & x_2 & \leq 6 \\ & 3x_1 & +2x_2 & \leq 18 \\ & x_1, & x_2 & \geq 0. \end{array}$$

Figure 1 shows the feasible region of this problem as well as some typical isoprofit lines.

Now we consider how the problem changes when we replace the second and third constraints with the constraint $9x_1^2 + 5x_2^2 \leq 216$, shown in Figure 2. We note that the optimal solution remains the same, but it is no longer a corner point of the feasible region. The important fact to note here is that we no longer have that optimal solutions are limited to corner point feasible solutions, an important fact that we used with previous methods of solving LPs.

Next we consider the problem that has the original feasible region but with the new objective function given by $126x_1 - 9x_1^2 + 182x_2 - 13x_2^2$. Again we note from Figure 3 that the optimal solution need not be a corner point of the feasible region, even if the feasible region is given by linear inequalities.

Now consider again a problem with the original feasible region and the objective function given by $54x_1 - 9x_1^2 + 78x_2 - 13x_2^2$. Figure 4 reveals that the optimal solution for the problem is the point (3,3), which lies *inside* the feasible region. This shows that the optimal solution of a nonlinear programming problem need not even be a point on the boundary of the feasible region.

In our last example, we again replace the second and third constraints of the original problem, this time with $8x_1 - x_1^2 + 14x_2 - x_2^2 \leq 49$. We see in Figure 5 that (0,7) is the optimal solution. More importantly, if we use an algorithm on this problem that seeks to move from a given point in a direction that increases the objective value, then the algorithm may conclude that the point (4,3) is optimal (since moving to any nearby feasible solutions will decrease the objective value). So, a general method for solving nonlinear problems must be able to distinguish between local and global maxima.

2.2 Convex and Concave Functions

Algorithms for solving nonlinear problems generally are unable to distinguish between local and global maxima. Recall from calculus that if we have a doubly differentiable function f, then we can guarantee a local maximum of f is a global maximum of f if we have $\frac{d^2f}{dx^2} \leq 0$ for all x. Such a function is called a **concave** function. Similarly, if $\frac{d^2f}{dx^2} \geq 0$ for all x, then we say that f is a **convex** function. Note that linear functions are both convex and concave.

We can extend the concept of concave and convex functions to multivariable functions by checking conditions imposed on the partial derivatives of the function. We will not go into any detail about how to determine if a function is concave, convex, or neither, but some special types of nonlinear programming methods rely on these properties.

3 Types of Nonlinear Programming Problems

3.1 Unconstrained Optimization

Unconstrained problems require the maximization of an objective function and have no constraints. These problems are important in that many methods of solving other types of nonlinear problems start by transforming the problem into an unconstrained problem. We will learn how to solve some special cases of unconstrained problems in the next section.

3.2 Linearly Constrained Optimization

A linearly constrained problem is a problem with a nonlinear objective function and linear constraints. A number of algorithms for solving these problems extend the simplex method.

3.3 Quadratic Programming

Quadratic programming problems again have linear constraints, but require the objective function $f(\mathbf{x})$ to be quadratic. A common approach to solving gerenal linearly constrainted problems is to solve a sequence of quadratic programming approximations.

3.4 Convex Programming

Convex programming problems require two special properties of the functions it involves. First, the objective function must be concave. Second, each constraint must be a convex function. These assumptions ensure that any local maximum is a global maximum.

3.5 Separable Programming

Separable programming is a special case of convex programming that requires one additional property – the objective function and constraints must be separable functions. That is, each term of each function involves only one variable, so that each function can be rewritten as the sum of functions in one variable.

3.6 Nonconvex Programming

Nonconvex programming includes all problems that do not satisfy the requirements of a convex programming problem. Many methods for these problems settle on obtaining local maxima, as no algorithm can find the global maxima for all nonconvex problems.

