MA 614: LINEAR AND NONLINEAR PROGRAMMING
THE GENERAL OPTIMIZATION PROBLEM

1. Introduction
We start out by making some general remarks on optimization problems. We will see that we can say a fair amount even when our problems have little structure. Additionally, in Appendix A we summarize some useful results from real analysis and convexity.

2. The Generic Problem
The generic mathematical optimization problem, \( (P) \), is:

\[
(P) \quad \text{Minimize } f(x) \text{ subject to } x \in D
\]

The real valued function \( f \) is called the objective function, and \( D \) the constraint set, or decision set. We choose to minimize “cost.” The problem \( (P) \) is called feasible if \( D \) is non-empty, and a point in \( D \) is called a feasible solution. The decision \( x^* \in D \) solves \( (P) \) or is optimal if \( x \in D \) implies that \( f(x) \geq f(x^*) \). In this case, \( x^* \) is an optimal solution or optimizer of \( (P) \), and \( f(x^*) \) is the optimal value.

A simple modeling trick allows us to maximize “profit” instead of minimizing “cost” because:

\[
\text{Maximum } f(x) \text{ subject to } x \in D = -\text{Minimum } -f(x) \text{ subject to } x \in D
\]

This is the first of many tricks we will introduce that make the techniques we discuss much more powerful than they at first seem. These tricks allow us to convert wide varieties of optimization problems into standard forms that have effective algorithms. These tricks, while usually very simple, can be quite powerful. We now formalize one class of such tricks, and give some non-trivial applications of them.
2.1 The Monotone Transformation Lemma

Lemma: Suppose we modify the objective of (P):

(P') Minimize $f'(x)$ subject to $x \in D$,

If it is always the case that $(f'(x) < f'(y) \text{ if and only if } f(x) < f(y)$ whenever $x \in D$, and $y \in D)$, then $x^* \in D$ is an optimizer of (P) if and only if it is an optimizer of (P').

Proof: Obvious

Applications:
- $f'(x) = kf(x)$, for any positive constant $k$.
- $f'(x) = f(x) + c$, for any constant $c$; for this reason you rarely see constant terms in objective functions. (Of course the optimal value of the objective depends on the constant!)
- $f'(x) = g(f(x))$, where $g$ is strictly monotone increasing
- $f'(x) = f(x)^2$, when $f(x) \geq 0$ for $x$ of interest.
- $f'(x) = f(x)^{1/2}$, when $f(x) > 0$ for $x$ of interest.
- $f'(x) = \log f(x)$, when $f(x) > 0$ for $x$ of interest.
- $f'(x) = e^{f(x)}$

Example 1 (Boris Aronov, Lisa Hellerstein):

In some recent research, the following nasty expression arose. We can use the monotone transformation lemma to get the expression into more manageable form. It is important to note that the $x_j$'s here are all $> 0$. 


MA614/CS919 Notes Revised 9/7/05 R. Van Slyke© p. 2
\[ f(x_1, x_2, \ldots, x_n) = \frac{\sum x_i^2 + \sum_{i \neq j} x_i x_j}{\sum x_i^2} \]

\[ = \frac{1}{2} \sum x_i^2 + \frac{1}{2} \sum_{i \neq j} x_i x_j + \frac{1}{2} \sum x_i^2 \]

which has the same optimizers as:

\[ f'(x_1, x_2, \ldots, x_n) = \frac{1}{2} \sum x_i^2 + \sum_{i \neq j} x_i x_j \]

\[ = \frac{\sum \left( \frac{x_i}{\sqrt{2}} \right)^2 + 2 \sum_{i \neq j} \frac{x_i x_j}{\sqrt{2} \sqrt{2}}}{\sum x_i^2} \]

\[ = \frac{\left( \sum \frac{x_i}{\sqrt{2}} \right)^2}{\sum x_i^2} = \frac{1}{2} \left( \frac{\sum x_i}{\sqrt{2}} \right)^2 \]

which has the same optimizers as:

\[ f''(x_1, x_2, \ldots, x_n) = \left( \sum x_i \right)^2 \frac{1}{\sum x_i^2} \]

which has the same optimizers as:

\[ f'''(x_1, x_2, \ldots, x_n) = \frac{\sum x_i}{\sqrt{\sum x_i^2}} \]

Why, one might reasonably ask, should one go through all this? We have converted the original, obscure, expression to a ratio of the \(l_1\) norm of the vector \(x\) to the \(l_2\) or euclidean norm of \(x\). This, last, expression is much more recognizable and allows us to make use of the theory of such norms. The euclidean norm is the traditional measure of the length of a vector, but it cannot be directly used in linear
programs. We can, however, often use other norms, particularly the $l_1$ and the $l_\infty$ norms. We will say more about the use of these norms later.


