In linear programming problems, values for a set of \( n \) design variables, \((x_1, x_2, \cdots, x_n)\), are to be found that maximizes a weighted sum of the design variables, such that a set of \( m \) inequality constraints, are satisfied. Linear programming problems are generally expressed as

\[
\begin{align*}
\max_{x_1, x_2, \cdots, x_n} & \quad f = c_1x_1 + c_2x_2 + \cdots + c_n x_n \\
\text{such that} & \quad a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n \leq b_1 \\
& \quad a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n \leq b_2 \\
& \quad \vdots \\
& \quad a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n \leq b_m
\end{align*}
\] (1)

If the objective function is linear in the design variables and the constraint equations are linear in the design variables, the linear programming problem usually has a unique solution.

**Linear Programming in One Dimension**

Consider the simplest of all linear programming problems, a problem in one dimension:

\[
\max_x \quad cx \\
\text{such that} \quad ax \leq b
\] (2)

where \( a, b, \) and \( c \) are all positive. This problem has a single design variable, the objective function is linear \((f = cx)\), there is a single inequality constraint, which is also linear in \( x \). For values of \( x \) for which \( ax > b \), the solution is infeasible. In this problem, it is easy to see that the solution must be \( x^* = b/a \). Not all linear programming problems are so easy; most linear programming problems require more advanced solution methods. The methods of Lagrange multipliers is one such method.

Lagrange multiplier methods involve the augmentation of the objective function through the addition of terms that describe the constraints. The objective function is augmented by the constraint equations through a set of non-negative multiplicative Lagrange multipliers, \( \lambda_j \geq 0 \). The augmented objective function, \( f_A(x) \), is a function of the \( n \) design variables and \( m \) Lagrange multipliers,

\[
f_A(x_1, x_2, \cdots, x_n, \lambda_1, \lambda_2, \cdots, \lambda_m) = c_1x_1 + c_2x_2 + \cdots + c_n x_n \\
+ \lambda_1 (a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n - b_1) \\
+ \lambda_2 (a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n - b_2) \\
\vdots \\
+ \lambda_m (a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n - b_m).
\]

\[
f_A(x, \lambda) = c^T x + \lambda^T (Ax - b)
\] (3)
For the one-dimensional problem of equation (2), \( n = 1 \) and \( m = 1 \), so

\[
f_A(x, \lambda) = cx + \lambda (ax - b) = cx + ax\lambda - b\lambda
\]

(4)

The Lagrange multiplier, \( \lambda \), serves the purpose of modifying (augmenting) the objective function from a linear equation \( f(x) = cx \) to a bilinear equation \( f_A(x, \lambda) = cx + ax\lambda - b\lambda \).

Figure 1(a) plots \( f_A(x, \lambda) \) for a few non-negative values of \( \lambda \) and Figure 1(b) plots contours of \( f_A(x, \lambda) \).

Figure 1. \( f_A(x, \lambda) = cx - ax\lambda + b\lambda \) for \( a = 1 \), \( b = 2 \), and \( c = 3 \). \((x^*, \lambda^*) = (b/a, c/a) = (2, 3)\)
These figures show that:

- \( f_A(x, \lambda) \) is independent of \( \lambda \) at \( x = b/a \),
- \( f_A(x, \lambda) \) is independent of \( x \) at \( y = -c/a \),
- the surface \( f_A(x, \lambda) \) is a saddle shape,
- the point \((x^*, y^*) = (b/a, -c/a)\) is a saddle point.

At a saddle point the slope in all directions is zero.

\[
\frac{\partial f_A(x, \lambda)}{\partial x} \bigg|_{x=x^*, \ \lambda=\lambda^*} = 0 \quad (5a)
\]
\[
\frac{\partial f_A(x, \lambda)}{\partial \lambda} \bigg|_{x=x^*, \ \lambda=\lambda^*} = 0 \quad (5b)
\]

For this one-dimensional problem,

\[
\frac{\partial f_A(x, \lambda)}{\partial x} \bigg|_{x=x^*, \ \lambda=\lambda^*} = 0 \Rightarrow c + ay^* = 0 \Rightarrow y^* = -c/a \quad (6a)
\]
\[
\frac{\partial f_A(x, \lambda)}{\partial \lambda} \bigg|_{x=x^*, \ \lambda=\lambda^*} = 0 \Rightarrow ax^* - b = 0 \Rightarrow x^* = b/a \quad (6b)
\]

and the optimal objective is

\[
z^* = cx^* = cb/a \quad (7)
\]

Linear Programming in Multiple Dimensions

Linear programming problems involving \( n \) primary variables \( x_1, \ldots, x_n \), and \( m \) inequality constraints, are written

\[
\max_x f = c^T x \quad \text{such that} \quad Ax \leq b \quad (8)
\]

where \( c \) and \( x \) are \( n \)-dimensional column vectors, \( A \) is an \( m \)-by-\( n \) matrix and \( b \) is an \( m \)-dimensional column vector. Bound inequalities such as \( x_j \geq 0 \) are incorporated into the inequalities \( Ax \leq b \).