4 One-Variable Unconstrained Optimization

We discuss how to solve the previously mentioned problems by starting with the simplest case – unconstrained optimization in a single variable x, where the differentiable function to be maximized, $f(x)$, is concave. Recall from calculus that a solution x^* is optimal (a global maximum) if and only if $\frac{df}{dx} = 0$ at $x = x^*$. In some cases, we can solve this equation to find our optimal value, but for slightly more complex functions, we must use a different method.

4.1 The One-Dimensional Search Procedure

The idea behind the one-dimensional search procedure is as follows. First, we find the rate of change, $m = \frac{df}{dx}$ at $x = x'$ for some value x'. If $m \geq 0$, then $x' \leq x^*$ (since f is concave). If $m \leq 0$, then $x' \geq x^*$. By updating our trial solution x' appropriately and applying this test repeatedly, then we can find a sequence of solutions that converge to the optimal solution x^*. Notice that since we are finding a sequence of solutions that converge to x^*, we may not find x^* exactly, and thus need some sort of error tolerance for this method.

First, we give some notation:

- Let x' denote our current trial solution.
- Let x_l denote a lower bound on x^*.
- Let x_u denote an upper bound on x^*.
- Let ϵ denote the error tolerance for x^*.

Then the one-dimensional search procedure is as follows.

Initialization: Select $\epsilon > 0$. Find an initial x_l and x_u by inspection (or by finding values of x at which the derivative is positive and negative, respectively). Let $x' = \frac{x_l + x_u}{2}$.

1. Let $m = \frac{df}{dx}$ at $x = x'$.
   (a) If $m \geq 0$, then $x_l \leftarrow x'$.
   (b) If $m \leq 0$, then $x_u \leftarrow x'$.
2. $x' \leftarrow \frac{x_l + x_u}{2}$

Stopping rule: If $x_u - x_l < 2\epsilon$, then x' is within ϵ of x^*; stop. Otherwise, perform another iteration.

4.2 Example

Suppose we are required to maximize the function $f(x) = 12x - 3x^4 - 2x^6$. The first two derivatives of f are

$$\frac{df(x)}{dx} = 12(1 - x^3 - x^5)$$
$$\frac{d^2f(x)}{dx^2} = -12(3x^2 + 5x^4).$$

We have that the second derivative is nonpositive for all values of x, so f is concave and we may use the one-dimensional search procedure. We'll choose $\epsilon = 0.05$ to be our error tolerance. We check the slope of the function at $x = 0$ and $x = 1$ and find that

$f'(0) > 0, f'(1) < 0$, so we let $x_l = 0, x_u = 1$ be our initial lower and upper bounds on the optimal solution respectively.

This means that our first trial solution is $\frac{0+1}{2} = \frac{1}{2}$. The slope at this point is $f'(\frac{1}{2}) = \frac{81}{8} > 0$, so we let $x_l = \frac{1}{2}$. $1 - \frac{1}{2} = 0.5 > 2\epsilon$, so we continue the procedure. Our new trial solution is $x' = \frac{3}{4}$.

The slope at x' is $f'(\frac{3}{4}) \approx 4.09 > 0$, so we let $x_l = \frac{3}{4}$. We have $x_u - x_l = 1.25$, so we continue. Our new trial solution is $x' = \frac{7}{8}$.

The slope at x' is $f'(\frac{7}{8}) \approx -2.19 < 0$, so we let $x_u = \frac{7}{8}$. We now have that $x_u - x_l = 0.875 < 2\epsilon$, so conclude that the next trial solution $x' = \frac{13}{16}$ is within ϵ of the optimal solution x^* and stop.

5 Multivariable Unconstrained Optimization

We extend the previous problem to the case in multiple variables (say $\mathbf{x} = (x_1, ..., x_n)$). Suppose again that $f(\mathbf{x})$ is concave and differentiable. We get a similar condition for optimality; namely, x^* is an optimal solution if $\frac{\partial f}{\partial x_j} = 0$ for $j = 1, 2, ..., n$.