Duffin et al, made extensive use of one of these tricks in their development of geometric programming. Suppose $f(x)$ is of the rather forbidding form: $f(x) = \prod g_i(x)$, then we can take the log, obtaining an $f'$ like $f'(x) = \log \prod g_i(x) = \sum \log g_i(x)$. (This, of course assumes that the the $g_i(x)$ are positive for feasible $x$.) The set of optimizers is then unchanged. This works particularly well when $g_i(x)$ is of the form $x^a$ or $e^{h(x)}$, for some simple $h(x)$.

Now we consider another general result for (P).

2.2 The Relaxation Lemma

Lemma: Let $f': D' \to R$, where $D' \supseteq D$ and $f'(x) \leq f(x)$ for $x \in D$. This is called a relaxation of $f$, $D$ (Figure 1). Then any optimal solution value for $f'(x)$ on $D'$ is a lower bound for any solution of (P). Moreover suppose that $x^* \in D'$ and that $f'(x^*) \leq f'(x)$ for any $x \in D'$ (that is, $x^*$ solves (P) with $f'$, $D'$ replacing $f$, $D$). Then if $x^* \in D$, and $f(x^*) = f(x^*)$ then $x^*$ solves the original problem (P).
Proof: By hypothesis, \( x^* \in D \), and since \( f(x^*) = f'(x^*) \leq f'(x) \leq f(x) \) for any \( x \in D' \), then \( f(x^*) \leq f(x) \) must also hold on \( D \) since \( D \subseteq D' \).

Examples: The simplest application of the relaxation lemma is to ignore constraints. Suppose we wish to solve (P) with \( f(x) = 2x^2 - 3x + 5 \), and \( D = \{x | 0 \leq x \leq 1\} \), where \( x \) is a real variable. Note, that to find an optimal \( x \) we can ignore the constant 5 in the objective using the monotone transformation lemma. The relaxation we consider is replacing \( D \) with \( D' \) where \( D' \) is the set of all real numbers. That is, we essentially do away with the constraints. In your earliest calculus courses you should have learned how to solve the relaxed problem (you probably did not learn how to deal with inequality constraints). To solve the relaxed problem you start by setting the derivative of \( f(x) \), which is \( 4x - 3 \), to 0, and solving for \( x^* = \frac{3}{4} \). If you were taught well, you also know to look at the sign of the second derivative to make sure it is positive—or least non-negative. It is! Since \( \frac{3}{4} \) is in \( D \) the conditions of the lemma apply and we know that \( x^* = \frac{3}{4} \) is an optimal solution of the original problem. We did not even have to deal with the inconvenient constraints. This idea is very simple, but has a huge number of applications.
Two important, non-trivial, applications of the relaxation lemma, are to integer and convex programming.

**Integer Programming:**

One form of linear programs is:

\[
\begin{align*}
\text{Min } f(x) &= cx \\
\text{Subject to } D' &= \{x | Ax = b, x \geq 0\}
\end{align*}
\]

where \(x\) is an \(n\)-vector of real numbers, \(c\) is an \(n\)-vector of given parameters, \(b\) is an \(m\)-vector of given parameters, and \(A\) is a given \(mxn\) matrix.

An integer linear program is given by:

\[
\begin{align*}
\text{Min } f(x) &= cx \\
\text{Subject to } D &= \{x | Ax = b, x \geq 0, x \text{ integer}\}
\end{align*}
\]

Then the linear program can be interpreted as a relaxation of the integer program, with \(f \equiv f'\) and \(D\) and \(D'\) as given. Then, if we solve the relaxation (the linear program) and the result happens to be a vector of integers, then the solution of the linear program also solves the integer program.

This is valuable because, in the language of computer science, algorithms for linear programming exist with polynomial time worst case performance, while integer programming is NP hard, which strongly suggests that worst case time is exponential in the size of the data.

Moreover, for important special cases, of integer programming, the usual methods of solving the linear programming relaxation
automatically yields integer solutions. For example when the matrix $A$ is *totally unimodular* this happens. This class includes transportation, assignment, and maximum flow problems, all of which are of practical interest. For a complete discussion of this, see Chapters 19-21 of [Schrijver, 1986].

Even if one is not guaranteed an integer solution of the relaxation, most integer programming solution methods start with solving the linear programming relaxation.

Another particularly important application of the relaxation lemma is to convex non-linear programming. We need some preliminaries before we can discuss this application; we will turn to it in Section 2.5.

### 2.3 Saddle Points


Often associated with optimization problems (P), are so-called **dual problems**. The primal and the dual are frequently related through a real valued *Lagrangian function*, $L$. Again, we can derive some very useful results even with little structure. So we let $L: \mathbb{R}^{m \times n} \to \mathbb{R}$, $X \subseteq \mathbb{R}^n$, $Y \subseteq \mathbb{R}^m$ with $X$ and $Y$ each closed and non-empty. Then we can define a saddle point problem, (SPP).

