

Simplex Solver



Manual de Usuario

David Jiménez Xolalpa A00994415

Alejandro Méndez González A01210989

Introducción:

El método simplex sirve para obtener soluciones numéricas a un problema de programación

lineal. Da una solución óptima a un problema de maximización o minimización. El presente programa aplica el método simplex a problemas de maximización. No considera ni valida casos especiales del simplex.

Algoritmo en Pseudocódigo:

1. Condición de termino: Se revisa que todos los coeficientes del renglón Z no sean negativos, cuando esto se cumpla se habrá terminado
2. Se escoge el coeficiente más negativo del renglón Z y se fija la posición del pivote vertical
3. Se dividen los valores de la columna LD entre los de la columna pivote, y se elige la posición del menor valor no negativo resultante para fijar el pivote horizontalmente
4. Se hace 1 la interseccion entre la columna y fila pivotes, dividiendo esa fila entre el valor pivote
5. Con operaciones básicas de renglón, se hace 0 el resto de los valores de esa columna
6. Entregar los valores de la respuesta final

Código Fuente:

```
/**
 * Object that wraps simplex operations
 */
function Simplex() {}

// Globals
/** Stores the current simplex objects */
Simplex.currentSimplex = null;

// Variables -----
/** Variables array */
Simplex.prototype.variables = null;
/** Backup of variables array */
Simplex.prototype.backup = null;
Simplex.prototype.numVariables = 0;
Simplex.prototype.numRestricciones = 0;
/** Answer array */
Simplex.prototype.answer = null;

// Methods -----

/** Gets data from HTML form */
Simplex.prototype.getData = function() {
    var si = Simplex.currentSimplex;
    console.log("GettingData-----");
    var arr = new Array(si.numRestricciones + 1);

    // Build objective function
```

```

var obj = Simplex.getRowData("objetivo");
arr[0] = obj;

// Build restrictions
for (var i = 0; i < si.numRestricciones; i++) {
    obj = Simplex.getRowData("restriccion" + i);
    arr[i+1] = obj;
}

// Build title
var title = new Array(2 + si.numVariables + si.numRestricciones + 1);
title[0] = "VB";
title[1] = "Z";
for (var i = 0; i < si.numVariables; i++) {
    title[i+2] = "X" + (i+1);
}
for (var i = 0; i < si.numRestricciones; i++) {
    title[i+2+si.numVariables] = "S" + (i+1);
}
title[title.length -1] = "LD";

arr.unshift(title);
var copy = new Array();
for (var i = 0; i < arr.length; i++) {
    copy.push(arr[i].slice(0));
}
si.variables = arr;
si.backup = copy;
console.log(arr);
}

/** Apply simplex method */
Simplex.prototype.solve = function() {
    return principal();
}

/** Display simplex tables */
Simplex.prototype.displayResults = function(html) {
    var result = $("#result");
    result.html(html);
}

/** Display graph from results */
Simplex.prototype.displayGraph = function() {
    var si = Simplex.currentSimplex;
    var res = si.answer;

    var data = new google.visualization.DataTable();
    data.addColumn('number', "X1");
    data.addColumn('number', "X2");
    for (var i = 0; i < si.numRestricciones; i++) data.addColumn('number', "R" + i);

    // Determine max value
    var max = Math.max(res[0], res[1]);

```

```

var copy = si.backup;
copy.shift(); // names row

var rRow = copy.shift(); // objective function
rRow[rRow.length - 1] = res[res.length - 1];
rRow[2] = rRow[2]*-1;
rRow[3] = rRow[3]*-1;
var rPoints = si.getPoints(rRow);

// Get points for objective function
var x1 = rPoints[0];
var x2 = rPoints[1];
data.addRows(2 + si.numRestricciones*2);
data.setValue(0,0,x1);
data.setValue(0,1,0);
data.setValue(1,0,0);
data.setValue(1,1,x2);

for (var i = 0; i < si.numRestricciones; i++) {
    var points = si.getPoints(copy.shift());
    console.log(points);

    data.setValue(2 + (i*2), 0, points[0]);
    data.setValue(2 + (i*2), 2+i, points[0] != 0? 0:points[1]);
    data.setValue(3 + (i*2), 0, points[1] != 0 ? 0:points[0]);
    data.setValue(3 + (i*2), 2+i, points[1]);
}

var chart = new google.visualization.ScatterChart(document.getElementById("graph"));
chart.draw(data, {width: 600, height: 600,
    title: "Grafica",
    hAxis: {title: "X1", minValue:0, maxValue: max},
    vAxis: {title: "X2", minValue:0, maxValue: max},
    legend: 'right',
    lineWidth: 1
});
}

/** Get points for graph */
Simplex.prototype.getPoints = function(row) {
    console.log(row.toString());
    var x1 = row[2];
    var x2 = row[3];
    var ld = row[row.length-1];
    var res = [0,0];
    if (x1 !== 0) res[0] = ld / x1;
    if (x2 !== 0) res[1] = ld / x2;
    return res;
}

/* * * * * *
* Main Solving Function

