

TRABAJO FINAL - 2 PGE

Diego Fernando Marquez Betancur c.c: 8.355.196

Requerimientos aplicación de software genérico para el método Dantzing-Wolf. (generación de columnas)

El presente son los requerimientos que abarca el diseño, desarrollo y puesta en marcha de una pequeña aplicación de software genérico para el método Dantzing-Wolf (Generación de Columnas).

Alcance del proyecto

Se realizará el diseño y puesta en ejecución del software para el método Dantzing-Wolf (Generación de Columnas). El cual está compuesto por los siguientes módulos:

1. Aplicación de software genérico (GAMS - EXCEL).

1.1. Módulo GAMS.

1.1.1. Implementación de un modelo genérico del método Dantzing-Wolf (Generación de Columnas) Mediante un ejercicio de Formulación de programación lineal.*

1.2. Módulo Visual o Interfaz Gráfica en EXCEL.

1.2.1. Implementación de una interfaz gráfica con Excel mediante macros creados en VBA, esto para permitir una fácil manipulación e ingreso de las variables, función objetivo entre otros.

1.3. Drivers.

1.3.1. Solvers de varios tipos en este caso programación lineal y no lineal para Excel.

1.3.2. Gams previamente instalado con todas sus librerías

1.4. Tecnologías.

1.4.1. El desarrollo de la aplicación se realizará en Gams y Excel y no es necesario configuraciones adicionales ya que la idea del producto será genérica y se pueda ejecutar en cualquier máquina que contenga el Microsoft Office dentro del sistema operativo Windows.

Metodología de trabajo

- Para este caso se realizará 2 entregas del software:

Entrega 1: El código o Archivo ejecutable en GAMS de un problema genérico de programación lineal. Adjunto

Entrega 2: La integración de la interfaz gráfica en EXCEL y GAMS esto con las posteriores pruebas.

Entrega 1: Archivo GAMS: *

Problema de prueba: Anexo ejecutable en GAMS. (Dantwolfe.gms)

El coordinador de una Agencia Central debe adquirir servicios de buques cisterna para ayudar a su distribución. Un subcontratista maneja todo el detalle de envío. Este escenario se usa para demostrar el principio de descomposición (Mediante Dantzing-Wolf) en el envío de su producto desde dos plantas a cuatro terminales. el personal odia los detalles y le ha solicitado a Sub que le proporcione solo dos números, el costo, c , del plan de envío propuesto y el número de tanqueros, t , necesarios para respaldarlo. El distribuidor, sub, tiene dos matrices, una matriz de costo unitario c_{ij} y una unidad de requerimiento de matriz de tanqueros t_{ij} (se requieren petroleros por unidad enviada). (Los ceros en la matriz t_{ij} indican envíos por tubería en lugar de por bote).

Desde el objetivo general de minimizar los costos, sub establece para resolver el problema de transporte, a continuación, donde a_i son las disponibilidades conocidas en las plantas y b_j son los requisitos en las terminales.

\$title Decomposition Dantzing-Wolf - Animated (DECOMP,SEQ=164)

sets i plants / plant-1, plant-2 /

j terminals / term-1*term-4 /

table c(i,j) cost matrix

	term-1	term-2	term-3	term-4
plant-1	3	6	6	5
plant-2	8	1	3	6

table t(i,j) tankers required

	term-1	term-2	term-3	term-4
plant-1		2		
plant-2		2		

parameters a(i) availability / plant-1 9, plant-2 8 /

b(j) requirements / term-1 2, term-2 7, term-3 3, term-4 5 /

ctank tanker cost

cship shipping cost

variables cost total cost

tank total tankers used

ship shipping cost

x(i,j) shipments

positive variable x

equations defcost cost definition

defship shipping cost

deftank tanker usage

supply(i) supply balance

demand(j) demand balance ;

defcost.. cost =e= cship*ship + ctank*tank;

defship.. ship =e= sum((i,j), c(i,j)*x(i,j));

deftank.. tank =e= sum((i,j), t(i,j)*x(i,j));

supply(i).. sum(j, x(i,j)) =L= a(i);

demand(j).. sum(i, x(i,j)) =g= b(j);

model sub / defcost, defship, deftank, demand, supply /;

sets ss master column labels / 1*10 /

s(ss) active columns;

parameter mcost(ss) cost solutions

mtank(ss) tanker solutions;

variables mobj

lam(ss) column selection

positive variable lam;

equations cbal cost balance

tbal tanker balance

convex combination;

cbal.. mobj =e= sum(s, mcost(s)*lam(s));

tbal.. sum(s, mtank(s)*lam(s)) =l= 9;

convex.. sum(s, lam(s)) =e= 1;

model master / cbal, tbal,convex /;

parameter rep(ss,*);

