

PROYECTO FIN DE GRADO
INGENIERÍA Y TECNOLOGÍAS INDUSTRIALES

PROGRAMACIÓN DE TRABAJOS EN
MÁQUINAS EN PARALELO CON
RELACIONES DE PRECEDENCIA

Autor: Daniel Mena Jiménez
Tutor: José Manuel García Sánchez

Dep. de Organización Industrial y Gestión de Empresas I
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla
Sevilla, 2015



ÍNDICE

1. INTRODUCCIÓN Y OBJETIVOS	4
2. PLANIFICACIÓN DEL MODELO Pm/Prec/Cmax	7
2.1 INTRODUCCIÓN	7
2.2 PROGRAMACIÓN DE TRABAJOS EN MÁQUINAS	7
2.3 MODELO Pm/prec/Cmax	16
2.4 HEURÍSTICAS	20
2.5 Pm/m>n, prec/Cmax	24
2.5.1 Introducción	24
2.5.2 Gestión de Proyectos.....	24
2.5.3 Pm/m>=n, prec/Cmax	24
3. IMPLEMENTACIÓN Y RESOLUCIÓN DEL MODELO Pm/prec/Cmax	27
3.1 INTRODUCCIÓN	27
3.2 DESCRIPCIÓN DE LA IMPLEMENTACIÓN.....	27
3.3 VISUAL BASIC	29
3.3.1 Introducción	29
3.3.2 Estructura	29
3.3.3 Funciones del modelo y heurísticas.....	29
3.3.4 Pasos para la ejecución de las funciones	33
3.4 LINGO.....	36
3.4.1 Introducción	36
3.4.2 Sintaxis	37
3.4.3 Pasos para la resolución del modelo	39
4. RESULTADOS DE LA EXPERIMENTACIÓN	45
4.1 INTRODUCCIÓN	45
4.2 BATERIA DEL PROBLEMA	45
4.3 COMPROBACIÓN DEL MODELO Y HEURÍSTICAS	48
4.4 RESULTADOS OBTENIDOS.....	51

4.5 CONCLUSIONES.....	54
5 CONCLUSIONES GENERALES	56
BIBLIOGRAFÍA	57
ANEXO	59

1. INTRODUCCIÓN Y OBJETIVOS

1. INTRODUCCIÓN Y OBJETIVOS

En este proyecto se estudia el comportamiento del modelo de un determinado problema con máquinas en paralelo, así como la eficiencia de métodos heurísticos.

El problema a resolver es $Pm/pre/Cmax$, que significa m máquinas en paralelo, en las cuales los trabajos deben cumplir unas relaciones de precedencia y cuyo objetivo es minimizar el tiempo de finalización de todos los trabajos.

Para la comparación entre el óptimo del problema, se crea una batería de problemas con diferentes características. En todos los casos las máquinas se consideran idénticas. El modelo es creado con el uso de “Visual Basic”, que es un lenguaje de programación dirigido por eventos, posteriormente es resuelto con “Lingo”, que es la librería de optimización que hemos elegido. Por último se han implementado heurísticas con el uso de “Visual Basic”.

Las heurísticas que se compararan con el modelo son:

- CP (CriticalPath): Esta heurística consiste en dar la mayor prioridad a los trabajos con una mayor cadena de trabajos.
- CP (CriticalPath), con tiempo de proceso: Esta heurística sigue el mismo procedimiento que la anterior (CP), pero también se tiene en cuenta que tiempo de proceso sigue a un trabajo.
- LNS (LargestNumber of Successors): Esta heurística consiste en dar la mayor prioridad a los trabajos con mayor número de sucesores.
- LNS (LargestNumber of Successors), con tiempo de proceso: Esta heurística sigue el mismo procedimiento que la anterior (LNS), pero también se tiene en cuenta que tiempo de proceso sigue a un trabajo.

La documentación que forma el proyecto está constituida por los siguientes capítulos:

En este primer capítulo se ha realizado una breve introducción sobre el proyecto realizado, en el capítulo dos se hace una introducción teórica sobre la secuenciación de máquinas y los factores que determinan estos problemas, después se describe el modelo concreto del problema que estudiamos y se explica las diferentes heurísticas. Por último hablaremos de un caso particular de nuestro problema en el cual el número de máquinas es mayor o igual que el número de trabajos ($m \geq n$), debido a que este tipo de problema es uno de los primeros métodos que origina la gestión de proyectos.

En el capítulo tercero se explicará detalladamente como se crea el modelo de nuestro problema y las diferentes heurísticas con el programa Visual Basic y como se obtiene las soluciones mediante Lingo.

En el capítulo cuatro analizaremos las soluciones del modelo, las compararemos con las heurísticas y se estudiará la eficiencia de estas.

Finalmente en el capítulo cinco comentaremos las conclusiones que hemos obtenidos en el proyecto.

2. PLANIFICACIÓN DEL MODELO Pm/Prec/Cmax

2. PLANIFICACIÓN DEL MODELO Pm/Prec/Cmax

2.1 INTRODUCCIÓN

En este apartado se explica la secuencia de trabajos en máquinas, después se habla de nuestro problema, Pm/prec/Cmax, en el que planteamos su modelo y diferentes heurísticas para su resolución. Por último se menciona un caso particular de nuestro problema, Pm/m>n,prec/Cmax, en el que el número de máquinas es mayor o igual que el número de trabajos a realizar.

2.2 PROGRAMACIÓN DE TRABAJOS EN MÁQUINAS

La secuenciación de trabajos en máquinas, universalmente conocido como Scheduling, se puede definir como la asignación en el tiempo de los recursos disponibles con objeto de optimizar una determinada medida de comportamiento.

A partir de un determinado criterio, se trata de establecer la secuencia para el procesamiento de una serie de trabajos sobre un conjunto de máquinas.

Existe un amplio espectro de características que pueden asociarse a los trabajos y al modo de procesamiento en el sistema.

Un problema se determina según tres factores:

- La arquitectura del taller.
- Las características de los trabajos.
- El criterio de optimización

La arquitectura del taller:Atiende a la disposición de las maquinas en el taller.

1. Maquinas en serie:

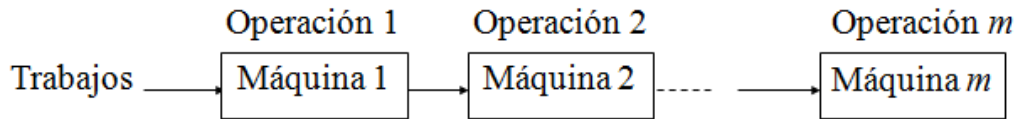


Ilustración 1

Los entornos de máquinas en serie se clasifican en función del modelo o esquema de paso de los trabajos por las diferentes máquinas:

- Sistema de flujos uniforme (Flow Shop): El modelo de paso es el mismo para todos los trabajos. Todos los trabajos pasan por cada una de las máquinas del sistema usando el mismo orden de paso por las mismas.
- Sistema de tipo taller (Job Shop): Cada trabajo tiene su propio esquema de paso por las máquinas.
- Sistema de taller abierto (Open Shop): El modelo de paso de cada trabajo por las máquinas es libre.

2. Máquinas en paralelo:

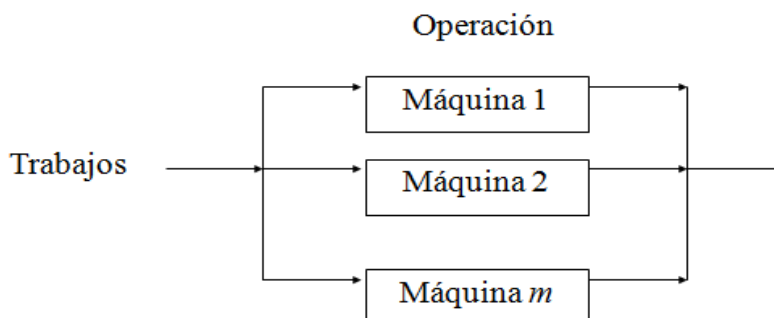


Ilustración 2

Respecto a los sistemas de máquinas en paralelo se distinguen tres tipos:

- Máquinas idénticas: El tiempo de proceso de una operación es idéntico en cada máquina.
- Máquinas uniformes: Cada máquina pose una velocidad de proceso diferente, independiente de los trabajos.
- Máquinas no relacionadas: Cada máquina posee una velocidad de proceso diferente sobre cada trabajo.

3. Sistema híbrido: m centros o estaciones de máquinas en serie, cada una de ellas formado por un conjunto de máquinas en paralelo.

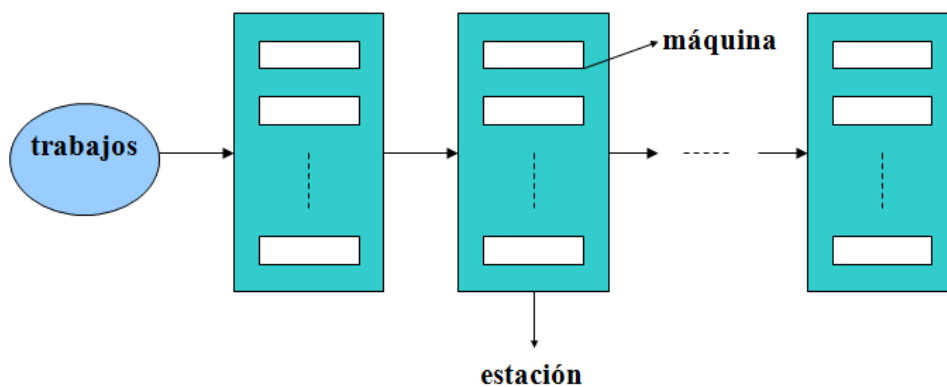


Ilustración 3

Para la representación de los trabajos que se realiza en cada máquina utilizamos los Diagramas de Gantt.

El Diagrama de Gantt es una popular herramienta gráfica cuyo objetivo es mostrar el tiempo de dedicación previsto para diferentes tareas o actividades a lo largo de un tiempo total

determinado. A pesar de esto, el diagrama de Gantt no indica las relaciones existente entre cada trabajo.

Esta representación se realiza sobre un eje horizontal y otro vertical donde cada eje representa la siguiente información:

- En el eje horizontal: Un calendario, o escala de tiempo definido en términos de la unidad más adecuada al trabajo que se va a ejecutar: hora, día, semana, mes, etc.
- En el eje vertical: El número de máquinas que disponemos para realizar todos los trabajos.

En función a los dos ejes se representa todos los trabajos que se deben realizar, según el espacio temporal y la máquina en la que se realice.

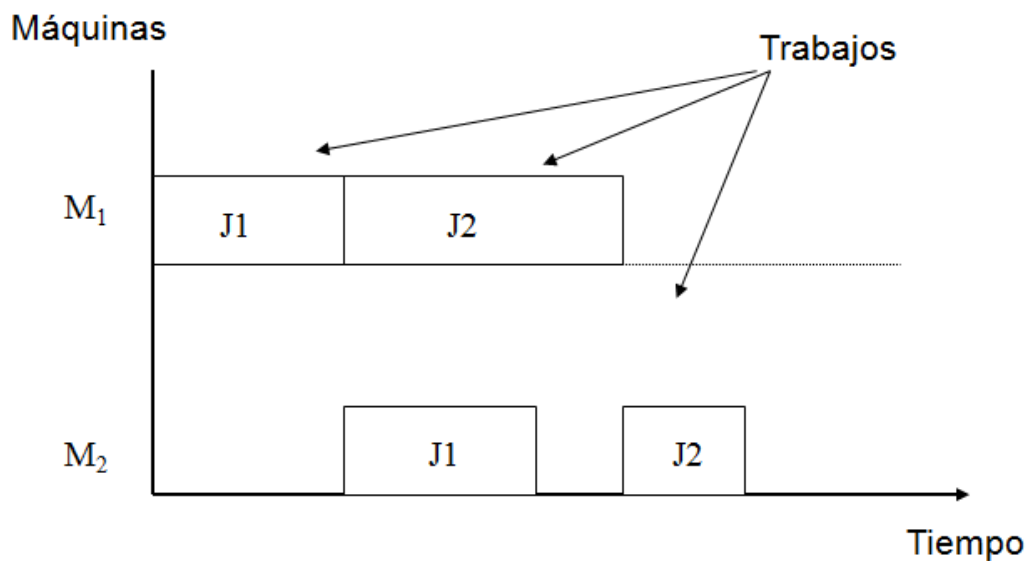


Ilustración 4

Notación: Un problema se describe por una tripleta $\alpha/\beta/\gamma$:

- El campo α indica el entorno de las máquinas (número de máquinas y tipo de arquitectura del sistema).
- El campo β describe las características y restricciones de procesamiento de los trabajos.
- El campo γ especifica el objetivo del problema.