If it is the case that each $\frac{\partial f}{\partial x_j}$ is linear in the remaining variables, then we can solve the system of n equations to get our optimal solution x^*. If not, we must apply a more clever method to find our solution.

Recall that when dealing with one variable, when faced with a trial solution, we selected only one of two directions (increase x or decrease x) to move based on the derivative. In multiple variables, we have innumerably many directions in which to move.

Recall from calculus the **gradient** of f, denoted $\nabla f(\mathbf{x})$, where $\nabla f(\mathbf{x}') = (\frac{\partial f}{\partial x_1}, ..., \frac{\partial f}{\partial x_n})$ at $\mathbf{x}' = \mathbf{x}$. As a result from calculus, the (infinitesimal) change in $\mathbf{x}$ that maximizes the rate at which $f(\mathbf{x})$ increases is proportional to $\nabla f(\mathbf{x})$ ([1]). So, it would be wise to try to move in the direction of $\nabla f(\mathbf{x})$ as much as possible.

5.1 The Gradient Search Procedure

It would be ideal to move in the direction of $\nabla f(\mathbf{x})$ continuously, but to do so we would need to constantly re-evaluate the value of $\nabla f(\mathbf{x})$. Instead, when starting from a point $\mathbf{x}'$, we choose to follow the direction $\nabla f(\mathbf{x}')$ until doing so will no longer give us any benefit (that is, until it no longer increases $f(\mathbf{x})$). When we reach this point, we reset our trial solution $\mathbf{x}'$ and find the gradient at this point to find the new direction we will follow.

In other words, an iteration of this procedure involves updating $\mathbf{x}' = \mathbf{x}' + t^* \nabla f(\mathbf{x}')$, where t^* is the positive value of t that mazimizes $f(\mathbf{x}' + t\nabla f(\mathbf{x}'))$. Note here that this is maximization of a function in one variable, t, so we can use the one-dimensional search procedure to obtain the value t^* (or analytically, if possible).

We continue iterating until the solution that we have is "good enough". That is, given $\epsilon > 0$, we continue to iterate until $\left|\frac{\partial f}{\partial x_j}\right| < \epsilon$ for $j = 1, 2, ..., n$.

5.2 Example

Let $f(\mathbf{x}) = 2x_1x_2 + 2x_2 - x_1^2 - 2x_2^2$ be the function we require to be maximized. We will not set an error tolerence $\epsilon > 0$, but rather run a few iterations to get a feel for the procedure. It can be verified that $f(\mathbf{x})$ is concave. We have that

$$\frac{\partial f}{\partial x_1} = 2x_2 - 2x_1$$
$$\frac{\partial f}{\partial x_2} = 2x_1 + 2 - 4x_2.$$

We begin the gradient search procedure by choosing $\mathbf{x}' = (0,0)$ as our initial trial solution. Since the partial derivatives at this point are 0 and 2 respectively, the gradient at this point is $\nabla f(0,0) = (0,2)$.

So, for the first iteration, we must find the value t^* which maximizes

$$\begin{aligned} f(\mathbf{x}' + t\nabla f(\mathbf{x}')) &= f(0 + t(0), 0 + t(2)) \\ &= f(0, 2t) \\ &= 2(0)(2t) + 2(2t) - 0^2 - 2(2t)^2 \\ &= 4t - 8t^2. \end{aligned}$$

The function $4t - 8t^2$ is concave, so it follows that it is maximized when $\frac{d}{dt}(4t - 8t^2) = 4 - 16t = 0$. Solving for t, we get that $t^* = \frac{1}{4}$ maximizes this function.