*** --- obtener la primera solución sin costo para los buques tanque**

cship = 1;

ctank = 0;

solve sub using lp minimizing cost;

mcost('1') = ship.l;

mtank('1') = tank.l;

*** --- obtener una segunda solución sin costo para los buques tanque**

option limcol=0,limrow=0;

solve sub using lp minimizing tank;

mcost('2') = ship.l;

mtank('2') = tank.l;

*** --- resolver el primer problema maestro**

s('1') = yes; s('2') = yes;

solve master using lp minimizing mobj;

rep('2','obj') = mobj.l;

rep('2','s-pi') = convex.m;

rep('2','t-pi') = -tbal.m;

rep('2','gap') = inf;

*** --- ahora estamos listos para iterar entre el problema principal y secundario;**

loop(ss\$((not s(ss)) and (rep(ss-1,'gap') > .01)),

ctank = -tbal.m;

solve sub using lp minimizing cost;

mcost(ss) = ship.l;

mtank(ss) = tank.l;

s(ss) = yes;

solve master using lp minimizing mobj;

rep(ss,'obj') = mobj.l;

rep(ss,'s-pi') = convex.m;

rep(ss,'t-pi') = -tbal.m;

```

rep(ss,'bnd') = rep(ss-1,'obj') - rep(ss-1,'s-pi')
               + mcost(ss) + mtank(ss)* rep(ss-1,'t-pi');

rep(ss,'best-bnd') = max(rep(ss-1,'best-bnd'),rep(ss,'bnd'));

rep(ss,'gap')    = rep(ss,'obj') - rep(ss,'best-bnd'); );

display mcost, mtank, rep;

```

Entrega 2: La integración de la interfaz gráfica en EXCEL y GAMS esto con las posteriores pruebas.

	1	2	3	4	5	6	7	8	9
GAMS Directorio:	C:\GAMS\win64\24.9\								
GAMS Modelo:	C:\Users\Diego Márquez\Documents\PGE\Trabajo_Final\DWGams\Dantwolfe.gms								
GAMS Parametros:									
Iteraciones:	1	2	3	4	5	6	7	8	9
Job Date	24/11/2017	24/11/2017	24/11/2017	24/11/2017	24/11/2017	24/11/2017	24/11/2017	24/11/2017	24/11/2017
Job Time	21:41:02	21:41:02	21:41:02	21:41:02	21:41:02	21:41:02	21:41:02	21:41:02	21:41:02
Nombre del Solver	CPLEX	CPLEX	CPLEX	CPLEX	CPLEX	CPLEX	CPLEX	CPLEX	CPLEX
Estado del Solver	1	1	1	1	1	1	1	1	1
Estado del Modelo	1	1	1	1	1	0:00	1	1	1
FO: Valor Z	53	0	74	80,333333	60,8	87	60	87	60
Tiempo (sec.)	0,016	0,031	0,031	0,031	0,016	0,015	0,031	0,047	0,032

Fig.1 – Interfaz de Excel para la ejecución de Dantzing-Wolf

Se realiza la integración de la interfaz grafica en EXCEL y GAMS para la invocación del algoritmo de Dantzing-Wolf mediante macros en VBA(Visual Basic Application), este se realiza mediante la carga del archivo Dantwolfe.gms y su posterior corrida como se muestra en la **Fig 1**. El resultado se muestra directamente en Excel para su posterior análisis, para este microproyecto el tiempo fue muy corto ya que unas de las metas propuestas era ingresar cada uno de los parámetros al Excel y que este se resolviera llamando al gams mostrando sus respectivas soluciones de los subproblemas como las soluciones de los problemas primales.

Nota: Se debe instalar gams como ejecutable para Windows este se encuentra en la carpeta del todo el micro proyecto esto para poder correrlo todo.