Terminología para los entornos de máquinas (α):

$M = \{M_1, M_2, \dots, M_j, \dots, M_m\}$: Conjunto de m máquinas

F_m : Sistema de flujo uniforme con m máquinas en serie.

J_m : Sistema de taller con m máquinas en serie.

O_m : Sistema de taller abierto con m máquinas en serie.

P_m : m máquinas idénticas en paralelo.

Q_m : m máquinas uniformes en paralelo.

R_m : m máquinas no relacionadas en paralelo.

S_m : Sistema de flujo uniforme con m centros de máquinas en paralelo.

Terminología asociada a los trabajos (β):

DATOS

$J = \{J_1, J_2, \dots, J_i, \dots, J_n\}$: Conjunto de n trabajos;

p_{ij} : Tiempo de proceso del trabajo J_i en la máquina M_j . (p_i para el caso de una sola máquina)

r_i : Instante de llegada de J_i al sistema. Cuando todos los trabajos están disponibles al comienzo del procesamiento, $r_i = 0$.

d_i : Fecha de entrega del trabajo J_i .

w_i : Peso (coste o valor) del trabajo J_i .

VARIABLES

C_i : Tiempo de finalización de J_i . C_{ij} tiempo de J_i en M_j .

$F_i = C_i - r_i$: Tiempo de permanencia del trabajo en el sistema.

$L_i = C_i - d_i$: Mide la desviación respecto a la fecha de entrega, Si $L_i < 0$ (retraso negativo), $|L_i|$ representa las unidades de adelanto.

$T_i = \text{máximo} \{0, L_i\} = \text{máximo} \{0, C_i - d_i\}$: Tardanza de J_i o número de instantes de retraso de J_i .

$E_i = \text{máximo} \{0, d_i - C_i\}$: Número de instantes de adelanto de J_i .

Variables booleanas para el control de los trabajos que se retrasan y adelantan:

$$U_i = \begin{cases} 1 & \text{si } C_i > d_i \\ 0 & \text{en otro caso} \end{cases} \quad V_i = \begin{cases} 1 & \text{si } C_i < d_i \\ 0 & \text{en otro caso} \end{cases}$$

De manera análoga, se pueden definir variables booleanas para el control de los trabajos finalizados sin retrasos ni adelantos:

$$X_i = \begin{cases} 1 & \text{si } C_i = d_i \\ 0 & \text{en otro caso} \end{cases}$$

Un ejemplo de la notación de la terminología asociada a los trabajos en un diagrama de Gantt es el siguiente:

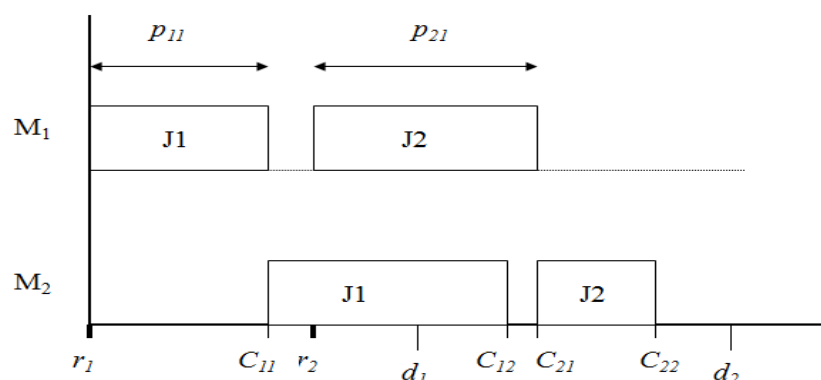


Ilustración 5

En este apartado destacamos determinadas características que pueden presentarse en el procesamiento de los trabajos:

- Preemption (prmt): Rotura de trabajos. Esta característica hace referencia a la posibilidad de abandonar el procesamiento de un trabajo en una máquina sin haber concluido la operación, regresando mas tarde para finalizarla.
- Restricciones de precedencias (pre): En algunos entornos aparecen relaciones de precedencia obligadas entre pares de trabajos.
- No-wait (No-wait): Aparece en entornos de máquinas en serie en los que los trabajos deben ser procesados desde su inicio en la primera máquina, hasta su finalización en la última máquina, sin ninguna interrupción entre máquinas.
- Blocking (Block): Esta característica aparece en sistemas de fabricación en serie en los que no están permitidos los buffers intermedios de un cierto tamaño entre máquinas, puesto que bloquean el funcionamiento de las mismas.
- Tiempo de setup (s_{ik}): Tiempo de cambio de procesamiento entre trabajo i y trabajo k .

Objetivos (γ):

Los objetivos o criterios para la búsqueda de soluciones se pueden agrupar en tres grandes grupos:

1. Criterio basado en tiempos de finalización de los trabajos:

ΣC_i : Minimizar la suma de los tiempos de finalización de los trabajos.

$\Sigma w_i C_i$: Minimizar el coste total asociado a la finalización de los trabajos. El peso w_i se entiende como un coste de espera o un valor añadido al trabajo J_i .

$C_{\max} = \text{Max}\{C_1, \dots, C_n\}$: Minimizar el tiempo de finalización de todos los trabajos, también llamado longitud de la programación (*makespan*).

2. Criterios basados en la fecha de entrega:

ΣL_i : Minimizar la suma de retrasos o retraso total. Equivalente a minimizar el retraso medio. De forma análoga al grupo anterior se estudian $\Sigma w_i L_i$ y L_{\max} .

ΣT_i : Minimizar la tardanza total. De forma análoga se definen $\Sigma w_i T_i$ y T_{\max} .

ΣU_i : Minimizar el número de trabajos retrasados. También se estudia la minimización del coste de los trabajos retrasados, representado por $\Sigma w_i U_i$.

$\Sigma(E_i + T_i)$: Minimizar la suma de las desviaciones de los instantes de finalización de los trabajos respecto a sus fechas de entrega d_i . Cuando se establecen penaltis u_i para los adelantos y v_i para los retrasos de cada trabajo, el objetivo viene a ser $\Sigma(u_i E_i + v_i T_i)$.

$\Sigma(U_i + V_i)$: Minimizar el número de trabajos adelantados y retrasados. Igual que anteriormente, también se plantea el criterio $\Sigma w_i (U_i + V_i)$. Este objetivo es equivalente al de maximizar los trabajos terminados justo en su fecha de entrega (*on time*).

3. Costes basados en costes de inventarios y utilización de máquinas:

ΣI_j : Minimizar el tiempo total en que están desocupadas las máquinas, siendo $I_j = C_{\max} - \Sigma p_{ij}$, y Σp_{ij} la suma de los tiempos de procesado de todos los trabajos sobre la máquina M_j .

$\Sigma v_j I_j$: Minimizar el tiempo ponderado de desocupación de las máquinas, siendo v_j un peso por unidad de operación.

El problema que vamos a estudiar es $P_m/prec/C_{max}$, es decir, es un problema en el que habrá m máquinas en paralelo donde los trabajos que se tienen que realizar deben cumplir unas relaciones de precedencias y el objetivo es minimizar el tiempo de finalización de todas las tareas.

Este tipo de problema es llamado Problema de Optimización, puesto que es un problema donde se intenta optimizar la utilización de una serie de recursos para realizar una determinada actividad.

Existen dos tipos de resolución de un Problema de Optimización:

1. Resolución Exacta: Se busca la mejor opción (denominada óptimo) con respecto al criterio de eficiencia.
2. Resolución aproximada o heurística: Se busca una buena solución con respecto al criterio de eficiencia sin necesidad de que tenga que ser la mejor.

También podemos dividir un Problema de Optimización en dos tipos según su complejidad:

1. \in Clase P: Son problemas No Complejos. Resolverlos de forma exacta no implica mucho tiempo.
2. \in Clase NP: Son problemas Complejos. Obtener el óptimo suele ser costoso en tiempo de proceso siempre que el tamaño del problema sea de una cierta dimensión.

2.3 MODELO Pm/prec/Cmax

Una vez definido nuestro problema vamos a ver su modelo de resolución:

Variables:

C_{\max} = Mayor tiempo de finalización de todas las máquinas encargadas de realizar los trabajos

$$X_{ij'} \begin{cases} 1 & \text{Si el trabajo } i \text{ se realiza después del trabajo } i' \\ 0 & \text{Si el trabajo } i \text{ no se realiza después del trabajo } i' \end{cases}$$

t_i = Tiempo de inicio del trabajo i

C_i = Tiempo de finalización del trabajo i

Datos:

N = Número de trabajos.

M = Número de máquinas.

S = Número de precedencias.

p_i = Tiempo de proceso del trabajo i

L = Constante

Función Objetivo:

Min C_{\max}

Restricciones:

$$\sum_{i'=1}^{N+M} x_{ii'} = 1 \quad \forall i = 1 \dots N$$

$$\sum_{i=1}^N x_{ii'} \leq 1 \quad \forall i' = 1 \dots M + N$$

$$t_i \geq t_{i'} + p_i - L(1 - x_{ii'}) \quad \forall i = 1 \dots N; \forall i' = 1 \dots N$$

$$C_i = t_i + p_i \quad \forall i = 1 \dots M + N$$

$$C_i \leq t_k \quad \forall i = 1 \dots S$$

$$C_{\max} \geq C_i \quad \forall i = 1 \dots N$$

La función objetivo trata de minimizar el tiempo de finalización de todos los trabajos, siendo el tiempo de finalización de un trabajo el tiempo de inicio de ese trabajo más su tiempo de proceso. Para la implementación del modelo se considera que las máquinas son trabajos cuyos tiempos de procesos son iguales a cero.

Pasamos a describir las restricciones:

- Primera restricción: Esta restricción nos garantiza que se realice cada trabajo y que solo se pueda realizar una vez.
- Segunda restricción: Esta restricción nos obliga a realizar cada trabajo en una sola máquina.
- Tercera restricción: Esta restricción indica que el tiempo de inicio de un trabajo tiene que ser superior o igual a la finalización de un trabajo que se realizó anterior a él en esa misma máquina.
- Cuarta restricción: Esta restricción calcula el tiempo de finalización de un trabajo como el tiempo en que se inició dicho trabajo más su tiempo de proceso.

- Quinta restricción: Esta restricción asegura que el tiempo de inicio de un trabajo sea mayor o igual que el tiempo de finalización de un trabajo que le precede.

- Sexta restricción: Esta restricción define C_{max} como el mayor tiempo de finalización de un trabajo.