```

```

* * * * *
Simplex.prototype.run = function() {
    console.log("running");
    var si = Simplex.currentSimplex;
    si.getData();
    var html = si.solve();
    si.displayResults(html);
    si.displayGraph();
}

/** Just creates a sample equations according to the number of variables */
Simplex.prototype.getSampleEquation = function() {
    var texto = "";
    var si = Simplex.currentSimplex;
    console.log("sE.numVar:" + si.numVariables)
    for (var i = 1; i<=si.numVariables; i++) {
        if (i != 1) texto += " + ";
        texto += "X" + i;
    }
    return texto;
}

// UI functions -----
Simplex.getRowData = function(id, ld) {
    console.log("gettingRowData");
    var r = $("#"+ id);
    console.log(r);
    var si = Simplex.currentSimplex;
    var cols = 1 + si.numVariables + si.numRestricciones + 1;

    var a = new Array(cols);
    console.log(cols);

    for (var i = 0; i < si.numVariables; i++) {
        var value = r.find("#var" + i).val();
        value = value == "" ? 0 : parseFloat(value);
        a[i+1] = value;
    }

    if (id === "objetivo") {
        a[0] = 1;
        for (var i = 0; i < si.numVariables; i++) a[i+1] = -a[i+1];
        for (var i = si.numVariables, len = cols - 1; i < len; i++) a[i+1] = 0;
        a.unshift("Z");
    }
    if (id.indexOf("restriccion") != -1) {
        a[0] = 0;
        var num = parseFloat(id.substr(11)) + si.numVariables;
        console.log("restriccionNum: " + num);
        for (var i = si.numVariables, len = cols - 1; i < len; i++) {
            a[i+1] = num == i ? 1 : 0;
        }
        var value = r.find("#var" + si.numVariables).val();
        value = value == "" ? 0 : parseFloat(value);
    }
}

```

```

        a[cols-1] = value;
        a.unshift("S" + (num-si.numVariables + 1));
    }

    return a;
}

Simplex.showOptions = function() {
    var si = new Simplex();
    si.numVariables = parseFloat($("#input#numVar").val());
    si.numRestricciones = parseFloat($("#input#numRes").val());

    Simplex.currentSimplex = si;
    $("#div#startupSection").hide();
    var inputSection = $("#div#inputSection");
    var iSE = inputSection.get(0);
    var objetivo = Simplex.createRow(si.numVariables, "objetivo");
    iSE.appendChild(Simplex.createTextLabel("Muestra: " + si.getSampleEquation()));
    iSE.appendChild(objetivo);

    iSE.appendChild(Simplex.createTextLabel("Sujeto a"));
    for (var i = 0; i < si.numRestricciones; i++) {
        iSE.appendChild(Simplex.createRow(si.numVariables, "restriccion"+i, true));
    }

    var boton = Simplex.createButton("Resolver", "resolver");
    $(boton).click(function() {
        Simplex.currentSimplex.run();
        $("#inputSection").hide();
        $("#resultSection").show();
    });
    iSE.appendChild(boton);

    inputSection.show();
}

Simplex.createTextField = function(id) {
    var fieldElement = document.createElement("input");
    var field = $(fieldElement);
    field.attr("type", "text");
    field.attr("size", "3");
    if (id) field.attr("id", id);
    return fieldElement;
}

Simplex.createButton = function(text, id) {
    var fieldElement = document.createElement("input");
    var field = $(fieldElement);
    field.attr("type", "button");
    field.attr("value", text);
    if (id) field.attr("id", id);
    return fieldElement;
}