So, we get a new trial solution $\mathbf{x}' = (0,0) + \frac{1}{4}(0,2) = (0, \frac{1}{2})$. For this new trial solution, the gradient is $\nabla f(0, \frac{1}{2}) = (1, 0)$. Thus, for the second iteration, we want to find t^* which maximizes

$$\begin{aligned} f(\mathbf{x}' + t\nabla f(\mathbf{x}')) &= f(0 + t(1), \tfrac{1}{2} + t(0)) \\ &= f(t, \tfrac{1}{2}) \\ &= 2(t)(\tfrac{1}{2}) + 2(\tfrac{1}{2}) - (t)^2 - 2(\tfrac{1}{2})^2 \\ &= t - t^2 + \tfrac{1}{2}. \end{aligned}$$

Again we have that $t - t^2 + \frac{1}{2}$ is concave, and that it is maximized when $\frac{d}{dt}(t - t^2 + \frac{1}{2}) = 1 - 2t = 0$. Solving for t, we get that our optimal solution is $t^* = \frac{1}{2}$.

Our new trial solution is $\mathbf{x}' = (0, \frac{1}{2}) + \frac{1}{2}(1, 0) = (\frac{1}{2}, \frac{1}{2})$, and the gradient at this point is $\nabla f(\frac{1}{2}, \frac{1}{2}) = (0, 1)$.

If we continue iterating, we will get the points $(\frac{1}{2}, \frac{3}{4})$, $(\frac{3}{4}, \frac{3}{4})$, $(\frac{3}{4}, \frac{7}{8})$, and so on, converging to (1,1) (but never reaching the point (1,1)). We can verify that (1,1) is the optimal solution since $\nabla f(1,1) = (0,0)$.

If we set an error tolerance $\epsilon > 0$, we will continue to iterate until have have a trial solution $\mathbf{x}'$ such that

$$\frac{\partial f}{\partial x_1} < \epsilon$$
$$\frac{\partial f}{\partial x_2} < \epsilon$$

when evaluated at $\mathbf{x} = \mathbf{x}'$.

6 References

[1] F. Hillier, G. Lieberman, *Introduction to Operations Research*, McGraw Hill, 2001