Una vez explicado el modelo vamos a realizar la implementación en un pequeño ejemplo, el cual debe realizar 4 trabajos, se dispone de 2 máquinas idénticas para realizarlos y existen 4 precedencias, los datos del ejemplo son los siguientes:

Tarea	1	2	3	4
Tiempo	3	4	2	3

Precedencia	1	2	3	4
Antes	1	1	2	3
Después	2	3	4	4

La implementación del modelo será:

Restricción 1

$$x_{1_2} + x_{1_3} + x_{1_4} + x_{1_5} + x_{1_6} = 1;$$

$$x_{2_1} + x_{2_3} + x_{2_4} + x_{2_5} + x_{2_6} = 1;$$

$$x_{3_1} + x_{3_2} + x_{3_4} + x_{3_5} + x_{3_6} = 1;$$

$$x_{4_1} + x_{4_2} + x_{4_3} + x_{4_5} + x_{4_6} = 1;$$

Restricción 2

$$x_{2_1} + x_{3_1} + x_{4_1} \leq 1;$$

$$x_{1_2} + x_{3_2} + x_{4_2} \leq 1;$$

$$x_{1_3} + x_{2_3} + x_{4_3} \leq 1;$$

$$x_{1_4} + x_{2_4} + x_{3_4} + x_{4_4} \leq 1;$$

$$x_{1_5} + x_{2_5} + x_{3_5} + x_{4_5} \leq 1;$$

$$x_{1_6} + x_{2_6} + x_{3_6} + x_{4_6} \leq 1;$$

Restricción 3
$$\begin{aligned}t1 &\geq t2 + 4 - 10000 * (1 - x1_2); \\t1 &\geq t3 + 2 - 10000 * (1 - x1_3); \\t1 &\geq t4 + 3 - 10000 * (1 - x1_4); \\t1 &\geq t5 + 0 - 10000 * (1 - x1_5); \\t1 &\geq t6 + 0 - 10000 * (1 - x1_6); \\t2 &\geq t1 + 3 - 10000 * (1 - x2_1); \\t2 &\geq t3 + 2 - 10000 * (1 - x2_3); \\t2 &\geq t4 + 3 - 10000 * (1 - x2_4); \\t2 &\geq t5 + 0 - 10000 * (1 - x2_5); \\t2 &\geq t6 + 0 - 10000 * (1 - x2_6); \\t3 &\geq t1 + 3 - 10000 * (1 - x3_1); \\t3 &\geq t2 + 4 - 10000 * (1 - x3_2); \\t3 &\geq t4 + 3 - 10000 * (1 - x3_4); \\t3 &\geq t5 + 0 - 10000 * (1 - x3_5); \\t3 &\geq t6 + 0 - 10000 * (1 - x3_6); \\t4 &\geq t1 + 3 - 10000 * (1 - x4_1); \\t4 &\geq t2 + 4 - 10000 * (1 - x4_2); \\t4 &\geq t3 + 2 - 10000 * (1 - x4_3); \\t4 &\geq t5 + 0 - 10000 * (1 - x4_5); \\t4 &\geq t6 + 0 - 10000 * (1 - x4_6); \end{aligned}$$
Restricción 4
$$\begin{aligned}C1 &= t1 + 3; \\C2 &= t2 + 4; \\C3 &= t3 + 2; \\C4 &= t4 + 3; \\C5 &= t5 + 0; \\C6 &= t6 + 0; \end{aligned}$$
Restricción 5
$$\begin{aligned}C1 &\leq t2; \\C1 &\leq t3; \\C2 &\leq t4; \\C3 &\leq t4; \end{aligned}$$

Restricción 6

- $C_{max} \geq C_1;$
- $C_{max} \geq C_2;$
- $C_{max} \geq C_3;$
- $C_{max} \geq C_4;$
- $C_{max} \geq C_5;$
- $C_{max} \geq C_6;$

Donde los números del 1 al 4 de las variables binarias representan los 4 trabajos que se deben realizar y los números 5 y 6 a las dos máquinas idénticas de las cuales se dispone.

2.4 HEURÍSTICAS

Para explicar las cuatro heurísticas que utilizaremos, primero vamos a explicar que es un grafo y los tipos de grafos que hay.

Un grafo es un conjunto de objetos llamados vértices o nodos unidos por enlaces llamados aristas o arcos, que permiten representar relaciones binarias entre elementos de un conjunto.

Tipos de Grafos:

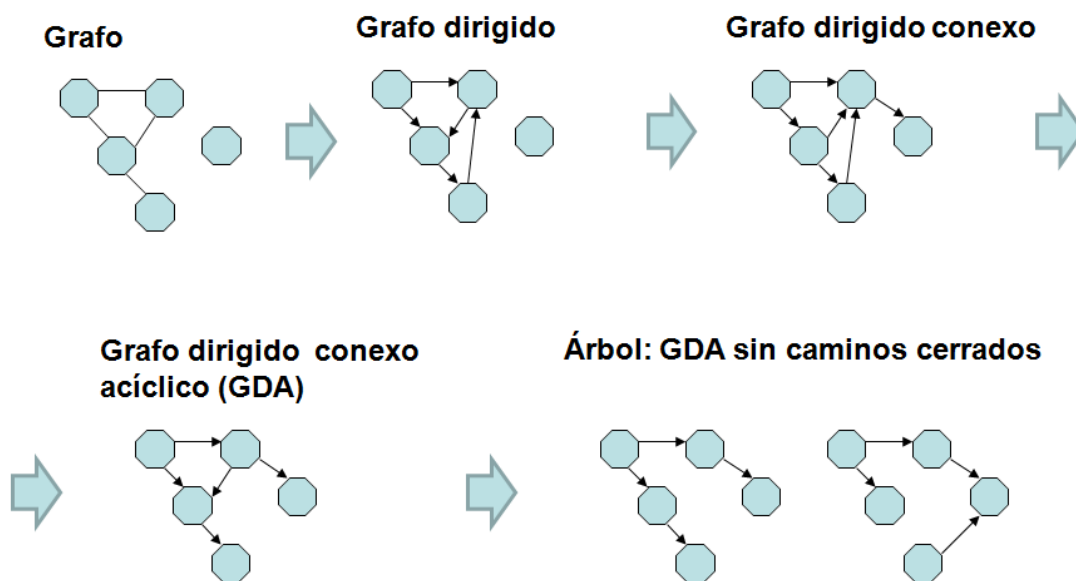


Ilustración 6

Una vez visto los tipos de grafos pasamos a explicar dos tipos de estructuras de grafos, Intree y Outtree:

- **Intree:** Árbol donde cada trabajo tiene a lo sumo un sucesor, el trabajo sin sucesores se encuentra en el nivel 1. Los trabajos que le preceden en el nivel 2. Los trabajos que preceden a los del nivel 2, en el nivel 3, y así sucesivamente.

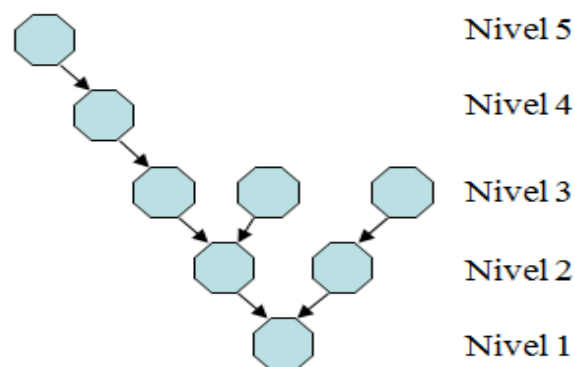


Ilustración 7

- **Outtree:** Árbol donde cada trabajo tiene a lo sumo un predecesor. Los trabajos sin sucesores se encuentran en el nivel 1. Los trabajos que sólo tienen sucesores en el nivel 1 están en el nivel 2. Los trabajos que sólo tienen sucesores en el nivel 1 y 2 están en el nivel 3, y así sucesivamente.

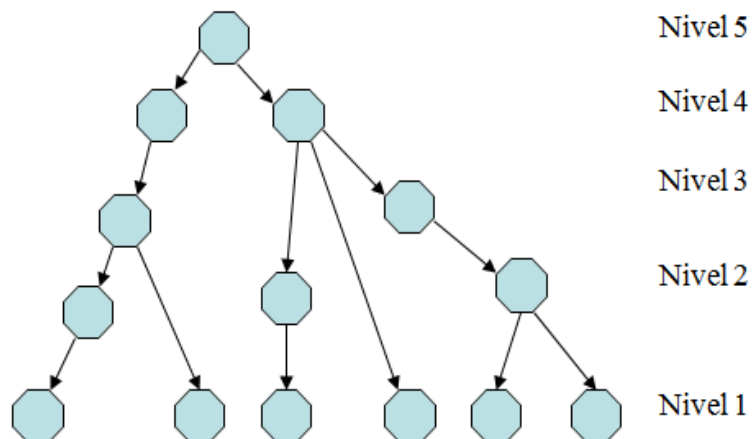


Ilustración 8

Teniendo en cuenta los tipos de estructura de los grafos vamos a explicar las diferentes heurísticas:

Regla CP (CriticalPath): Dar la mayor prioridad a los trabajos con una mayor cadena de trabajos en el grafo de precedencias. Equivalente a dar prioridad a los trabajos de mayor nivel.

Regla CP (CriticalPath), con tiempo de proceso: Dar la mayor prioridad a los trabajos con una mayor cadena de trabajos en el grafo de precedencias. Equivalente a dar prioridad a los trabajos de mayor nivel y también teniendo en cuenta que tiempo de proceso sigue a un trabajo.

Regla LNS (Largestnumber of successors): Dar la mayor prioridad a los trabajos con mayor número de sucesores.

Regla LNS (Largestnumber of successors), con tiempo de proceso: Dar la mayor prioridad a los trabajos con mayor número de sucesores y también teniendo en cuenta que tiempo de proceso sigue a un trabajo.

Una vez explicadas vamos a implementar dichas heurísticas en un breve ejemplo con 6 trabajos y 2 máquinas:

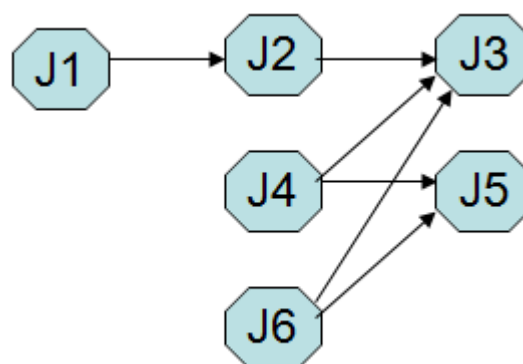


Ilustración 9

Según la regla CP debemos dar prioridad al trabajo que posea una cadena mayor, que en nuestro ejemplo sería el trabajo J1, seguidamente están los trabajos J2, J4 y J6, los cuales poseen el mismo tamaño y por último los trabajos J3 y J5. Por lo que la solución es:

M2	J4	J6	J5		
M1	J1	J2	J3		
0	1	2	3	4	5

Para implementar CP con tiempo de proceso se debe hacer lo mismo que CP pero además del tamaño de la cadena se tendrá en cuenta el tiempo de proceso que le sigue a un trabajo.

Según la regla LNS debemos dar prioridad al trabajo que tenga más sucesores, en nuestro ejemplo serian los trabajos J4 y J6, que tienen dos sucesores y seguidamente van los demás trabajos los cuales poseen un sucesor. Por lo que la solución es:

M2	J6	J5			
M1	J4	J1	J2	J3	
0	1	2	3	4	5

Para implementar LNS con tiempo de proceso se debe hacer lo mismo que LNS pero además del número de sucesores se tendrá en cuenta el tiempo de proceso que le sigue a un trabajo.

2.5 Pm/m>n, prec/Cmax

2.5.1 Introducción

En este apartado se va a comentar un caso particular de nuestro problema en el cual el número de máquinas es mayor o igual que el número de trabajos ($m \geq n$), este tipo de problema es uno de los primeros métodos que origina la gestión de proyectos.

2.5.2 Gestión de Proyectos

La gestión de proyectos se utiliza para planificar todos los trabajos que forman parte de un proyecto, con el propósito de alcanzar uno o varios objetivos.

Las restricciones prioritarias para la gestión de proyectos son el tiempo, la calidad y el coste. El objetivo es optimizar todas las restricciones lo máximo posible. Cumplir este objetivo puede llegar a ser muy complicado debido a que mejorar una de las restricciones suele causar que las otras dos empeoren.

2.5.3 Pm/m>=n, prec/Cmax

Este problema es No Complejo, para resolverlo se utiliza el algoritmo del camino crítico, el cual consta de los siguientes pasos:

1. Crear el grafo de precedencia.
2. Numerar el grafo de precedencia.
3. Calcular E_i = Tiempo más corto en el que podría terminar el trabajo i .
4. $C_{max} = \max\{E_i\}$
5. Calcular el camino crítico. Se calcula un valor L_i que representa el tiempo más tarde en el que podría terminar cada trabajo i
 - 5.1 Para todo J_i sin sucesores:
$$L_i = C_{max}$$
 - 5.2 Para el resto: {En orden decreciente a la numeración}
$$L_i = \min (L_k - p_k)$$

6. El camino crítico es aquel en el que $E_i = L_i$
 $L_i - E_i$ marca la holgura de cada trabajo para su procesamiento
 sin que se produzca un aumento del valor de C_{max} .