Find $(x,y) \in X \times Y$ so that:

\begin{equation}
(SPP) \quad L(x,u) \leq L(x,y) \leq L(v,y) \text{ for all } (u,v) \in X \times Y
\end{equation}

Associated with every (SPP) is a dual pair of optimization problems:

(PP) **Minimize** $\phi(x)$ subject to $x \in X$, and
(DP) Maximize $\psi(y)$ subject to $y \in Y$

Where $\phi(x) = \sup \{L(x,v): v \in Y\}$, and $\psi(y) = \inf \{L(u,y): u \in X\}$. $\phi(x) = \infty$, and $\psi(y) = -\infty$ are possible (extended value functions).

Then we have the:

Weak Saddle Point Theorem: For any $L: X \times Y \subseteq \mathbb{R}^{n+m} \rightarrow \mathbb{R}$, we have:

\[
\inf_{x \in X} \sup_{y \in Y} L(x,y) \geq \sup_{y \in Y} \inf_{x \in X} L(x,y)
\]

Moreover, for $(x^0, y^0) \in X \times Y$ the following are equivalent:

(a) $(x^0, y^0)$ is a saddle point of $L$ on $X \times Y$, that is, $L(x^0, y) \leq L(x^0, y^0) \leq L(x, y^0)$.

(b) $x^0$ is a minimizer of $\phi(x)$ subject to $x \in X$, $y^0$ is a maximizer of $\psi(y)$ subject to $y \in Y$, and equality holds in (1).

(c) $\phi(x^0) = \psi(y^0) = L(x^0, y^0)$.

Proof: First we have for any $(x, y) \in X \times Y$, by the definitions, that: $\phi(x) \geq L(x, y) \geq \psi(y)$. For fixed $x$, $\sup_y \psi(y) \leq \phi(x)$. Since this holds for any $x$, we then have: $\inf_x \phi(x) \geq \sup_y \psi(y)$, but this is just another way of writing (1).

(a) $\Rightarrow$ (b):
Suppose $(x^0, y^0) \in X \times Y$ is a saddle point

From this we get:
\[
\begin{align*}
\inf_x \sup_y L(x, y) &\leq \sup_{x \in X} \inf_{y \in Y} L(x^0, y) \equiv \phi(x^0) \\
&\leq L(x^0, y^0) \text{ by the left hand side} \\
&\leq \inf_{x \in X} \sup_{y \in Y} L(x, y^0) \equiv \psi(y^0) \text{ by the right hand side} \\
&\leq \sup_{y \in Y} \inf_{x \in X} L(x, y)
\end{align*}
\]

Since \( \inf_x \sup_y L(x, y) \geq \sup_{x \in X} \inf_{y \in Y} L(x, y) \), equality holds throughout yielding (b).

(b) \implies (c):

If we have (b), then:

\[
\phi(x^0) = \inf_x \phi(x) \equiv \inf_x \sup_y L(x, y) \\
= \sup_{y \in Y} \inf_{x \in X} L(x, y) \equiv \sup_{y \in Y} \psi(y) = \phi(x^0)
\]

also from the definitions \( \phi(x^0) \geq L(x^0, y^0) \geq \psi(y^0) \) this yields \( \phi(x^0) = L(x^0, y^0) = \psi(y^0) \).

(c) \implies (a):

\[
L(x^0, y^0) = \phi(x^0) \equiv \sup_{y \in Y} L(x^0, y), \text{ similarly}
\]

\[
L(x^0, y^0) = \psi(y^0) \equiv \inf_{x \in X} L(x, y^0)
\]

So the saddle point relation holds (with equality throughout).

\[2.4 \text{ Existence of an Optimal Solution}\]

See Theorem A.2 in Appendix A for:

**Theorem:** If \( D \) is closed and bounded (compact), and non-empty, and \( f \) is continuous, then (P) attains an optimal solution.
2.5 Convex/Concave Programming

To say much more we have to introduce more structure to (P).

We now restrict ourselves to the special case where the objective function and constraint set are convex.

See Appendix A for definitions and some properties of these structures.

If $f$ is a differentiable convex function in some convex domain $D$, with non-empty interior in $\mathbb{R}^n$, then for any $x^0$ we have $f(x) \geq L(x, x^0) = f(x^0) + \nabla f(x^0)(x - x^0)$ where $\nabla f(x^0)$ is the gradient of $f$ at $x^0$. (See Theorem A.11.) Thus $L(x, x^0)$ is a relaxation of $f$. Even if $f$ is not differentiable we can always use the subgradient in the same way. (See Theorem A.10.) So if we are solving (P) where $f$ is convex and $D$ is defined by linear constraints, we can try to replace $f$ by its linearization $L(x, x^0)$. If we solve the relaxation obtaining $x^*$, and $L(x^*, x^0) = f(x^*)$, then we have solved the original problem. Note, in this case, $x^* = x^0$. In fact, under general assumptions we have that $L(x^*, x^*) = f(x^*)$ at the optimum, $x^*$. This leads to a special case of the Karush-Kuhn-Tucker conditions, which we will discuss later in the course.

