Linear Programs

Initializing The Simplex Algorithm

Bashar Dudin

January 3, 2017

EPITA

Where We Stand, What We Face?

We now have an algorithm SIMPLEX (*Restricted*), which is conjectured to solve linear programs. For the time being we've shown that it terminates either by specifying that we have an unbounded linear program, or by returning a finite value *expected* to be the maximum we are looking for.

Uptill now, we've been working under the following two assumptions :

- Our linear program is ***feasible*** ; meaning that it has at least one feasible solution
- The basic solution of the initial slack form is feasible.

During this lecture, we are going to build a function INITSIMPLEX taking in a linear program $(\boldsymbol{A}, \boldsymbol{b}, \boldsymbol{c})$ and either returning back the fact it is ***not*** feasible or a linear program $(N, B, \underline{\boldsymbol{A}}, \underline{\boldsymbol{b}}, \underline{\boldsymbol{c}}, v)$ in slack form and that has feasible basic solution.

Feasibility

Deciding On Feasibility

Let L be the linear program, given in standard form :

$$\begin{aligned} &\text{maximize} && z = v + \sum_{j=1}^{n} c_j x_j \\ &\text{subject to} && \\ & && \forall i \in B, \quad \sum_{j=1}^{n} a_{ij} x_j \leq b_i \\ &\text{with} && \forall j \in N, \quad x_j \geq 0 \end{aligned}$$

Out of L we build the following auxiliary linear program L_{marker} :

$$\begin{aligned} &\text{maximize} && z = -x_0 \\ &\text{subject to} && \\ & && \forall i \in B, \quad \sum_{j=1}^{n} a_{ij} x_j - x_0 \leq b_i \\ &\text{with} && \forall j \in N \cup \{0\}, \quad x_j \geq 0 \end{aligned}$$

Deciding On Feasibility

Proposition

L is feasible if, and only if, the optimal objective value of L_{marker} is 0.

Deciding On Feasibility

Proposition

L is feasible if, and only if, the optimal objective value of L_{marker} is 0.

Proof : Notice first that the optimal objective value of L_{marker} is 0. Therefore, if we show that 0 is an objective value of L_{marker} it is necessarily the optimal one.
If L is feasible then there is a tuple $(t_1, \ldots, t_n)$ of non-negative real numbers satisfying all linear constraints of L. The tuple $(0, t_1, \ldots, t_n)$ does thus satisfy L_{marker} and 0 is then a objective value of L_{marker}, i.e. the optimal one.
Conversely, if L_{marker} has objective value 0 (thus optimal) it is of the form $(0, t_1, \ldots, t_n)$. Plugging this tuple in the linear constraints of L_{marker} it implies $(t_1, \ldots, t_n)$ is a solution of L. ∎

Deciding On Feasibility

Proposition

L is feasible if, and only if, the optimal objective value of L_{marker} is 0.

The difference between L and L_{marker} is that L_{marker} is always feasible :

If E is the set of indices of B such that b_i is negative then the tuple $(-\min_i\{b_i\}, 0, \ldots, 0)$ is a feasible solution of L_{marker}.

Thus, temporarily admitting validity of the *restricted* simplex algorithm, if we're able to find a linear program equivalent to L_{marker} which has feasible basic solution then we can decide on feasibilty of L.

Deciding On Feasibility

Consider the following slack form of L_{marker}

$$
\begin{array}{lrl}
\text{maximize} & & z = -x_0 \\
\text{subject to} & & \\
& \forall i \in B, & x_i = b_i - \sum_{j=1}^{n} a_{ij} x_j + x_0 \\
\text{with} & \forall j \in N \cup B \cup \{0\}, & x_j \geq 0
\end{array}
$$

Basic solution of L_{marker} is not feasible as soon as L is not, i.e. as soon as one of the b_i is negative. **We assume this is the case**.

Let k be the index of a minimum b_k among all b_i coefficients. We already know that $(-b_k, 0, \ldots, 0)$ is a feasible solution of L_{marker}.

Let's now use PIVOT with entering variable 0 and leaving one k. The **same** previous feasible solution of L_{marker} is now the basic solution of the linear program we got after pivotting.

We got an equivalent linear program to L_{marker} having feasible basic solution !