Una vez realizado estos pasos construimos un diagrama de Gant:

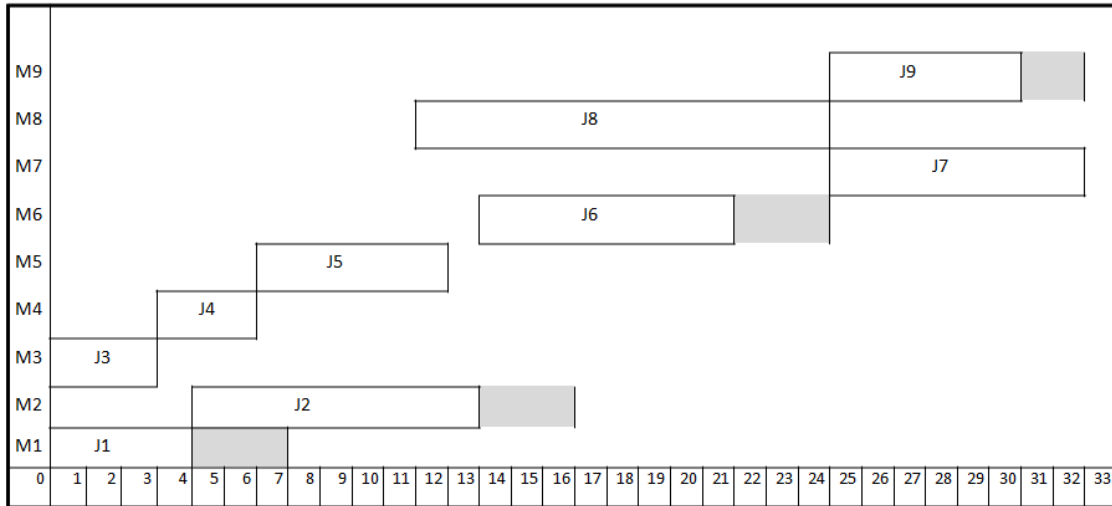


Ilustración 10

Cuando se obtiene el diagrama y sabiendo cual es el camino crítico el objetivo será distribuir todos los trabajos para realizarlo con el menor número de maquinas posible, cumpliendo todos los trabajos sus relaciones de precedencia.

3. IMPLEMENTACIÓN Y RESOLUCIÓN DEL MODELO Pm/prec/Cmax

3. IMPLEMENTACIÓN Y RESOLUCIÓN DEL MODELO

Pm/prec/Cmax

3.1 INTRODUCCIÓN

En este punto describe la implementación del código necesario en Visual Basic para crear el modelo de nuestro proyecto y como posteriormente se ha buscado la solución en LINGO de todos los escenarios de nuestro problema.

3.2 DESCRIPCIÓN DE LA IMPLEMENTACIÓN

Lo primero que tenemos que hacer para solucionar el problema Pm/prec/Cmax es escribir el modelo en Visual Basic. Para poder escribirlo necesitamos los datos del problema que vamos a resolver, en nuestro caso el número de trabajos y precedencias, los tiempos de proceso de cada trabajo y sus relaciones de precedencia, estos datos se recogen en un documento de texto, de extensión “.txt”.

Cuando se genera el código en Visual Basic, el cual recoge los datos y crea el modelo, se ejecuta y se genera un archivo de extensión “.lg4”, este archivo posteriormente se resuelve utilizando la librería de resolución del programa LINGO.

Para obtener la solución de las distintas heurísticas comenzaremos creando un código en Visual Basic que leerá los datos del problema como se hizo anteriormente para resolver el modelo, pero a diferencia de la resolución del modelo en este caso se creará un algoritmo también en Visual Basic, para cada una de las heurísticas, los cuales resuelve el problema y los guarda en un archivo de extensión “.txt”.

Como hemos comentado anteriormente para comprobar las eficiencias de las heurísticas se resuelve el problema

condiferentes características en cuanto al número de trabajos, precedencias y número de máquinas. El código en Visual Basic leerá los datos del problema en unos archivos creados de extensión “.txt” ubicados en una carpeta llamada Batería y posteriormente creará en esa misma carpeta unos archivos de la misma extensión, “.txt”, en la cual nos proporcionará la solución de cada problema según la heurística.

En resumen, la implementación del modelo será

- Datos del proyecto en un archivo de texto (.txt).
- Archivo Visual Basic del modelo Pm/prec/Cmax el cual lee los datos y genera el archivo del modelo matemático del problema (.lg4).
- Archivo LINGO (.lg4) con el que obtenemos la solución del modelo.

Y las implementaciones de las heurísticas:

- Datos del proyecto en archivo de texto (.txt).
- Archivo Visual Basic de las heurísticas el cual lee los datos y realiza el algoritmo que resuelve el problema según la heurística con la cual se está solucionando el problema, una vez se ejecuta el programa se generan la solución.
- Archivo de texto (.txt) el cual contiene la solución de todos los problemas.

3.3 VISUAL BASIC

3.3.1 Introducción

Visual Basic es un ambiente gráfico de desarrollo de aplicaciones para el sistema operativo Microsoft Windows.

Las aplicaciones creadas con Visual Basic están basadas en objetos y son manejadas por eventos, Visual Basic se deriva del lenguaje Basic, el cual es un lenguaje de programación estructurado. Sin embargo, Visual Basic emplea un modelo de programación manejada por eventos.

3.3.2 Estructura

Una vez hablado de la implementación vamos a explicar la estructura que sigue un proyecto en Visual Basic:

1. Declaración de variables: En esta primera parte se declaran las variables necesarias para realizar las funciones y las calcula.
2. Condiciones iniciales: Son condiciones que afectaran a todo el documento.
3. Funciones: Primero tenemos la función principal que es la encargada de llamar a las demás funciones, para que se lean los datos del problema y se realice todas las operaciones y cálculos necesarios.

3.3.3 Funciones del modelo y heurísticas

- El modelo de este proyecto tiene 3 funciones:
 - main**: Esta función es la función principal, es la encargada de llamar a las demás funciones para que se realicen todas las operaciones y cálculos necesarios. Esta función sigue la siguiente estructura:
 1. Se declara todas las variables necesarias para poder llevar a cabo todas operaciones de la función.

2. Con la función Select Case vamos seleccionando uno a uno todos los problemas de la Bateria.

3. Llama a la función LeerArchivo.

4. Llama a la función CrearModelo.

- **LeerArchivo:** Su función es la de encargada de abrir el problema de la batería y leer el archivo, para obtener los tiempos de proceso de cada trabajo y sus relaciones de precedencia. Esta función sigue la siguiente estructura:

1. Se declara todas las variables necesarias para poder llevar a cabo todas operaciones de la función.

2. Se abre el archivo el cual va a leer.

3. Se leen los archivos de la batería de problemas y guarda los datos del problema en las variables globales declaradas.

- **CrearModelo:** Su función es la encargada de crear el modelo que posteriormente se solucionará en LINGO. Esta función sigue la siguiente estructura:

1. Se declara todas las variables necesarias para poder llevar a cabo todas operaciones de la función.

2. Se crea un archivo de extensión .lg4 en el cual se escribirá el modelo.

3. Se escribe el nombre del modelo.

4. Se escribe la función objetivo.

5. Se escribe la restricción que obliga a realizar una única vez cada trabajo.

6. Se escribe la restricción que obliga a que cada trabajo pueda realizarse en una sola máquina.

7. Se escribe la restricción la cual nos asegura que el tiempo de inicio de un trabajo tenga que ser superior o igual a la finalización de un trabajo que se realizo anterior a él en esa misma máquina.
 8. Se escribe la restricción la cual calcula el tiempo de finalización de un trabajo como la suma de su tiempo de inicio mas su tiempo de proceso.
 9. Se escribe la restricción en la cual el tiempo de inicio de trabajo tiene que ser mayor o igual al tiempo de finalización de un trabajo que le precede.
 10. Se escribe la restricción la cual declara Cmax como el mayor tiempo de finalización de un trabajo.
 11. Se declaran las variables binarias del modelo.
- El código de las heurísticas son los cuatro similares, por lo que explicaremos solo una, en este caso la heurística CP, tiene cinco funciones:
 - main**: Esta función es la función principal, es la encargada de llamar a las demás funciones para que se realicen todas las operaciones y cálculos necesarios. Esta función sigue la siguiente estructura:
 1. Se declara todas las variables necesarias para poder llevar a cabo todas operaciones de la función.
 2. Con la función Select Case vamos seleccionando uno a uno todos los problemas de la Batería.
 3. Llama a la función LeerArchivo
 4. Llama a la función CalculoCP.
 5. Llama a la función CalculoCmax.
 - **LeerArchivo**: Su función es la de encargada de abrir el problema de la batería y leer el archivo, para obtener los tiempos de

proceso de cada trabajo y sus relaciones de precedencia. Esta función sigue la siguiente estructura:

1. Se declara todas las variables necesarias para poder llevar a cabo todas operaciones de la función.
2. Se abre el archivo el cual va a leer.
3. Se leen los archivos de la batería de problemas y guarda los datos del problema en las variables globales declaradas.

- **CalculoCP:** Esta función es la encargada de crear el archivo de texto donde se guarda la solución, llamar a la función ReCP y ordenar los trabajos según los criterios de la heurística. La estructura de esta función es la siguiente:

1. Se declara todas las variables necesarias para poder llevar a cabo todas operaciones de la función.
2. Se crea un archivo de extensión .txt en el cual se escribirá la solución.
3. Llama a la función ReCP.
4. Ordena los trabajos de mayor a menor cadena de trabajos.

- **RecCP:** Su función es la encargada de calcular la longitud de la cadena de cada trabajo. Su estructura es la siguiente:

1. Se declara todas las variables necesarias para poder llevar a cabo todas operaciones de la función.
2. Calcula la máxima longitud de la cadena de cada trabajo comprobando sus relaciones de precedencia.

- **CalculoCmax:** Esta última función es la encargada de calcular el trabajo que finaliza último, es decir calcular el Cmax del problema, una vez calculado escribe el resultado en el archivo de texto anteriormente creado por la función CalculoCP. La estructura de esta función es la siguiente:

1. Se declara todas las variables necesarias para poder llevar a cabo todas operaciones de la función.
2. Selecciona el trabajo que debe realizarse según la ordenación calculada anteriormente.
3. Comprueba si el trabajo que debe realizarse tiene algún otro trabajo que le precede, si es así se obliga a que su tiempo de inicio sea al menos igual que el mayor tiempo de finalización de los trabajos que le precedan.
4. Introduce el trabajo en la maquina en la cual el tiempo de finalización sea el menor.
5. Comprueba el tiempo de finalización de todos los trabajos y el mayor lo define como Cmax.
6. Escribe la solución en el archivo .txt anteriormente creado.

3.3.4 Pasos para la ejecución de las funciones

Como comentamos anteriormente, para la creación del modelo y resolución de las heurísticas se va a utilizar Visual Basic.