7 Figures (from [1])

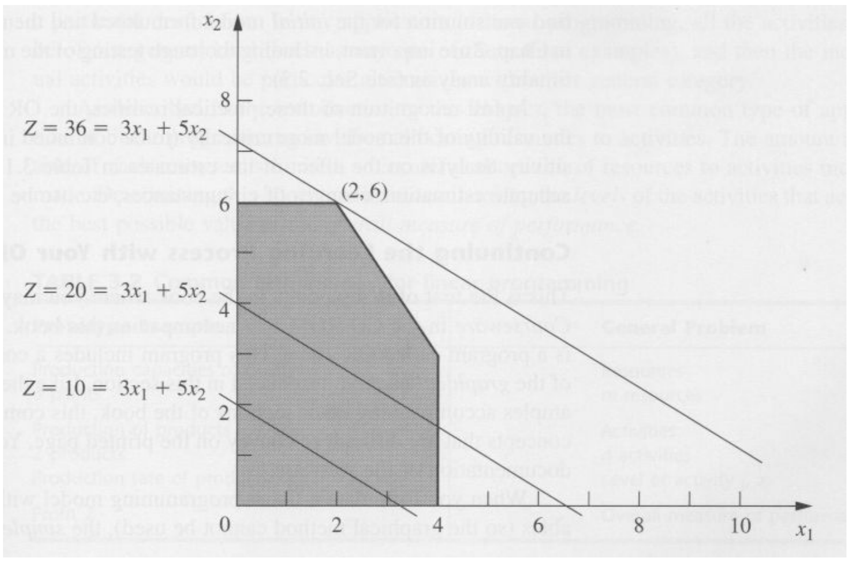

Figure 1

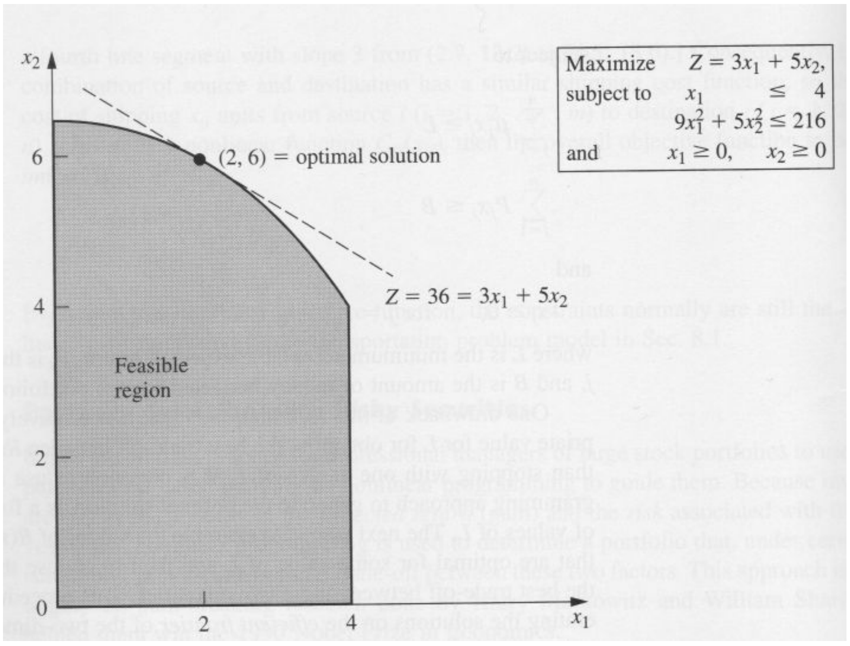

Figure 2