Testing Feasibility

Input: $L = (\boldsymbol{A}, \boldsymbol{b}, \boldsymbol{c})$ a linear program in standard form having infeasible initial basic solution

Output: An auxiliary linear program having feasible basic solution if L is feasible, raises an exception *infeasible* otherwise.

function IsFeasible($\boldsymbol{A}, \boldsymbol{b}, \boldsymbol{c}$)
    form L_{marker} by adding a column of -1s at the beginning of $\boldsymbol{A}$ and change $\boldsymbol{c}$ to $(-1, 0, \ldots, 0)$.
    ▷ corresponds to adding $-x_0$ on the left hand of each linear constraint and to putting objective function to $-x_0$
    let $(N, B, \boldsymbol{A}, \boldsymbol{b}, \boldsymbol{c}, 0)$ be the resulting slack form of L_{marker}
    $k \leftarrow$ index of $\min_i\{b_i\}$
    $(N, B, \boldsymbol{A}, \boldsymbol{b}, \boldsymbol{c}, 0, k, 0) =$ Pivot$(N, B, \boldsymbol{A}, \boldsymbol{b}, \boldsymbol{c}, 0, k, 0)$
    ▷ The basic solution for L_{marker} is now feasible
    if Simplex$(N, B, \boldsymbol{A}, \boldsymbol{b}, \boldsymbol{c}, 0) = 0$ **then**
        return $(N, B, \boldsymbol{A}, \boldsymbol{b}, \boldsymbol{c}, 0)$
    else
        raise exception *infeasible*
    end if
end function

Get A Feasible Basic Solution

Getting a Feasible Basic Solution

Assumption

We assume IsFeasible(L) didn't raise any exception.

Getting a Feasible Basic Solution

Assumption

We assume IsFeasible(L) didn't raise any exception.

Under the previous assumption IsFeasible returns an equivalent linear program L' to L_{marker} which has feasible basic solution with objective value 0.

Getting a Feasible Basic Solution

Assumption

We assume IsFeasible(L) didn't raise any exception.

Let's replace the objective value of L' with the one of L we started with. To get a linear program L'' out of this we also need to replace the possible basic variables of the objective value with their expressions in terms of non-basic variables of L'. Let's also make sure that x_0 is not a basic variable of L'' by pivoting L'' with leaving variable x_0 and any other entering variable having non-zero coefficient in the corresponding line.

Fact

Putting x_0 to 0 in L'' gives back a linear program whose equivalent to L and has feasible basic solution.

The InitSimplex Function

Input: $L = (\boldsymbol{A}, \boldsymbol{b}, \boldsymbol{c})$ a linear program in standard form

Output: An equivalent program having feasible basic solution if L is feasible.

function InitSimplex($\boldsymbol{A}, \boldsymbol{b}, \boldsymbol{c}$)
    if IsFeasible($\boldsymbol{A}, \boldsymbol{b}, \boldsymbol{c}$) raises an exception **then**
        raise exception *infeasible*
    else
        $(N, B, \boldsymbol{A}', \boldsymbol{b}', \boldsymbol{c}', v) =$ IsFeasible($\boldsymbol{A}, \boldsymbol{b}, \boldsymbol{c}$)
        if 0 is basic **then**
            Choose e such that $a'_{0e} \neq 0$
            $(N, B, \boldsymbol{A}', \boldsymbol{b}', \boldsymbol{c}', v) =$ Pivot($N, B, \boldsymbol{A}', \boldsymbol{b}', \boldsymbol{c}', v, 0, e$)
        end if
        Update $(\boldsymbol{c}, v)$ to express objective value in terms non-basic indices
        Remove column corresponding to index 0 from $\boldsymbol{A}'$
        return $(N, B, \boldsymbol{A}', \boldsymbol{b}', \boldsymbol{c}, v)$
    end if
end function

Where We Stand?

We are now able to

- know whether a linear program is feasible or not using IsFeasible
- if it is feasible we can build up an equivalent linear program which has feasible basic solution using InitSimplex
- once we get a linear program having feasible basic solution we can check whether our linear program is unbounded or has a hopefully optimal finite solution using the *restricted* Simplex.

We are left with showing validity of *restricted* Simplex in order to show we indeed got what we hope for in the previous last point.

We call Simplex the algorithm globally going through all previous steps.

That's it for today !