Al abrir el programa aparece la siguiente pantalla, en la cual se encuentra el menú principal:

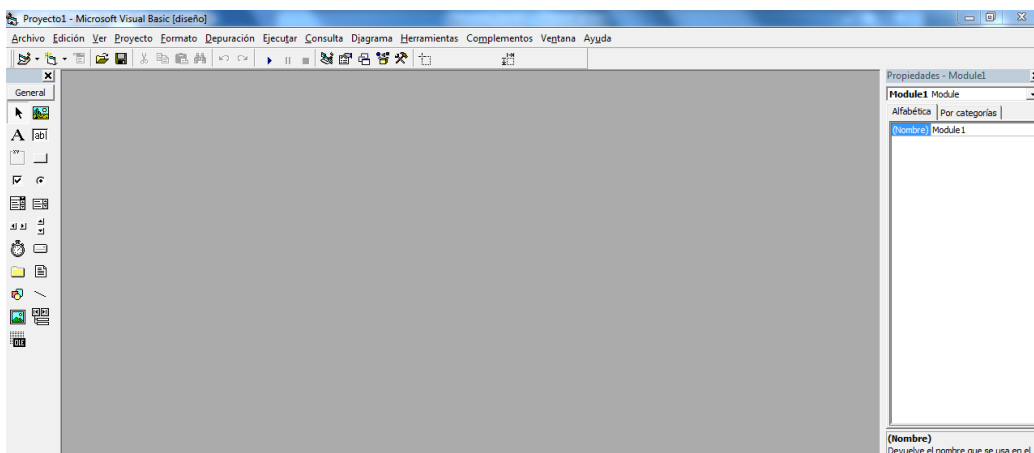


Ilustración 11

Una vez abierto el programa, para abrir el proyecto que crea el modelo o resuelve alguna de las heurísticas, en el menú principal le damos a Archivo, una vez se despliegue el submenú seleccionamos la opción de Abrir proyecto y elegimos el proyecto que queremos en ese momento:

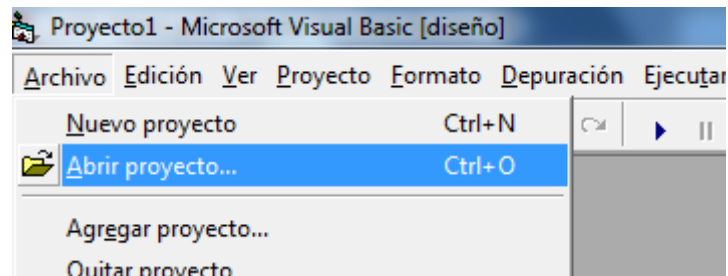


Ilustración 12

Para poder abrir un proyecto se nos abrirá una ventana en la que buscaremos nuestro proyecto según la ruta donde este almacenado:

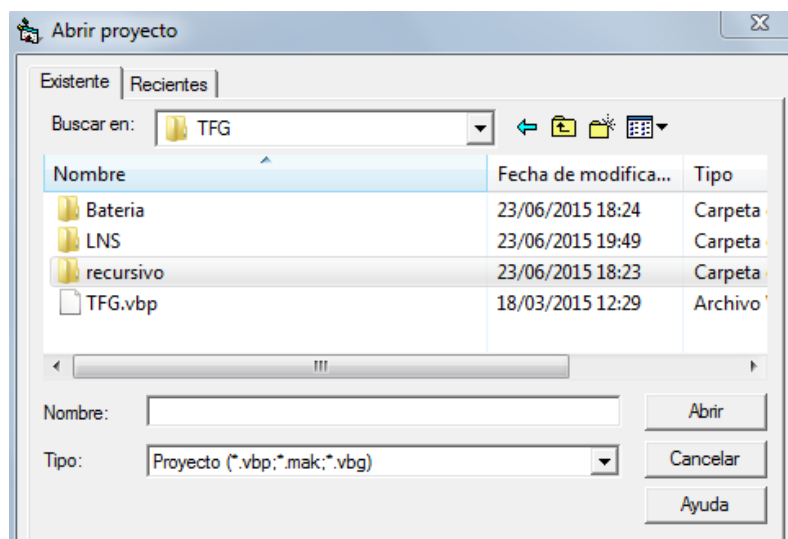


Ilustración 13

Cuando seleccionamos uno de los proyectos y le damos a Abrir aparecerá todo el código del proyecto seleccionado:

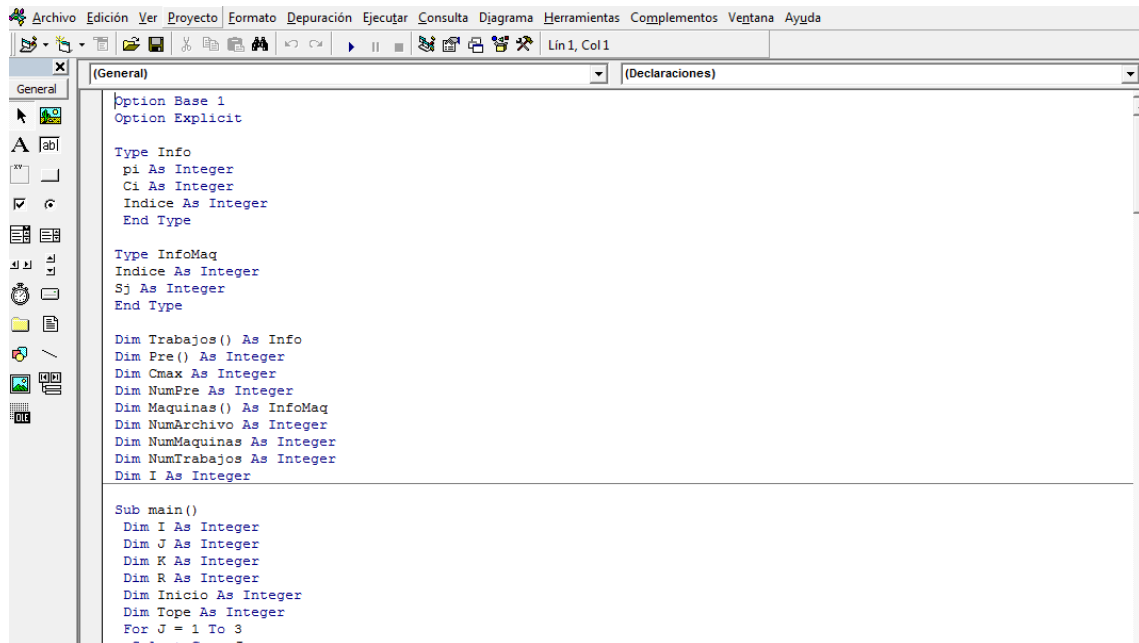


Ilustración 14

Para ejecutar el programa pulsamos el botón Play y comenzará a recorrer todo el código de Visual Basic:

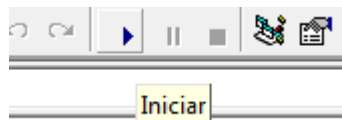


Ilustración 15

Una vez hayamos pulsado el botón Play se generará los modelos en extensión .lg4 o se resolverá alguna de las diferentes heurísticas, según el proyecto que se haya abierto.

3.4 LINGO

3.4.1 Introducción

LINGO: (Linear Generaliza Optimizer) es una herramienta simple para formular problemas lineales y no lineales, resolverlos y analizar su solución. El resultado que LINGO nos proporciona es la optimización que nos ayuda a encontrar el mejor resultado: la ganancia más alta, o el costo más bajo. A menudo estos problemas involucran el uso más eficiente de los recursos. Los problemas de optimización son clasificados a menudo como lineales y no lineales, dependiendo si las relaciones en el problema son con respecto a las variables.

Uno de los rasgos más poderosos de LINGO es su aplicación en el lenguaje de modelos matemáticos. El cual permite expresar un problema de una manera muy similar a la anotación matemática normal pudiendo también expresar una serie entera de restricciones en una declaración compacta. Esto lleva a modelos que son mucho más fáciles de mantener.

Otro aspecto es la sección de los datos, que le permite aislar los datos de la formulación de modelo. De hecho LINGO puede leer datos incluso de una hoja de cálculo separada, base de datos, o archivar de texto. Con datos independientes del modelo, es mucho más fácil de hacer cambios, y hay menos oportunidad de error cuando se realiza el modelo.

La estructura que sigue LINGO es la siguiente:

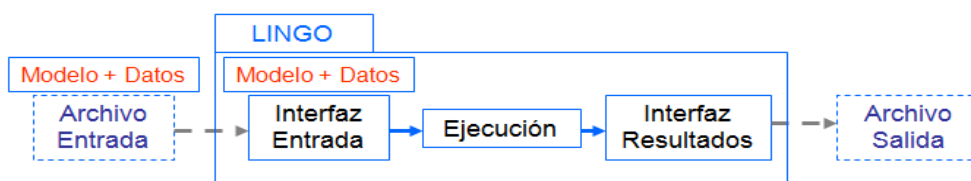


Ilustración 16

En nuestro caso tenemos un archivo de entrada, el cual se crea con anterioridad en Visual Basic, una vez creado se abre con LINGO y aparece el modelo creado con su función objetivo y sus restricciones, al abrirlo lo ejecutamos hasta que LINGO encuentre una solución la cual será nuestro archivo de salida.

El formato del modelo en LINGO es:

1. Título: Descripción del problema
2. Función Objetivo: Maximizar o minimizar con los valores de todas las variables.
3. Restricciones del problema.
4. Restricciones asociadas al tipo de variables.

3.4.2 Sintaxis

La sintaxis que se utiliza en este programa es muy sencilla. Para el nombre de las variables se establece que deben tener 32 caracteres como máximo. Deben comenzar con una letra seguida de letras dígitos o “_”. El compilador de LINGO no distingue entre mayúsculas o minúsculas.

Con respecto a las sentencias:

Todas las sentencias deben terminar en un punto y coma

Para darle un nombre a la función objetivo o las restricciones, esto se debe colocar entre corchetes.

Los comentarios deben comenzar con un signo “!”, los cuales son resultados en verde.

A continuación se habla de la estructura principal de la sintaxis de LINGO:

- **Título:**

No es obligatorio, pero nos puede ayudar a describir el modelo, debe tener como máximo 128 caracteres.

Su formato es:

FORMATO: {TITLE NOMBRE DEL MODELO ;}

Ilustración 17

- **Función Objetivo:**

Para declarar la función objetivo debemos colocar las palabras reservadas MIN o MAX, resaltadas en azul, seguidas del signo =.

Su formato es:

FORMATO: {[NOMBRE]} (MAX/MIN) = x1 + 2 * x2 + 3 * x3 – 5*x4;

Ilustración 18

- **Restricciones:**

LINGO tiene la habilidad de nombrar las restricciones en su modelo. Esta es una práctica buena por dos razones. Primero, los nombres de las restricciones se usan en el reporte de las soluciones que lo hacen más fácil interpretar. Segundo, muchos de los mensajes de error de LINGO se refieren a una restricción dada por el nombre.

Dar nombre a una restricción es bastante simple, se inserta el nombre entre corchetes delante de una línea de código. El nombre debe obedecer los requisitos normales para un nombre de LINGO.

Su estructura es la siguiente:

FORMATO: {[NOMBRE DE LA RESTRICCIÓN]} x1 + x2<=1;

Ilustración 19

- **Variables:**

A menos que especifiquen lo contrario, el valor de las variables por defecto en un modelo de LINGO son no-negativas y continuas. Más específicamente, las variables pueden asumir algún valor real desde cero al infinito positivo. En muchos casos, este dominio de valor por defecto puede ser impropio. Por ejemplo, podemos querer una variable que asuma valores negativos, o se podría querer una variable restringida puramente a valores enteros. LINGO proporciona cuatro funciones de variables dominio que permite sustituir el dominio predefinido de una variable. Los nombres de estas variables y una descripción breve de su uso son:

@GIN restringe una variable para comenzar con valores enteros.

@BIN hace una variable binario (es decir, 0 o 1).

@FREE permite que una variable pueda asumir algún valor real, positivo o negativa.

@BND limita una variable dentro de un rango finito.

3.4.3 Pasos para la resolución del modelo

Una vez generado el código en Visual Basic, el cual lee los problemas de la batería y genera el modelo en formato “.lg4”, se pasa a su resolución en LINGO.