Now we introduce Lagrange multipliers (also known as dual variables) to define the augmented objective function.

\[
f_A(x, \lambda) = c^T x + \lambda^T (Ax - b). \quad (9)
\]

where

\[
\sum_{j=1}^{n} a_{ij} x_j - b_i < 0 \quad \Rightarrow \quad \lambda_i = 0 \quad (10a)
\]
\[
\sum_{j=1}^{n} a_{ij} x_j - b_i = 0 \quad \Rightarrow \quad \lambda_i > 0 \quad (10b)
\]
Condition (10a) indicates that constraint inequality $i$ is satisfied, and therefore does not bound the solution. In that case the dual variable for that inequality constraint is set to zero. Condition (10b) indicates that constraint inequality $i$ bounds the solution and the dual variable $\lambda_i$ is set to be greater than zero. Note that in either case, the product of each dual variable, $\lambda_i$ and its associated constraint equation is zero,

$$\lambda_i \left( \sum_j a_{ij}x_j - b_i \right) = 0. \quad (11)$$

for every constraint $i = 1, \cdots, m$. So under these complementary slackness conditions the augmented objective $f_A$ is equal to the true objective, $z$, as long as the constraints are satisfied.

As was illustrated in the example of linear programming in one-dimension, the solution to any linear programming problem is at a saddle point. So,

$$\frac{\partial f_A}{\partial x} = 0 : \quad c + A^T\lambda = 0 \quad (12a)$$

$$\frac{\partial f_A}{\partial \lambda} = 0 : \quad Ax - b = 0 \quad (12b)$$

Substituting equations (12a) and (12b) into the objective function,

$$f = c^T x = -\lambda^T Ax = -\lambda^T b = -b^T \lambda \quad (13)$$

So, the objective can be written in terms of primal variables, $f = c^T x$, or in terms of dual variables, $f = -b^T \lambda$.

We can now present the Primal and Dual forms of linear programming problems.

<table>
<thead>
<tr>
<th>Primal</th>
<th>Dual</th>
</tr>
</thead>
<tbody>
<tr>
<td>objective $\max_x \left( c^T x \right)$</td>
<td>maximum $\max_\lambda \left( -b^T \lambda \right)$</td>
</tr>
<tr>
<td>such that $Ax \leq b$</td>
<td>$A^T \lambda \leq c$</td>
</tr>
<tr>
<td>$x \geq 0$</td>
<td>$\lambda \leq 0$</td>
</tr>
<tr>
<td>dual variable $\lambda$</td>
<td>$x$</td>
</tr>
<tr>
<td>$f_A$</td>
<td>$c^T x + \lambda^T (Ax - b) = -b^T \lambda + x^T (A^T \lambda + c)$</td>
</tr>
</tbody>
</table>

Using the derivations above, you can show that the dual of the dual is the primal. The dual variables for the dual problem are the primal variables. The Primal and the Dual forms of linear programs are completely equivalent. The choice as to which problem to solve depends on the number of primal variables and the number of dual variables. For problems with many primal variables ($x_j$) and few constraints, it is often easier to solve the dual problem than to solve the primal problem.
The dual variables indicate how much more objective can be gained by relaxing each of the constraints that bounds the solution. Most numerical methods used to solve linear programming problems (e.g., linprog in MATLAB) make use of primal and dual variables, and provide the optimal dual variables with the solution.

To solve a Primal LP in MATLAB:

```matlab
% compute the solution to the PRIMAL LP problem using 'linprog' in matlab
[x_opt, f_opt, ~, ~, lambda_opt] = linprog(c, A, b, [], [], x_lb, x_ub);
```

To solve a Dual LP in MATLAB:

```matlab
% compute the solution to the DUAL LP problem using 'linprog' in matlab
[lambda_opt, f_opt, ~, ~, x_opt] = linprog(b, [], [], A', c, zeros(m,1));
```

In linprog the second and third arguments define inequality constraints; the fourth and fifth arguments define equality constraints; and the sixth and seventh arguments give lower and upper bound constraints on \( x \).

It is easy to numerically check that the solution to the primal problem equals the solution to the dual problem.