```

```

Simplex.createTextLabel = function(text) {
    var element = document.createElement("p");
    var p = $(element);
    p.html(text);
    return element;
}

Simplex.createRow = function(rows, id, ld) {
    console.log("Rows:" + rows);
    var df = document.createElement("div");
    $(df).attr("id", id);

    for (var i = 0; i < rows; i++) {
        df.appendChild(Simplex.createTextField("var"+i));
    }
    if (ld) {
        var signo = $(document.createElement("span"));
        signo.html("&nbsp;≤&nbsp;");
        df.appendChild(signo.get(0));
        df.appendChild(Simplex.createTextField("var"+rows));
    }
    df.appendChild(document.createElement("br"));
    return df;
}

// Startup
$(new function() {
    $("div#inputSection").hide();
    $("div#resultSection").hide();
    $('#iniciar').bind("click", Simplex.showOptions);
});

// ----- SOLVER -----

var tabla; //This variable store the array obtained from the form
//These variables will help store de current location of the row and column pivots
var pivoteV;
var pivoteH;

function Crea() {

    tabla = Simplex.currentSimplex.variables;

}

//Prints the current values of the array in a readable way
function imprime() {
    var html = "<table width=90 border=1 cellpadding=1 cellspacing=1 style='table-
layout:fixed;'>";
    for (i=0;i<tabla.length;i++){
        for (j=0;j<tabla[i].length;j++){
            html += "<td width=90px height=25px style='overflow:hidden'" + tabla[i]
[j] + "</td>";
        }
        html += "</tr>";
    }
}

```

```

    }
    html += "</table>";
    html += "</br>";
    return html;
}

//checks that the Z row doesn't have negative values
function filaNegativa() {
    var negativa = 0;
    for (var i =1; i<tabla[1].length; i++){
        if(tabla[1][i] < 0)
            negativa = 1;
    }
    return negativa;
}

//finds the most negative value in the Z row and fix the pivot
function masNegativo() {
    var menor = 1;
    for (var i =1; i<tabla[1].length; i++){
        if(tabla[1][i] < menor){
            menor = tabla[1][i];
            pivoteV = i;
        }
    }
}

//Chooses the smallest value from the LD column and pivot column division
function menorCoeficiente(){
    //var menorValor = tabla[2][tabla[2].length - 1] / tabla[2][pivoteV];
    var menorValor = Number.MAX_VALUE;
    // alert(menorValor);
    for (var i=2;i<tabla.length;i++){
        var temp = (tabla[i][tabla[i].length - 1]) / (tabla[i][pivoteV]);
        if (temp <= menorValor && temp > 0){
            menorValor = temp;
            pivoteH = i;
        }
    }
    // alert(menorValor);
}

//Divides the pivot row by the pivot's value to get the 1
function igualarAUno(){
    // alert(tabla[pivoteH][pivoteV]);
    if(tabla[pivoteH][pivoteV] != 1){
        var pivoteFijado = tabla[pivoteH][pivoteV];
        for (var i =1; i<tabla[pivoteH].length; i++){
            var temp = tabla[pivoteH][i];
            tabla[pivoteH][i] = temp / pivoteFijado;
        }
    }
    tabla[pivoteH][0] = tabla[0][pivoteV];
}

```



```

//checks if the values of the pivot column are equal to 0, if not calls the
igualAACero function
function revisaFilas(){
    for(var i=1; i<tabla.length;i++){
        if(tabla[i][pivoteV] != 0 && i != pivoteH)
            igualAACero(i)
    }
}

//makes the required calculations to make the pivot column values equal to 0
function igualAACero(fila){
    var neg = -tabla[fila][pivoteV];
    for (var i =1; i<tabla[fila].length; i++){
        var temp = (neg * tabla[pivoteH][i]) + tabla[fila][i];
        tabla[fila][i] = temp;
    }
}

//It pairs the variable names to its correct value
function entregaRespuesta(){
    var si = Simplex.currentSimplex;
    var respuesta = new Array(si.numVariables + si.numRestricciones + 1);//variables
+ restricciones + 1
    for(var a=0;a<respuesta.length;a++){
        respuesta[a] = 0;
    }

    for (var i=1;i<tabla[0].length-1;i++){
        for (var j=1;j<tabla.length;j++){
            //alert(tabla[0][i]);
            //alert(tabla[j][0]);
            if(tabla[0][i] == tabla[j][0])
                respuesta[i-1] = tabla[j][tabla[0].length-1];
            //
            else
                // respuesta[i-1] = 0;
                //alert(respuesta[i-1]);
        }
    }
    var temp = respuesta.shift();
    respuesta.push(temp);
    return respuesta;
}

//This function loops the simplex method to solve the problem
function resuelve() {
    var html = "";
    while (filaNegativa()){
        masNegativo();
        menorCoeficiente();
        igualarAUno();
        revisaFilas();
        html += imprime();
    }
    var arregloRespuesta = entregaRespuesta();//the variable final values are
returned