![Figure 2: Convex Relaxation](image-url)
Definition: Let $D \subseteq \mathbb{R}^n$ in (P). Then $x^*$ in $D$ is said to be a (global) minimum of $f$ on $D$ if $x \in D$ implies $f(x) \geq f(x^*)$. $x^*$ is a local minimum if there exists $\varepsilon > 0$ so that if $x \in D$ and $\|x - x^*\| < \varepsilon$ then $f(x) \geq f(x^*)$.

Theorem: Suppose $f$ and $D$ in (P) are convex, suppose further that $D$ is closed, then any local optimum is a global optimum. On the other hand, if $f$ is concave and $D$ contains no lines, then, if the infimum $\{f(x)|x \in D\}$ is achieved by a point in $D$, then it is achieved at an extreme point (Figure 3).

Proof: Suppose $f$ is convex, and let $x^*$ be a local minimum point. Suppose there is an $x^0$ in $D$ so that $f(x^0) < f(x^*)$. The segment connecting $x^*$ and $x^0$ must be contained in $D$. Then $f(x) \leq (1-\lambda)f(x^*) + \lambda f(x^0) = f(x^*) + \lambda(f(x^0)-f(x^*)) < f(x^*)$ for $0 \leq \lambda < 1$. Therefore, for any positive $\lambda$ no matter how small, $f(x)$ will be strictly less than $f(x^*)$, and thus $x^*$ does not provide a local minimum—contradiction.

Now, suppose $f$ is concave. Let $x^*$ be a point at which an infimum is attained. If $D = \{x^*\}$ is a singleton then $x^*$ is an extreme point and we're done. If $x^*$ is an interior point of $D$ then $f$ is constant on $D$ (Lemma A.9). We can assume then, without loss of generality, that $x^*$ is a boundary point. To see this, let $x^0$ be any other point in $D$, and consider the line through $x^*$ and $x^0$. Since $D$ contains no lines, our line cannot all be in $D$. In at least one direction the line leaves $D$ at a
boundary point (by closure). Therefore we can assume that \( x^* \) is a boundary point of \( D \). Let \( H \) be a supporting hyperplane for \( D \) at \( x^* \). Consider \( D' = D \cap H \). \( D' \) satisfies the assumptions of the theorem for \( D \), that is, \( D' \) is closed convex, and contains no lines. However, the dimension of \( D' \) is, at least, one less than the dimension of \( D \). By induction we can repeat the process until we reach a singleton since every extreme point of \( D' \) is an extreme point of \( D \) (Lemma A.8).

**Corollary:** If a non-empty convex set contains no lines, it has an extreme point.

**Proof:** Let \( f \) be a constant in the theorem.

**Corollary:** If \( f \) in \( (P) \) is linear, any local optimum is a global optimum and if there is a global (local optimum) there is one at an extreme point.

The corollary has strong practical implications for linear programming. It says we need only look at extreme points. Moreover, at an extreme point all we have to do is check local optimality. It will turn out that the constraint set for linear programming has a finite number of extreme points, so this implies that solving linear programs might be relatively easy. Encouraged by this, we will start with a detailed treatment of this special case of \( (P) \).

3. **Summary**

We have been able to say quite a lot about \( (P) \), even with very little structure to it.

We:

(i) Introduced a modeling trick that allows you to modify, and perhaps simplify, the objective function without changing the optimizers.
(ii) Characterized problem relaxations that provide a bound on the objective function over all feasible solutions, and giving conditions that guarantee that the relaxation actually solves the original problem.

(iii) Provided conditions guaranteeing the existence of a solution to (P).

(iv) Introduced the notions of convexity and concavity, which, respectively guarantee that local minima are global minima, and that local minima occur at extreme points. Problems with linear objectives are both convex and concave so that we can look restrict our search to extreme points until we find the first local minimum, which is necessarily a global minimum. If in addition the constraints are linear, the number of extreme points is finite.

(v) Introduced the notion of dual optimization programs that are connected by a lagrangian function, and proved a general theorem that provides bounds on the objectives of the two problems.

There are many important special cases of (P), several of which we will study in this course.

**Special Cases:**

- Unconstrained optimization
- Equality constrained optimization
- Linear programming
- Combinatorial (0-1) programming
- Integer programming
- Quadratic (Objectives) Programming
- Convex programming
- Non-linear programming
- Stochastic programming
- Functional optimization (dynamic programming, optimal control theory, calculus of variations) The development of this last category actually predated much of the finite dimensional work.
REFERENCE: