UNIVERSIDAD DE BUENOS AIRES



**Facultad de Ingeniería**

***Modelos y Optimización I (71.14)***

**Carpeta de Trabajos Prácticos**

|  |  |  |
| --- | --- | --- |
| ***Padrón*** | ***Nombre*** | ***Email*** |
| 95470 | Gayoso, Gabriel Eliseo | ga-yo-so@hotmail.com |
| 95092 | Salas, Cristian Gustavo | cris\_gus\_10@hotmail.com |
| 96480 | Keklikian, NicolásRoberto | nkeklikian@gmail.com |

**Índice**

Consignas de la Segunda Entrega.................................3

Segunda Entrega...........................................................7

# Parte A

El problema de las elecciones de la AFA podría ser catalogado como un problema de Satisfacibilidad Booleana. Podemos definir que un problema SAT es un problema donde, a partir de una expresión booleana formada a partir de variables y operadores (trabajaremos con And, Or y Not), queremos determinar si existe una asignación de valores para las variables que derivan en que la expresión sea verdadera. Por ejemplo, podemos tener una expresión booleana de la siguiente forma:



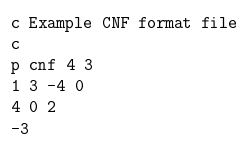
Para el estudio de este tipo de problemas, se puede utilizar la notación CNF:

* Los archivos suelen comenzar con una sección de comentarios que define el problema. Los comentarios en estos archivos se indican con la letra **c.**
* Luego de la sección de comentarios viene una línea donde se define el problema. Esta línea es de la forma:

p cnf VARIABLES CLAUSULAS

* Por último, viene la definición de cada una de las cláusulas. Las variables se numeran de 1 a n. Si el número es negativo quiere decir que la variable aparece en forma negada. Las cláusulas pueden expresarse en más de una línea. Cada clausula termina cuando se ingresa un **0**.

Por ejemplo, podríamos expresar la expresión booleana en notación CNF[[1]](#footnote-2):



Los softwares como GLPK o CPLEX cuentan con herramientas especiales para resolver este tipo de problemas. Por ejemplo, para obtener la solución de un problema expresado en este formato con GLPK deberíamos ejecutar el comando glpsol con la opción --minisat:

*glpsol --minisat --cnf <archivo\_cnf> [-o <archivo\_output>]*

Podríamos modificar el objetivo de un problema de SAT de forma tal de que en vez de tratar de determinar un valor de variables para que la expresión sea verdadera, busquemos maximizar la cantidad de cláusulas verdaderas. En este caso, en vez de ser un problema de decisión, sería uno de maximización.

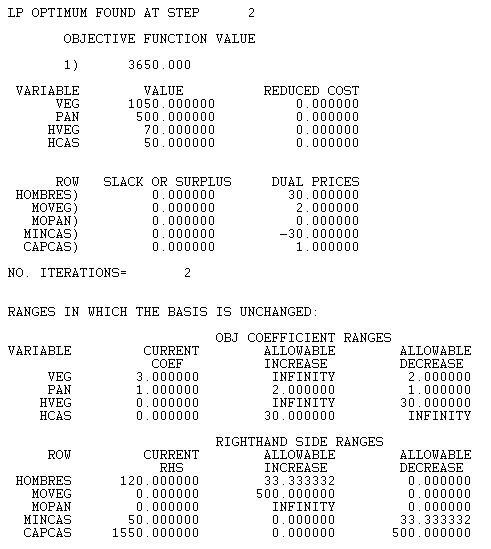
Se pide:

1. Armar una expresión booleana a partir del enunciado de la primera entrega del TP. Expresarlo también en formato CNF.
2. Utilizando los comandos incorporados por GLPK o CPLEX para la resolución de problemas SAT, se pide explicar la solución obtenida luego de ejecutar el programa para los conjuntos de datos:
   1. Elaborado en el punto 1.
   2. datos2.cnf*[[2]](#footnote-3)*
   3. datos3.cnf*[[3]](#footnote-4)*
   4. datos4.cnf[[4]](#footnote-5)
3. Desarrollar un programa, que a partir de archivos en formato cnf, genere modelos para resolver el problema de **maximización** utilizando programación lineal. Explicar lo obtenido luego de ejecutar el programa para los conjuntos de datos:
4. Elaborado en el punto 1
   1. datos2.cnf
   2. datos3.cnf
   3. datos4.cnf
5. Tomando como base las ejecuciones sobre el archivo *datos4.cnf*, graficar, en función del tiempo de ejecución, el actual valor del mejor funcional obtenido y el mejor funcional que podría llegar a obtenerse. Compararlo con la solución óptima en caso de ser conocida.
6. Desarrollar una heurística que reciba como entrada archivos en formato cnf y obtenga una solución para el problema de maximización de SAT. Explicar lo obtenido luego de ejecutar el programa para los conjuntos de datos:
   1. Elaborado en el punto 1
   2. datos2.cnf
   3. datos3.cnf
   4. datos4.cnf
7. Comparar los resultados obtenidos en los puntos 2, 3 y 4.

En todo momento, pueden interrumpir la ejecución de una corrida luego de tres horas sin llegar a un resultado. En esos casos capturar la pantalla luego de interrumpirla.

# Parte B

Cada una de las siguientes preguntas debe responderse a partir de los resultados de la corrida de LINDO indicada a continuación, sin realizar una nueva corrida del software. Si considerás que la información del resultado del software no es suficiente, indicá qué datos faltan y qué se puede decidir con la información disponible. Cada pregunta se responde en forma independiente.



* Contestar Verdadero o Falso y justificar.

1. Conviene cosechar vegetales antes que pan. Si por ejemplo disponemos de un hombre más, podremos aumentar nuestra cantidad de raciones en 15\*3 = 45.
2. Sin importar la cantidad de hombres que se disponga (superior a 50), siempre se garantizará un mínimo de 500 kg de pan ya que tenemos que destinar al menos 50 hombres al castillo y cada uno de estos hombres puede hornear hasta 10 kilogramos de pan.

* Les ofrecen 50 hombres de una aldea vecina que se dedica a hacer pimienta que podría encargarse de la tarea de cuidar el castillo. A cambio les piden una cierta cantidad de raciones. ¿Hasta cuántas raciones estarían dispuestos a dar a cambio de los 50 hombres?
* ¿Cómo se vería modificada nuestra solución si se descubriera un proceso que nos permitiera aumentar a 4 la cantidad de raciones que se saca a partir de un kilo de pan? Dar el nuevo valor del funcional o, en caso de modificarse la estructura de la solución óptima, una cota del valor del mismo.
* ¿Qué creen que pasa si reducimos la cantidad de hombres disponibles a, por ejemplo, 115?

**PARTE A**

**1)**

Manteniendo la misma nomenclatura que usamos en la primera entrega:

* PContFPT (X1): posición a favor de la continuidad de FPT.
* PSusE (X2): posición a favor de la suspensión de los estadios en malas condiciones.
* PBB (X3): posición en contra de las barras bravas.
* POf (X4): posición a favor del oficialismo.

Entonces, la expresión booleana del problema resulta:

En formato CNF:

p cnf 4 5

-1 3 0

1 4 0

-2 0

1 -3 0

2 3 0

**2)**

a. Cuando se corrió el programa se pudo llegar a una solución que cumpliera con todas las condiciones. Esto era de esperar, ya que este mismo problema había sido resuelto anteriormente a mano, por lo que un programa especifico para el caso como GLPK evidentemente debería poder resolverlo.

b. Al ejecutar estos datos se pudo llegar a correr exitosamente GLPK. Sin embargo no se pudo encontrar una solución que verificara todas las condiciones. Sin embargo, debido a que este es un problema de decisión, no se pudo saber si existía alguna opción que se le acerque.

c. Debido a la inmensa cantidad de datos de este conjunto la corrida de GLPK no pudo concluir en el tiempo establecido. A su vez, como lo que busca es encontrar si existe o no alguna combinación que permita que se cumplan todas las condiciones al mismo tiempo, el cortarlo en mitad del proceso causó que no se generase ningún resultado. Por lo tanto no podemos saber si es posible o no.

d. Nuevamente, con un set de datos de menor tamaño, se pudo correr con éxito el programa. A su vez se pudo ver en los resultados que era posible conseguir una solución que permitiese verificar todas las condiciones.

**4)**

a. Al correrlo como un problema de maximización se llegó a un funcional de 5. Nuevamente se corrió sin problemas dentro del tiempo requerido.

b. Corriendo el programa con este set de datos tomado como un problema de maximización se pudo observar qué tan cerca había estado de cumplirse el objetivo en la corrida SAT. En este caso el funcional dio 217, que es uno menos que las 218 restricciones a cumplir.

c. Igual que la otra vez, no se pudo correr el programa en el intervalo de tres horas, por lo cual se tuvo que interrumpir cumplido ese tiempo. Se esperaba que al menos esta vez, corriendo como problema de maximización, se obtuviera al menos un resultado parcial. Sin embargo al observar los resultados, se observó que el funcional se mantenía en 0. Esto quiere decir que no llegó a generar que ni una condición se cumpla. Como posible explicación suponemos que en el procesamiento interno GLPK debió mantenerse dentro de variables ficticias creadas para resolver el problema y nunca pudo entrar a modificar las variables del planteo original.

d. Cuando se corrió el set de datos 4 se pudo llegar a terminar el proceso en un tiempo poco menor a las 3 horas. Como se esperaba teniendo en cuenta su resolución SAT, el funcional es equivalente a la cantidad de restricciones. Z = 645.

**5)**

A continuación se expone el gráfico de el funcional en base al tiempo de ejecución. Sin embargo cabe aclarar que en el intervalo de tres horas se pudo efectivamente correr en su completitud el conjunto datos4 como problema de maximización. Por lo tanto el resultado obtenido ya es el óptimo.

**6)**

Al desarrollar la heurística primero pensamos en priorizar darle valor a las variables que aparecieran en más restricciones, lo que aumenta directamente el funcional. Esta idea tenía la falla de que si luego de darle valor a un conjunto de variables, quedan restricciones que solo pueden cumplirse por una variable, ya que las demás son 0, la heurística no lo considera y prioriza seguir sumando restricciones que se podrían cumplir de muchas formas todavía si es que sus variables aparecen en una mayor cantidad de restricciones.

Entonces decidimos asignarle un peso a cada variable dentro de cada restricción, inverso a la cantidad de variables sin definir en ella. Así, una variable en una restricción formada por ORs de tres variables tendrá un peso de , mientras que una variable indispensable para cumplir una restricción tendrá peso. Luego se calcula el peso de cada variable como la suma de los pesos de la variable en cada restricción que aparece, y en cada paso se le da valor a la variable con mayor peso.

Esta idea significó una mejora en el funcional, pero pensamos que el peso de cada variable dentro de la restricción podía ser cualquier función de la cantidad de variables en la restricción, no necesariamente el inverso multiplicativo. Así, haciendo pruebas con distintas funciones, se llegó a la expresión , que deja una restricción sin cumplir para el conjunto datos2.cnf, cincuenta y una restricciones sin cumplir para el conjunto datos3.cnf y una restricción sin cumplir para el conjunto datos4.cnf.  
Sabíamos que el conjunto datos4 tiene solución e intentamos variar parámetros para lograr una heurística que la encontrara sin perder efectividad para los otros conjuntos. Esto no lo logramos, cuando lográbamos que resolviera datos4.cnf, para datos2.cnf le faltaban cinco o más restricciones. Y para datos3.cnf le faltaban más de sesenta.

**PARTE B**

**1)**

A) Falso

Si bien es cierto que conviene cosechar vegetales antes que pan, el ejemplo propuesto no es verídico. Lo sería si no existiera la restricción de capacidad del castillo. Como se puede ver en el resultado de la corrida con LINDO, el valor marginal de relajar la restricción de hombres en una unidad (es decir, tener un hombre más) es de 30. Esto quiere decir que el funcional mejorará en 30 unidades, lo cual indica que se producen 30 raciones más, no 45. Esto es porque con el resultado obtenido ya se llena la capacidad del castillo. Para almacenar los 15kg de vegetales que cosecharía un hombre extra, habrá que perder 15kg de pan, por lo tanto se ganan 45 raciones pero se pierden 15.

B) Falso

Lo que garantiza dicha restricción es que siempre habrá como mínimo 50 hombres en el castillo horneando pan. Pero con cada hombre por encima de 70 produciendo vegetales se tendrán más kg de los mismos, que proveen más raciones por kg que el pan. Entonces, por cada hombre extra luego de los 120 hombres se tendrá 15kg menos de pan y 15kg más de vegetales (conviene tirar la producción de pan, y quedarse con los vegetales). Luego de los 153.33 hombres, podemos ver que el valor marginal de dicho recurso deja de ser válido, esto es porque en dicho punto se logra mantener 50 hombres en el castillo y aun así llenar los 1550kg del castillo de vegetales (habrán entonces 0kg de pan). Desde este punto en adelante, cada hombre extra no aporta nada a la cantidad de raciones producidas.

**2)**

Como se dijo en el punto anterior, solo nos sirven 33,33 hombres más y por cada uno se verá una ganancia de 30 raciones, llegando a un total de 4650 raciones producidas. Esto nos dice que los 50 hombres extra resultarán en un aumento de 1000 raciones, por lo que sería conveniente dar hasta dicho numero de raciones a cambio de los hombres, sin que resulte en una pérdida.

**3)**

Si se pudiera obtener 4 raciones de 1kg de pan, se tendría que cada hombre en el castillo produce 40 raciones, mientras que cada hombre cosechando vegetales produce 45 raciones. Como la cantidad de hombres disponible es limitada, la solución óptima también involucraría tener 50 hombres en el castillo y 70 cosechando vegetales, para llegar a un total de 5150 raciones. De este punto en adelante, cada hombre que produzca generará una pérdida de 5 raciones pero liberará 5kg de espacio en el castillo. Esto quiere decir que si bien dos hombres cosechando vegetales generan 90 raciones, dos hombres horneando pan generan 80 raciones y permiten que un hombre extra produzca 40 raciones más. Si la cantidad disponible de hombres no fuera un problema (se necesitan 155 hombres), sería óptimo que todos hornearan pan para llenar los 1550kg de capacidad y así producir 6200 raciones.

**4)**

Como se deben llenar los 50 hombres en el castillo, deberá disminuir la cantidad de hombres cosechando vegetales. De este modo, se llegará a un funcional de 3425 raciones, menor que el anterior.

1. Para leer más de la notación se puede consultar el documento referenciado desde : <http://www.cs.ubc.ca/~hoos/SATLIB/benchm.html> [↑](#footnote-ref-2)
2. Extraído del banco de datos SATLIB - Benchmark problems: <http://www.cs.ubc.ca/~hoos/SATLIB/benchm.html> [↑](#footnote-ref-3)
3. Extraído de la competición SAT 2014. URL:

   <http://goo.gl/ESKGXy> [↑](#footnote-ref-4)
4. Extraído del banco de datos SATLIB - Benchmark problems: <http://www.cs.ubc.ca/~hoos/SATLIB/benchm.html> [↑](#footnote-ref-5)