```

```
        Simplex.currentSimplex.answer = arregloRespuesta;
        html += arregloRespuesta.toString();
        return html;
    }

    function principal() {
        Crea();
        var html = imprime();
        html+= resuelve();
        return html;
    }
```

Manual de Usuario:

Decidimos hacer nuestro programa para Web por lo que nuestra entrega consta de dos archivos simplex.js e index.html, siendo este último el principal.

Instrucciones de uso:

Al abrir el archivo principal con un explorador, nos presenta la primera pantalla en la que se introduce el número de variables y restricciones a utilizar en la resolución del problema. Y posteriormente se presiona el botón iniciar.

Simplex Solver

Introduzca el número de variables

Introduzca el número de restricciones

Esto nos lleva a la forma en la que se llenan tanto la función objetivo como sus restricciones correspondientes, escribiendo los coeficientes en las casillas indicadas.

Simplex Solver

Función objetivo

Muestra: $X1 + X2$

10

15

Sujeto a

5

10

≤

6000

2.5

2.5

≤

2250

2

1

≤

1200

Resolver

Al presionar el botón de Resolver nos aparecerán las tablas necesarias para encontrar la solución óptima, los valores de cada variable y la gráfica correspondiente.

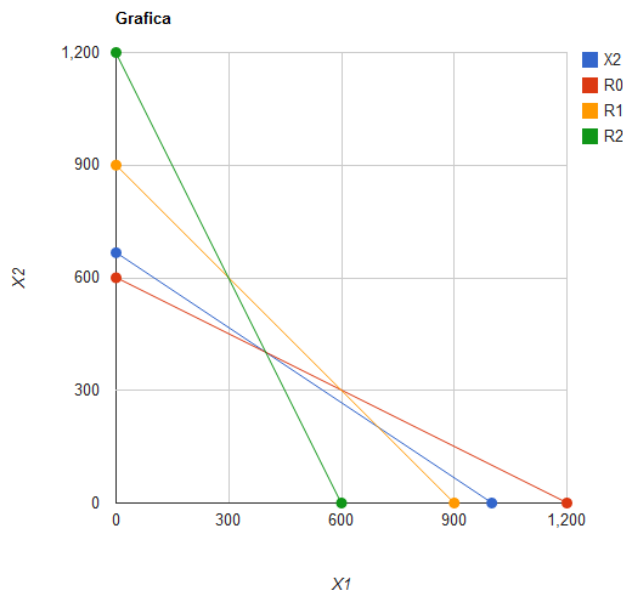
Simplex Solver

VB	Z	X1	X2	S1	S2	S3	LD
Z	1	-10	-15	0	0	0	0
S1	0	5	10	1	0	0	6000
S2	0	2.5	2.5	0	1	0	2250
S3	0	2	1	0	0	1	1200

VB	Z	X1	X2	S1	S2	S3	LD
Z	1	-2.5	0	1.5	0	0	9000
X2	0	0.5	1	0.1	0	0	600
S2	0	1.25	0	-0.25	1	0	750
S3	0	1.5	0	-0.1	0	1	600

VB	Z	X1	X2	S1	S2	S3	LD
Z	1	0	0	1.333333333	0	1.666666666	10000
X2	0	0	1	0.133333333	0	-0.333333333	400
S2	0	0	0	-0.166666666	1	-0.833333333	250
X1	0	1	0	-0.066666666	0	0.666666666	400

400,400,0,250,0,10000



Para resolver un nuevo problema solo hay que recargar la página, para regresar a la pantalla de variables y restricciones.

Notas:

- El programa solo grafica problemas con dos variables.

Comentarios y Conclusiones:

Pasar el algoritmo a código fue un proceso transparente. De igual manera la biblioteca utilizada para las gráficas facilitó la implementación de la parte gráfica. El principal reto fue la estructura de todo el programa e integrar las partes que lo componen. Usar la computadora para utilizar simplex facilita la solución de problemas con muchas variables. Como equipo nos faltó iniciar antes el desarrollo pero pudimos organizarnos.