Para comenzar con la resolución del modelo lo primero que hacemos es iniciar el programa LINGO, al abrirlo aparece la siguiente pantalla:

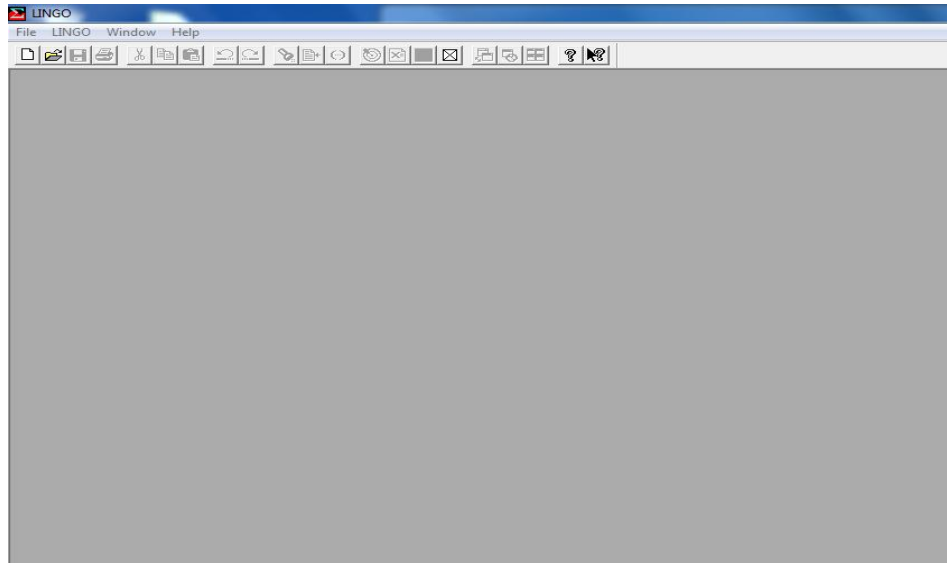


Ilustración 20

Una vez abierto el programa para poder abrir los modelos generados anteriormente se pulsa la pestaña File y posteriormente la subpestaña Open:

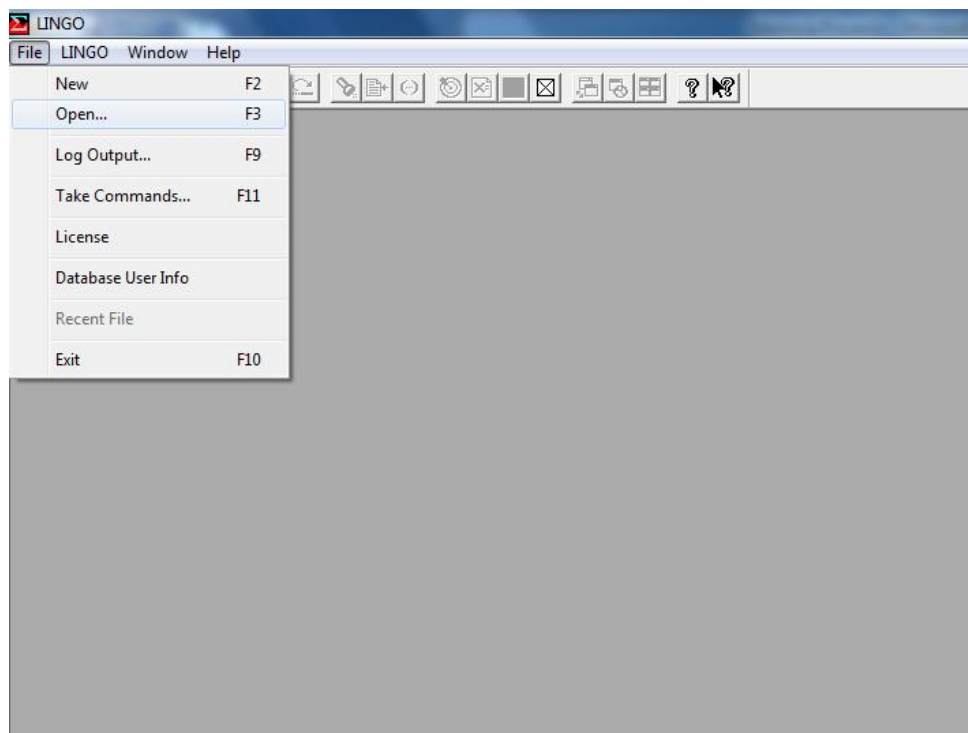


Ilustración 21

Al seleccionar la subpestaña Open debemos buscar la ruta en la cual se encuentran todos los modelos a resolver y elegir el que se quiere resolver:

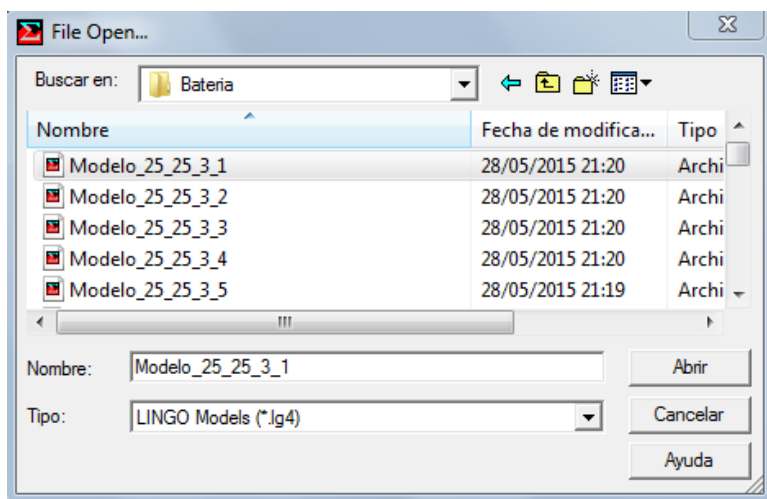


Ilustración 22

Una vez que entramos en la ruta en la cual se encuentran los modelos y seleccionamos el que se desea resolver se pulsa el botón abrir y aparecerá el modelo elegido definido por su título, función objetivo y restricciones que lo componen:

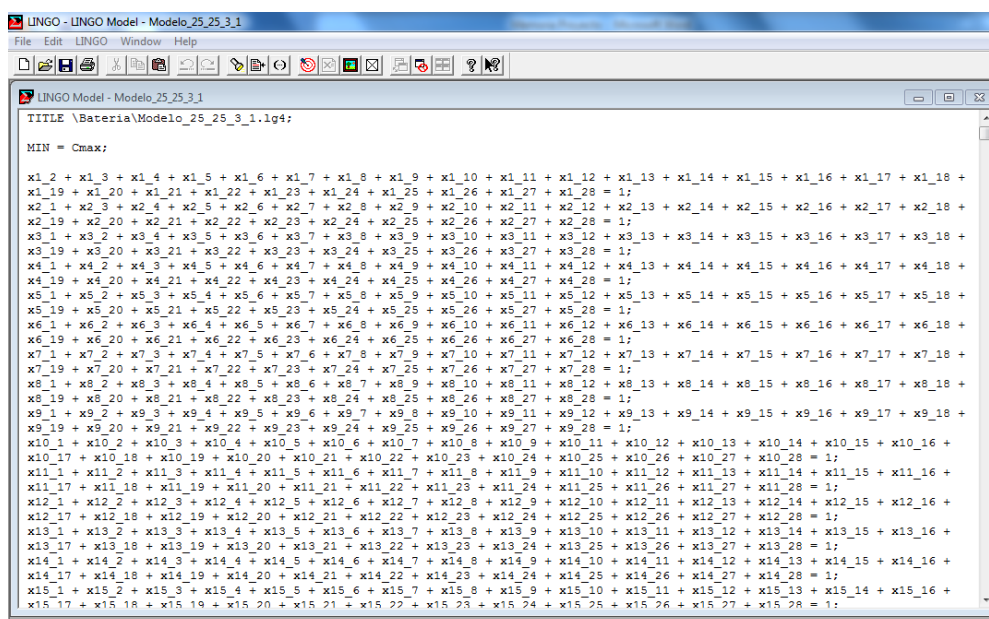


Ilustración 23

Cuando se abre el modelo pasamos a resolverlo, para ello tenemos dos opciones muy fáciles de ejecutar, una de ellas es pulsar la pestaña LINGO y una vez abierta pulsamos la subpestaña Solve y la otra es pulsar el icono con forma de diana que se encuentra en el menú principal:

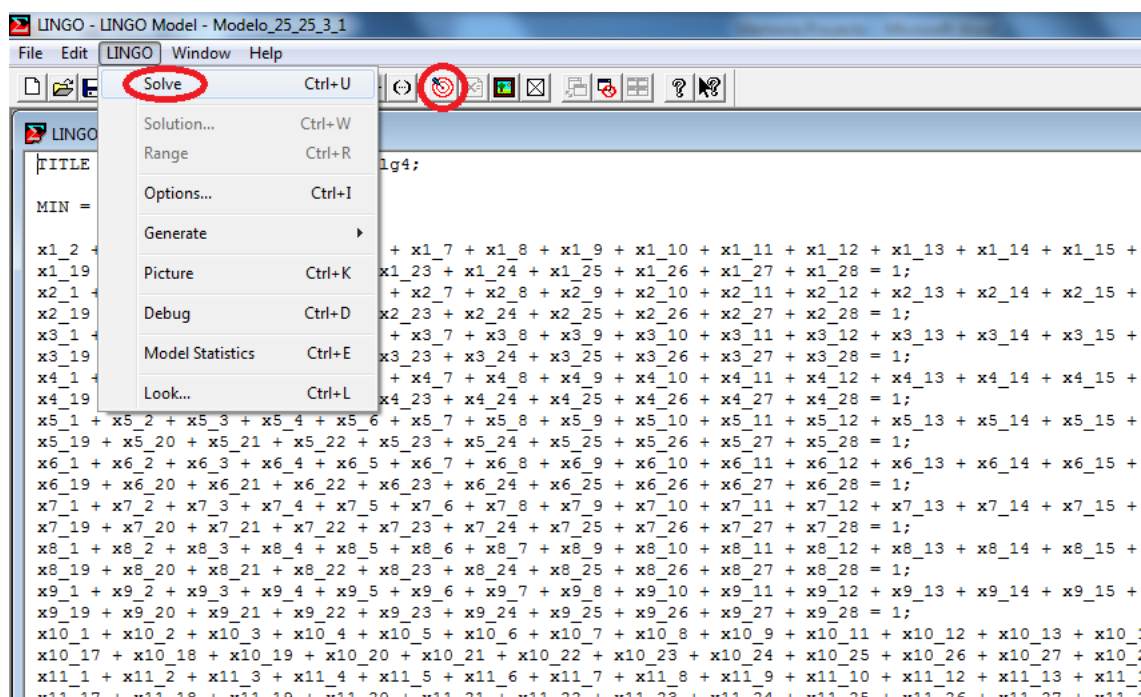


Ilustración 24

Tras pulsar el botón de resolución del modelo, la interfaz de LINGO pasa a buscar el óptimo, una vez que lo encuentra la solución aparece la siguiente pantalla, en la cual se puede ver el óptimo del modelo, el número de variables y restricciones y el tiempo que ha tardado en encontrar la solución obtenida:

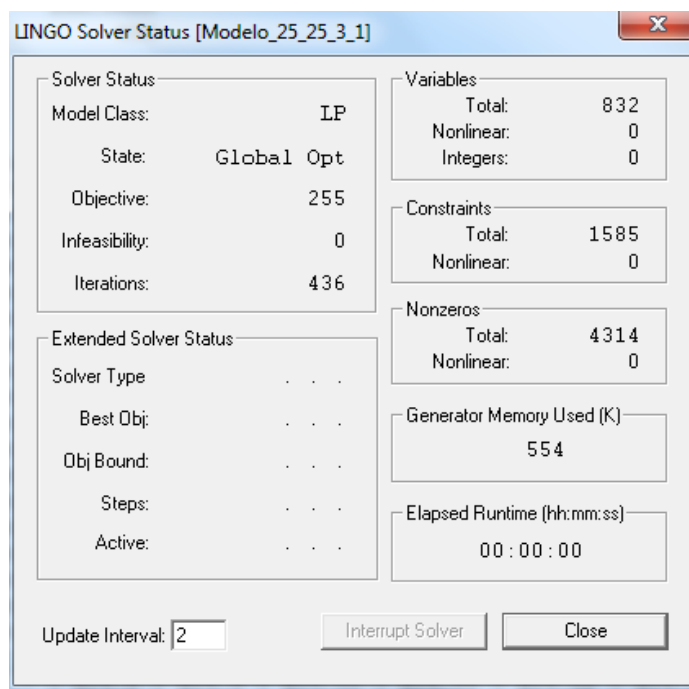


Ilustración 25

Cuando encuentra el óptimo también aparece otra pantalla que nos muestra todas las variables del modelo y el valor de cada una de ellas.

