

UNSimplexMO: Manual de Usuario

Juan David Quintero Perez
Sergio Andres Castro Castro
Juan Sebastián Vivero Jauregui
Marlon Enrique Noguera Ramírez
Cristian Camilo Cristancho Castaño

Universidad Nacional de Colombia
Facultad de Ingeniería, Departamento de Ingeniería de Sistemas e Industrial
Bogotá, Colombia
2017

Contents

| | | |
|----------|---|----------|
| 1 | Introducción | 3 |
| 2 | Prerequisitos para realizar un solver | 3 |
| 2.1 | Modelos de optimización - Programa matemático | 3 |
| 2.2 | Método de las dos fases | 4 |
| 3 | Descripción del modelo de programación | 4 |
| 3.1 | Ejecución | 5 |

1 Introducción

UNSimplexMO es un lenguaje de alto nivel en español que se encarga de resolver problemas de **Programación Lineal** ingresándolos directamente con un lenguaje matemático, tal y como se puede plantear antes de que se quiera programar.

2 Prerequisitos para realizar un solver

Para poder realizar el solver, necesitamos unos conocimientos previos para entender cómo funciona este lenguaje, el cual se enfocara en el método simplex que se realizará en dos maneras una por la gran M y la otra por medio de las dos fases.

Este método (*simplex*) es visto en libros de investigación y operaciones que tiene como principal objetivo encontrar el valor óptimo del modelo de un problema bajo ciertas restricciones.

Un modelo en este caso es un esquema de un sistema que se elabora para facilitar su comprensión y el estudio de su comportamiento, el cual posee una estructura general compuesto por:

2.1 Modelos de optimización - Programa matemático

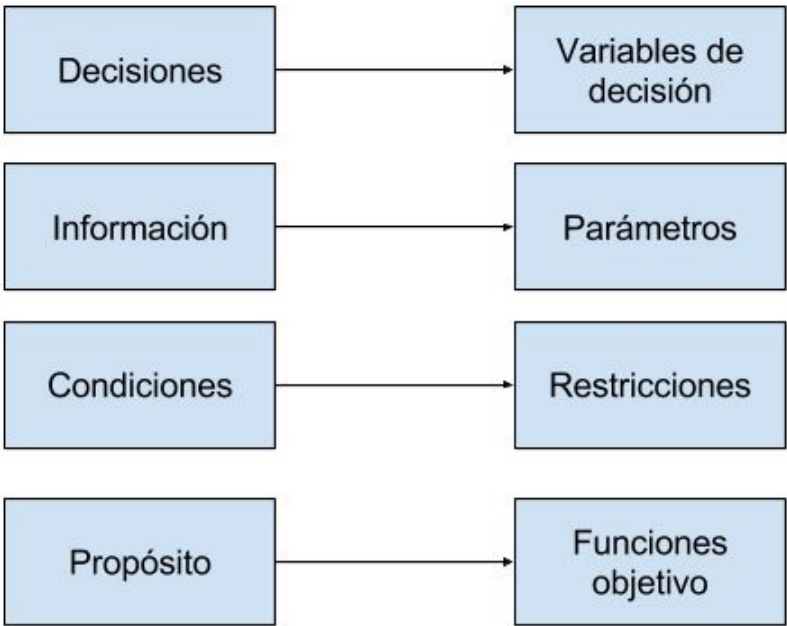


Imagen 1: Diagrama de la estructura de un problema de optimización.

Ahora explicaremos esta estructura desde el programa matemático, debido a que este es el enfoque a desarrollar para este trabajo, además cabe resaltar que esta estructura es lo que conocemos como la programación lineal.

- **Variables de decisión:** Es un elemento que tiene un dominio y que su valor puede afectar el resultado del problema.
- **Parámetros:** Son elementos que permiten lograr evaluar una determinada situación, y siempre son valores que se dan en el modelo.
- **Restricciones:** Son elementos que se dan en el modelo que nos permiten limitar su alcance.
- **Funciones objetivo:** Es la ecuación que será optimizada dada las anteriores partes de la estructura mencionadas, y con variables que necesitan ser maximizadas o minimizadas.

2.2 Método de las dos fases

Este método es otra alternativa dentro del algoritmo simplex, en el cual surge o se da debido a que el método de la gran M puede tener problemas de redondeo o escala, y sus dos fases constan de:

- **Fase I:** Esta busca minimizar las variables artificiales, y si el Z óptimo es 0, se sigue con la siguiente fase, de lo contrario no tiene solución.
- **Fase II :** Se realiza lo que hay inicialmente en el problema, teniendo como base la tabla de la anterior fase, se elimina las variables artificiales de las restricciones del modelo, luego reemplazamos la función objetivo en la función objetivo original y a partir de esto se resuelve por el método simplex normal.

3 Descripción del modelo de programación

El formato de programación es en texto plano donde se recibe las variables a usar, la función objetivo, las restricciones, las restricciones de no negatividad

Cabe resaltar que la estructura mencionada anteriormente debe estar resuelta o dada para el usuario para que este software (*solver*), pueda colocar esta estructura dentro del programa y así evaluar las variables con sus respectivos parámetros y restricciones, para encontrar el óptimo.

- **Comienzo del programa (Dantzig)** Se empleó el nombre de la persona que creó el método simplex básico para empezar nuestro programa.

```
Dantzig
...
```

- **Variables (variabilis)** Aquí se agregan las variables que se vayan a emplear en el problema.

```
variabilis
  x
  y
  ...
```

- **Función objetivo (maximieren, minimieren)** Aca se declaran las funciones objetivo donde se declaran de una vez con lo que se quiera hacer con este.

```
maximieren
  10x+3y
  ...
```

- **Restricciones (cortapisa)** Se definen las restricciones todas las necesarias después de esta palabra, como se vio anteriormente, se escriben empleando la notación matemática.

```
cortapisa
  x + y >= 130
  2.5x + y <= 250
  ...
```

- **Restricciones de no negatividad (fines)** Se deben declarar las variables empleadas en el programa con la restricción de mayor o igual a 0 (≥ 0) es recomendable no usarla, ya que la mayoría de problemas de optimización se plantean con cantidades físicas, y estas no son negativas.

```
fines
  x >=0
  y >=0
  ....
```

- **Fin del Programa (MayoDe2005)** Se tiene esta fecha como fin del programa, en honor a la muerte de Dantzig, se decidió emplearlo como palabra clave que indique el final del programa.

...

MayoDe2005

3.1 Ejecución

Para la ejecución de este lenguaje se necesita de:

- **Java:** Se emplea la versión JDK 1.8+ para la compilación del código.
- **ANTLR:** Se emplea la versión 4.7+ para la creación de los visitors y los demás archivos necesarios. Para esto es recomendable ver el manual que se encuentra en la página ya que este está constantemente actualizado [1].

Hay 2 alternativas para la ejecución del programa:

- Se puede emplear Eclipse en la versión Mars 2 importando el proyecto que se encuentra en la carpeta *UNSimplexMO-eclipse.zip*. Con el proyecto ya importado, se ejecuta normalmente como un proyecto de java, modificando el código en el archivo *test.smp* que se encuentra en la carpeta test
- Se puede ejecutar en consola estando dentro de la carpeta que el archivo *UNSimplexMO-console.zip* ejecutando los siguientes comandos:

```
antlr4 Simplex.g4 -visitor
javac *.java
java Main
```

Con esto ya se puede ejecutar un programa que se encuentre en la misma carpeta que el programa.

References

- [1] Página Oficial de ANTLR - www.antlr.org