Simplex Algorithm\(^1\)

Extended rules for deriving a dual problem

Below, we list more general rules for converting a primal problem into a dual problem. Assume that in the primal problem: the set of inequalities is indexed by \(\mathcal{I}\), the set of equalities is indexed by \(\mathcal{E}\), and \(\mathcal{I} \cup \mathcal{E} = \{1, 2, \ldots, m\} \triangleq [m]\). Similarly, the set of variables of the primal problem is indexed by \(\{1, 2, \ldots, n\} \triangleq [n]\), the set of nonnegative variables is indexed by \(\mathcal{N}\), the set of unrestricted variables is indexed by \([n]\setminus \mathcal{N}\).

<table>
<thead>
<tr>
<th>Primal LP</th>
<th>Dual LP</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\max \sum_{j=1}^{n} c_j x_j)</td>
<td>(\min \sum_{i=1}^{m} b_i y_i)</td>
</tr>
<tr>
<td>(\forall i \in \mathcal{I} : \sum_{j=1}^{n} a_{ij} x_j \leq b_i)</td>
<td>(\forall j \in \mathcal{N} : \sum_{i=1}^{m} a_{ij} y_i \geq c_j)</td>
</tr>
<tr>
<td>(\forall i \in \mathcal{E} : \sum_{j=1}^{n} a_{ij} x_j = b_i)</td>
<td>(\forall j \in [n]\setminus \mathcal{N} : \sum_{i=1}^{m} a_{ij} y_i = c_j)</td>
</tr>
<tr>
<td>(\forall j \in \mathcal{N} : x_j \geq 0)</td>
<td>(\forall i \in \mathcal{I} : y_i \geq 0)</td>
</tr>
</tbody>
</table>

**Note.** Please observe that the dual of the dual problem is the primal problem.

**Strong duality**

The following theorem is a stronger result than the weak duality. We omit the proof.

**Theorem 1** There are four possibilities:

1. Both primal and dual problems have no feasible solution.
2. The primal problem is unfeasible and the dual problem is unbounded.
3. The primal problem is unbounded and the dual problem is unfeasible.
4. Both the primal and the dual problem are feasible, and the maximum of the primal is equal to the minimum of the dual.

In Case 4, the picture of the world is as follows:

\(^1\)Additional reading: Sections 7.4 and 7.6 in the book of S. Dasgupta, C. Papadimitriou and U. Vazirani “Algorithms”.
Simplex Algorithm

George Dantzig ’1947. The simplex algorithm generally has high (exponential) complexity, but often is very efficient in practice, and therefore quite popular.

**Definition 1** Vertex is a point at which several hyperplanes intersect.

If the LP problem under consideration has \( n \) variables, then we need at least \( n \) linear equations in order to have a unique solution. The required number of equations might be higher if some equations are redundant.

**Definition 2** Two vertices are neighbors if they have \( n - 1 \) defining equalities in common.

The high-level idea of the simplex algorithm is as follows:

- Start at a feasible point.
- Move to the neighboring vertex, which has a better value of the objective function.

**Example**

Consider the following LP problem.

\[
\begin{align*}
\text{max} & \quad 2x_1 + 5x_2 \\
\text{s.t.} & \quad 2x_1 - x_2 \leq 4 \quad (1) \\
& \quad x_1 + 2x_2 \leq 9 \quad (2) \\
& \quad -x_1 + x_2 \leq 3 \quad (3) \\
& \quad x_1 \geq 0 \quad (4) \\
& \quad x_2 \geq 0 \quad (5)
\end{align*}
\]

**Step 0: initialization**

The simplex algorithm can start in the origin, in this case \((x_1, x_2) = (0, 0)\). For this selection of the feasible point, the equations (4) and (5) are tight, i.e. they are satisfied with equality (we mark tight constraints with a special sign \(\checkmark\) below.) The corresponding value of the objective function is 0.
We have then the following situation.

\[
\begin{align*}
\text{max} & \quad 2x_1 + 5x_2 \\
\text{s.t.} & \quad 2x_1 - x_2 \leq 4 \quad (1) \\
& \quad x_1 + 2x_2 \leq 9 \quad (2) \\
& \quad -x_1 + x_2 \leq 3 \quad (3) \\
& \quad x_1 \geq 0 \quad (4) \checkmark \\
& \quad x_2 \geq 0 \quad (5) \checkmark 
\end{align*}
\]

**Step 1**

We try to increase gradually one of the variables. Let us assume that we arbitrarily picked the variable \( x_2 \) to be increased. The inequality (2) prevents \( x_2 \) from being increased above 4.5, the inequality (3) prevents \( x_2 \) from being increased above 3, while inequality (1) allows for unlimited increase in \( x_2 \).

Thus, we increase \( x_2 \) from 0 to 3. Inequality (3) now becomes tight (while equation (5) is not tight anymore). For convenience, we also replace variables as follows:

- \( y_1 = x_1 \);
- \( y_2 = 3 + x_1 - x_2 \) (so now (3) becomes \( y_2 \geq 0 \)).

Equivalently, \( x_1 = y_1 \) and \( x_2 = 3 + y_1 - y_2 \).