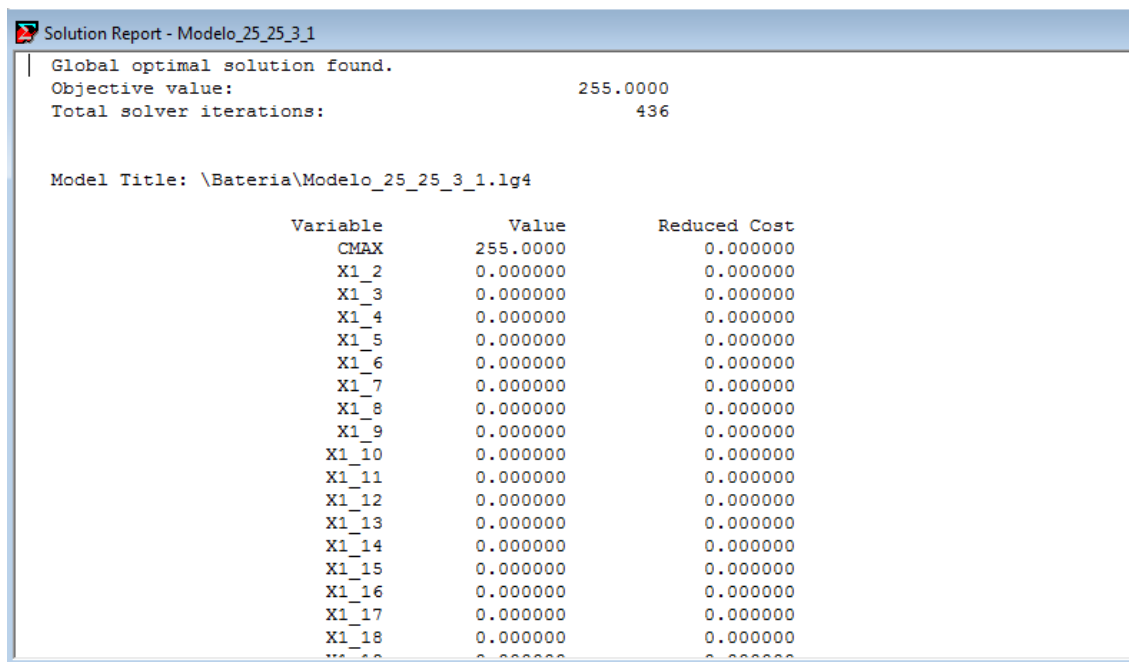


Ilustración 26

4. RESULTADOS DE LA EXPERIMENTACIÓN

4. RESULTADOS DE LA EXPERIMENTACIÓN

4.1 INTRODUCCIÓN

En este punto se habla de la batería del problema que se resuelve para poder analizar las eficiencias de las heurísticas del problema Pm/prec/Cmax. Después se presenta los resultados obtenidos por el modelo LINGO y el de las diferentes heurísticas y al final comentaremos que heurística es la más eficiente.

4.2 BATERÍA DEL PROBLEMA

La batería de problema se utiliza para que una vez resuelta por el modelo y las heurísticas poder compararlas y tomar conclusiones sobre la eficiencia de cada una.

Cada archivo de la batería contiene la siguiente información:

- Número de trabajo.
- Número de precedencias.
- Duración de cada tarea.
- Relaciones de precedencia de las tareas.

En cada problema de la batería se encuentra en su nombre información específica de ese problema. Por ejemplo, "File_25_25_1.txt", cada número del título es una característica de ese problema:

El primer número indica el número de precedencias que tiene ese problema, en nuestro ejemplo 25, los valores que pueden aparecer en la batería son 25, 50 y 100.

El segundo número nos muestra el número de trabajos del problema, en nuestro caso 25. El número de trabajos de la batería se crean de tal manera que debe ser mayor o igual al número de precedencias, por lo que no puede haber un

problema con 100 precedencias y 25 trabajos, los valores que pueden aparecer son 25, 50, 100 y 200.

Por último el tercer número nos indica las diferentes versiones de problemas que poseen las otras dos características anteriores iguales, los valores que pueden tomar en la batería son 1, 2, 3, 4 y 5.

En resumen los tipos de problemas son los siguientes:

Nº PRECEDENCIAS	Nº TRABAJOS	VERSIÓN
25	25	1, 2, 3, 4, 5
25	50	1, 2, 3, 4, 5
25	100	1, 2, 3, 4, 5
25	200	1, 2, 3, 4, 5
50	50	1, 2, 3, 4, 5
50	100	1, 2, 3, 4, 5
50	200	1, 2, 3, 4, 5
100	100	1, 2, 3, 4, 5
100	200	1, 2, 3, 4, 5

En total se trata de 45 problemas que al resolverlos lo utilizaremos para comprobar la eficiencia de cada modelo de resolución.

A continuación vamos a mostrar un ejemplo de un problema, por ejemplo el archivo “.txt” del problema anteriormente mencionado “File_25_25_1.txt”:

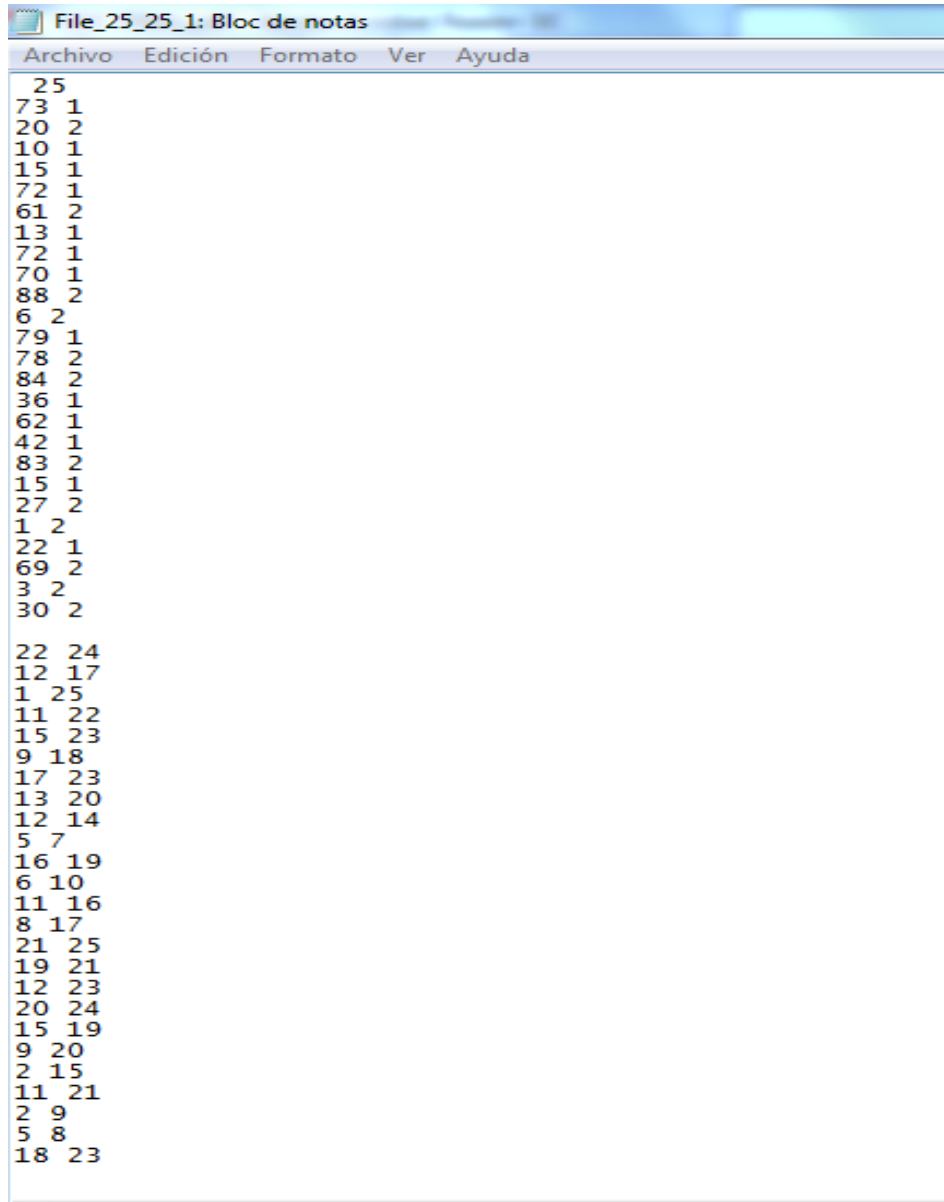


Ilustración 27

4.3 COMPROBACIÓN DEL MODELO Y HEURÍSTICAS

Para comprobar que el modelo funciona correctamente primero se resolverá un ejemplo pequeño con las siguientes características: 4 tareas, 2 máquinas y 4 precedencias.

Datos del ejemplo:

Tarea	1	2	3	4
Tiempo	3	4	2	3

Precedencia	1	2	3	4
Antes	1	1	2	3
Después	2	3	4	4

Una vez leído los datos se pasa a la implementación del modelo del problema $P_m/pre/C_{max}$ y se obtiene la siguiente solución:

Tarea	1	2	3	4
t_i	0	3	5	7
C_i	3	7	7	10
Maq. 1	1	2		4
Maq. 2			3	

La duración óptima del problema es 10.

Ahora vamos a comprobar que las implementaciones de las diferentes heurísticas funcionan correctamente con el mismo ejemplo:

Empezamos por la regla CP, al resolverla nos da la siguiente solución:

Tarea	1	2	3	4
Nivel prioridad	1	2	2	2
t_i	0	3	3	7
C_i	3	7	5	10
Maq. 1	1	2		4
Maq. 2			3	

Podemos observar que el trabajo 2 y 3 tiene el mismo nivel de prioridad debido a que los dos trabajos tienen una cadena de 2 trabajos. La solución que nos proporciona esa heurística es 10.

Al resolver CP teniendo en cuenta los tiempos de proceso obtenemos la siguiente solución:

Tarea	1	2	3	4
Nivel prioridad	1	2	3	4
t_i	0	3	3	7
C_i	3	7	5	10
Maq. 1	1	2		4
Maq. 2			3	

En este caso la solución es igual a la anterior, pero los trabajos 2 y 3 no tienen el mismo nivel de prioridad, porque aunque tienen el mismo número en su cadena el trabajo número dos será prioritario al 3 por su tiempo de proceso.

Pasamos a resolver la heurística LNS obteniendo la siguiente solución:

Tarea	1	2	3	4
Nivel prioridad	1	2	2	2
t_i	0	3	3	7
C_i	3	7	5	10
Maq. 1	1	2		4
Maq. 2			3	

Podemos observar que, al igual que ocurre en CP, el trabajo 2 y 3 tiene el mismo nivel de prioridad, en este caso debido a que los dos trabajos tienen el mismo número de sucesores. La solución que nos proporciona esa heurística es 10.

Al resolver LNS teniendo en cuenta los tiempos de proceso obtenemos la siguiente solución:

Tarea	1	2	3	4
Nivel prioridad	1	2	3	4
t_i	0	3	3	7
C_i	3	7	5	10
Maq. 1	1	2		4
Maq. 2			3	

Se resuelve igual que la heurística LNS, pero en este caso el trabajo 2 tendrá prioridad al trabajo 3 debido a su tiempo de proceso. La solución que ofrece esta heurística es 10.

Observamos que las soluciones que nos ofrecen las heurísticas son igual que el óptimo obtenido al resolver el modelo, esto es debido a que es un problema pequeño que hemos utilizado para comprobar que todo funciona correctamente, una vez empezamos a resolver los escenarios de la batería que ocurra esto será muy difícil debido al aumento del tamaño del problema.

4.4 RESULTADOS OBTENIDOS

Una vez comprobado que el modelo y las heurísticas funcionan correctamente se pasa a resolver el modelo con LINGO para obtener la solución exacta, pero nos damos cuenta que debido al gran tamaño del modelo el programa está durante horas e incluso días buscando el óptimo sin conseguir encontrarlo.

La solución que tomamos para resolver este problema es cambiar las variables binarias por variables menores o iguales a 1, esto consigue que LINGO encuentre una solución pero se aleja bastante del óptimo por lo que no nos servirá para comprobarlo con las heurísticas, así que solo compararemos la eficiencia de las cuatro heurísticas diferentes.