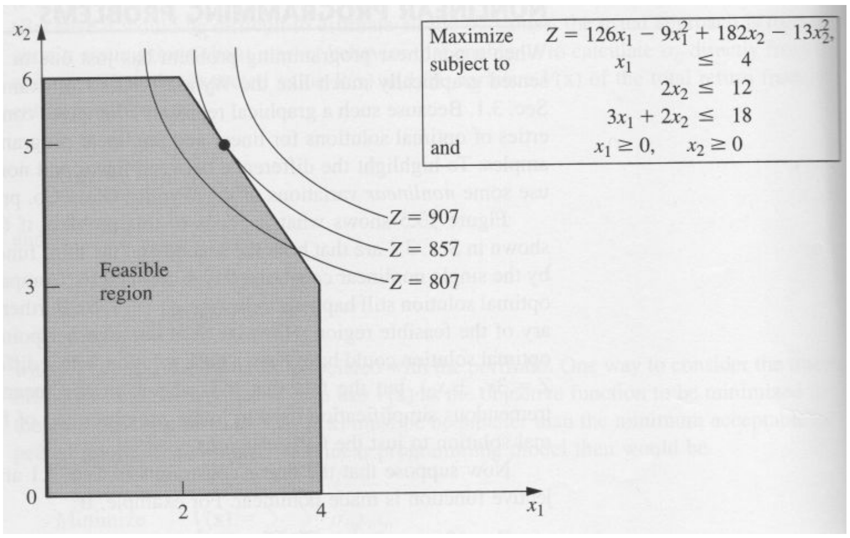

Figure 3

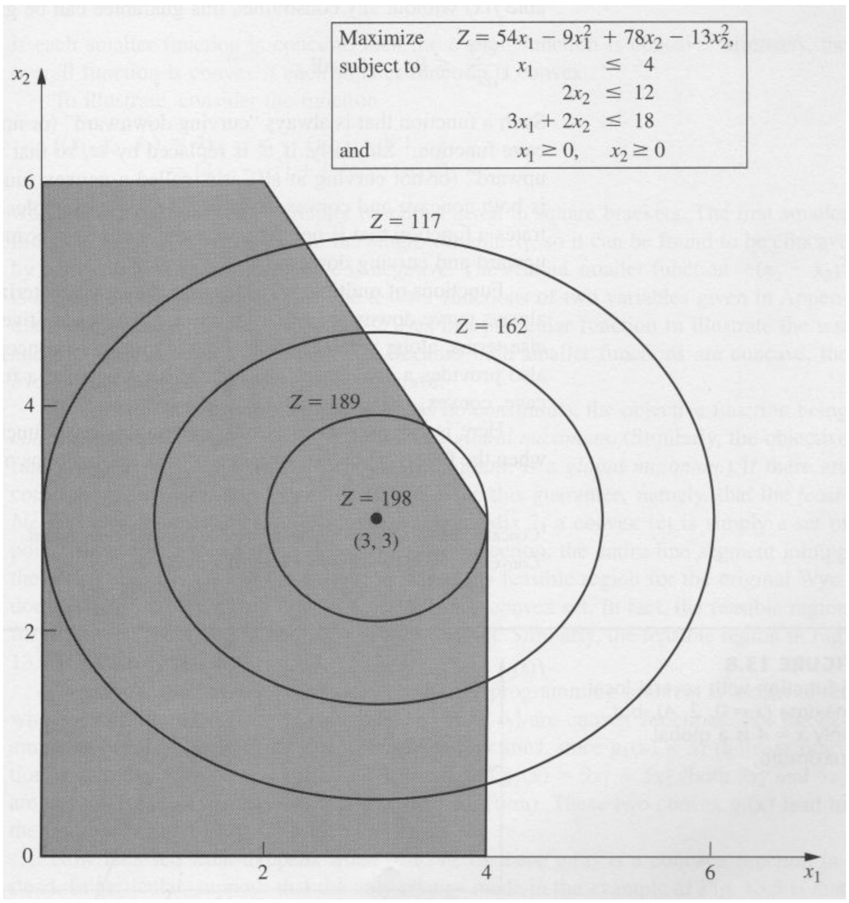

Figure 4

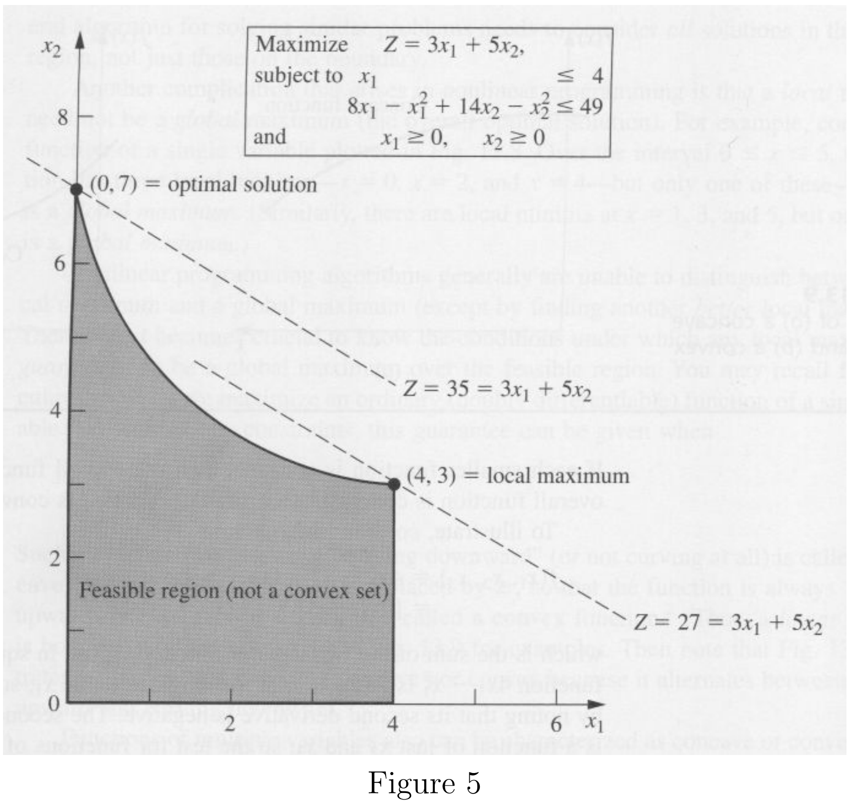

Figure 5