After the substitution, the LP problem becomes:

\[
\begin{align*}
\text{max} & \quad 15 + 7y_1 - 5y_2 \\
\text{s.t.} & \quad y_1 + y_2 \leq 7 \quad (1) \\
& \quad 3y_1 - 2y_2 \leq 3 \quad (2) \\
& \quad y_2 \geq 0 \quad (3) \checkmark \\
& \quad y_1 \geq 0 \quad (4) \checkmark \\
& \quad -y_1 + y_2 \leq 3 \quad (5)
\end{align*}
\]

Note that the new variables are \( y_1 = 0 \) and \( y_2 = 0 \), and the corresponding value of the objective function is 15.

**Step 2**

We increase the value of \( y_1 \) (note that increase in \( y_2 \) will actually make the objective value smaller). The inequality (2) is the most restricting in this case, and therefore we can only increase \( y_1 \) from 0 to 1, thus making (2) tight.

The corresponding substitution is:

- \( z_1 = 3 - 3y_1 + 2y_2 \) (and so (2) becomes \( z_1 \geq 0 \));
- \( z_2 = y_2 \).
Equivalently, \( y_1 = \frac{1}{3}(3 - z_1 + 2z_2) \) and \( y_2 = z_2 \).

The corresponding LP problem is:

\[
\begin{align*}
\text{max} & \quad 22 - \frac{7}{3}z_1 - \frac{1}{3}z_2 \\
\text{s.t.} & \quad -\frac{1}{3}z_1 + \frac{5}{3}z_2 \leq 6 \quad (1) \\
& \quad z_1 \geq 0 \quad (2) \quad \checkmark \\
& \quad z_2 \geq 0 \quad (3) \quad \checkmark \\
& \quad \frac{1}{3}z_1 - \frac{2}{3}z_2 \leq 1 \quad (4) \\
& \quad \frac{1}{3}z_1 + \frac{1}{3}z_2 \leq 4 \quad (5)
\end{align*}
\]

Here, \( z_1 = 0 \), \( z_2 = 0 \), and the value of the objective function is 22.

The obtained solution is optimal. Indeed, an increase in either \( z_1 \) or \( z_2 \) will decrease the value of the objective function.

To find the values of \( x_1 \) and \( x_2 \) corresponding to the optimum value can be done in the following way. Solve equations in (2) and (3) to find the values of \( x_1 \) and \( x_2 \).

\[
\begin{align*}
\begin{cases}
   x_1 + 2x_2 = 9 \\
   -x_1 + x_2 = 3
\end{cases}
\end{align*}
\]

and the corresponding solution is \( x_1 = 1, x_2 = 4 \).

The corresponding run of the simplex algorithm can schematically be illustrated by the following figure.

---

**Finding the starting vertex**

Generally, the origin is not always a feasible point of a given LP problem. Below, we present a technique for finding an initial feasible point to be used with the simplex algorithm.
Consider the following LP problem.

\[
\begin{align*}
\text{max} & \quad c^T \cdot x \\
\text{s.t.} & \quad Ax = b \\
& \quad x \geq 0 .
\end{align*}
\]

A feasible point can be found by using the following steps.

1. Make sure that that the right-hand side vector \( b \) is non-negative (otherwise, change the sign of the equation).

2. Create a new LP problem:
   - Introduce \( m \) new additional variables \( z_1 \geq 0, z_2 \geq 0, \ldots, z_m \geq 0 \);
   - Add \( z_i \) to the left-hand side of the \( i \)-th equation;
   - The new objective function to be minimized is:
     \[
     z_1 + z_2 + \cdots + z_m .
     \]

3. For this new LP problem, the starting vertex can be chosen as \( z_i = b_i \) for \( i = 1, 2, \ldots, m \), and zero for all other variables.

4. Solve this new LP problem.

There are two possible cases:

(1) If the optimum value of the objective obtained when \( z_1 = z_2 = \cdots = z_m = 0 \), then set all \( z_i = 0 \), and hence the optimal solution of the new problem is a feasible solution of the original problem.

(2) If the objective function in the optimum solution for the new problem is positive, then \( z_1, z_2, \ldots, z_m \) cannot be zeros all at the same time, and therefore there is no feasible solution to the original problem.

Special cases

Unboundness

In some cases the LP problem is unbounded. Thus, the objective function can be made arbitrarily large (or arbitrarily small for a minimization problem). The simplex algorithm will recognize these cases, and to give the corresponding message.
Degeneracy

Consider an LP problem with three variables, where some vertex is an intersection of four planes.

Any three of these inequalities can be chosen to describe the vertex. Simplex algorithm can mistakenly detect a suboptimal degenerate vertex because all of its neighbors have no better objective value.

This problem can be solved, for example, by a small perturbation: \( b_i = b_i \pm \epsilon_i \), where \( \epsilon_i \geq 0 \) are small.
**Time complexity**

Assume that the input problem has \( n \) variables, \( m \) constraints. One step (iteration) of the simplex algorithm requires \( O(mn) \) operations (increasing one variable, and substitution of \( n \) new variables into \( m \) inequalities).

The number of one step is bounded from above by the number of vertices in the polytope, which is \( O \left( \binom{m}{n} \right) \) – exponentially large in \( n \).

Therefore, the worse-case running time of the simplex algorithm is exponential in the size of the input, while in practice usually simplex algorithm works rather fast.