A continuación se realiza la comparación de las heurísticas, para ello se obtiene el error relativo, dicho error se calculará respecto a la heurística con mejor solución de cada escenario.

Los resultados con los cuales se obtiene los errores se encuentran en el Anexo de la Memoria.

Para su representación gráfica calculamos el valor medio de las 5 versiones que poseen los mismos números de trabajos y número de precedencias. Se realizará tres gráficas diferentes según el número de máquinas idénticas que disponemos para realizar las tareas:

3 Máquinas:

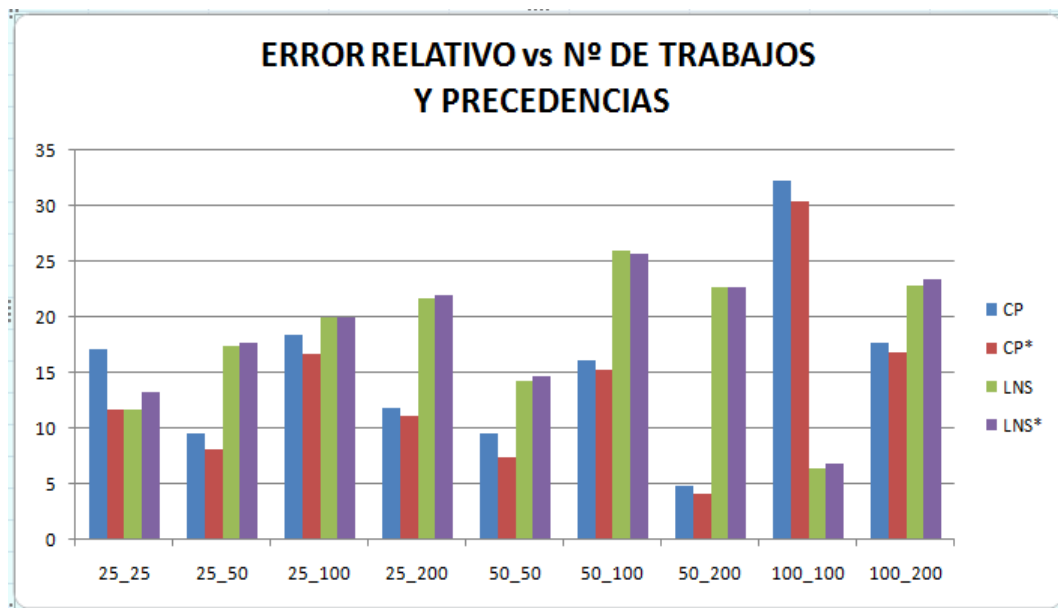


Ilustración 28

Como podemos apreciar en la imagen 28 la heurística que posee menor error relativo cuando disponemos de 3 máquinas para realizar los trabajos es CP teniendo en cuenta los tiempos de proceso, esto ocurre en todos los casos excepto en los escenarios con 100 precedencias y 100 trabajos, en el cual la mejor es LNS.

6 Máquinas:

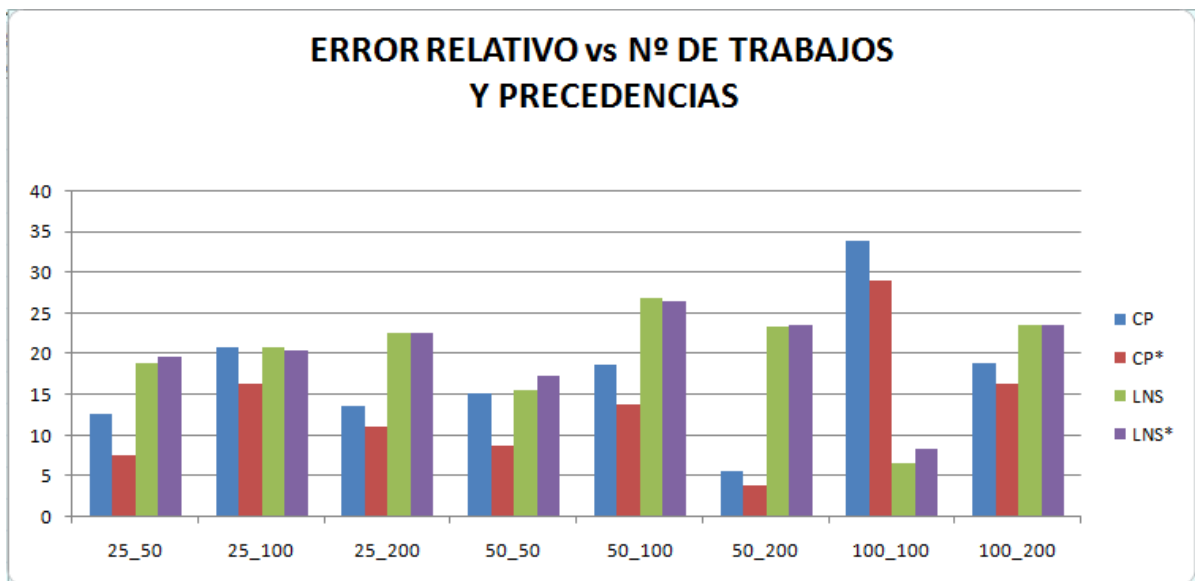


Ilustración 29

10 Máquinas:

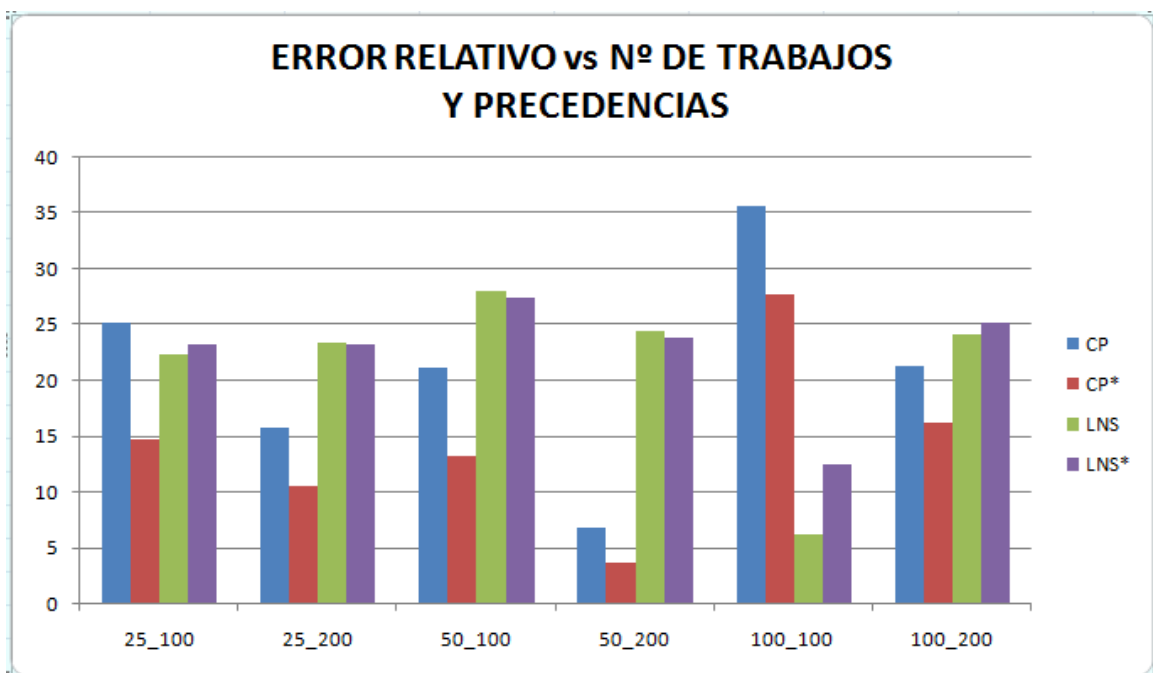


Ilustración 30

Como podemos observar en las imágenes 29 y 30 las heurísticas siguen el mismo comportamiento disponiendo de 6 y 10 máquinas que con 3, es decir, la mejor solución vuelve a proporcionarla CP teniendo en cuenta los tiempos de procesos excepto en un único caso en el cual es mejor LNS.

4.5 CONCLUSIONES

Tras haber realizado la secuenciación de trabajos de máquinas en paralelo y relaciones de precedencias, se concluye que el modelo resuelve nuestro problema estudiado, pero cuando es demasiado extenso al programa le es imposible encontrar el óptimo.

Además al realizar el análisis de la eficiencia de las heurísticas se determina que la más eficiente es CP teniendo en cuenta los tiempos de procesos, esto se observa fácilmente en las gráficas del apartado 4.3.

5. CONCLUSIONES GENERALES

5 CONCLUSIONES GENERALES

El Trabajo Fin de Grado Estudio del modelo matemático y heurísticas de secuenciación de máquinas en paralelo y relaciones de precedencias ha querido afrontar el problema de la gestión de proyecto con el objetivo de obtener la menor duración de un proyecto con restricciones. También se ha introducido diferentes heurísticas que nos proporcionen una buena solución de manera más sencilla.

Una vez resueltos todos los escenarios de la batería los cuales poseen diferentes características como número de precedencias de trabajos y máquinas de nuestro problema, se llega a la conclusión de que en la interfaz LINGO es imposible calcular el óptimo del modelo debido a su gran tamaño y que en casi todos los escenarios de la batería la solución más eficiente nos la proporciona la heurística CP teniendo en cuenta los tiempos de proceso de cada trabajo.

Con esto podemos decir que la implementación en Visual Basic para la creación de un modelo y su posterior resolución en LINGO pueden llegar a ser unas buenas herramientas para obtener el óptimo de muchos modelos, pero también hay veces que es incapaz de resolver modelos debido a su complejidad o extensión.

La realización de este proyecto personalmente me ha servido para aumentar mis conocimientos en la asignatura de Programación de Operaciones de 4ª del Grado en Ingeniería y Tecnologías Industriales, también me ha ayudado para mejorar a la hora de programar en Visual Basic y a saber implementar modelos matemáticos basados en problemas reales con el programa LINGO.

BIBLIOGRAFÍA

A continuación se muestra la bibliografía utilizada para la realización del proyecto:

- Apuntes de la asignatura Programación de Operaciones
Autor: José Manuel García Sánchez
- Investigación Operativa 2002, Software Para Programación Lineal LINGO
Autores: Erica Canizo, Paola Lucero
- Visual Basic
Autor: José Eduardo Maluf de Cervalho
Editorial: McGraw-Hill
- Enciclopedia de Visual Basic
Autor: Francisco Javier Ceballos
Editorial: ra-ma

ANEXO

Resultados del modelo:

- 25 precedencias:

N. PRECEDENCIAS	N. TRABAJOS	N. MÁQUINAS	CASO	TIEMPO
25	25	3	1	255
25	25	3	2	305
25	25	3	3	249
25	25	3	4	294
25	25	3	5	239
25	50	3	1	199
25	50	3	2	124
25	50	3	3	274
25	50	3	4	222
25	50	3	5	164
25	50	6	1	199
25	50	6	2	124
25	50	6	3	274
25	50	6	4	222
25	50	6	5	164
25	100	3	1	165
25	100	3	2	259
25	100	3	3	189
25	100	3	4	252
25	100	3	5	230
25	100	6	1	165
25	100	6	2	259
25	100	6	3	189
25	100	6	4	252
25	100	6	5	230
25	100	10	1	165
25	100	10	2	259
25	100	10	3	189
25	100	10	4	252
25	100	10	5	230
25	200	3	1	248
25	200	3	2	327
25	200	3	3	254
25	200	3	4	277
25	200	3	5	321
25	200	6	1	248
25	200	6	2	327
25	200	6	3	254
25	200	6	4	277

25	200	6	5	321
25	200	10	1	248
25	200	10	2	327
25	200	10	3	254
25	200	10	4	277
25	200	10	5	321

- 50 precedencias:

N. PRECEDENCIAS	N. TRABAJOS	N. MÁQUINAS	CASO	TIEMPO
50	50	3	1	305
50	50	3	2	228
50	50	3	3	294
50	50	3	4	236
50	50	3	5	383
50	50	6	1	305
50	50	6	2	228
50	50	6	3	294
50	50	6	4	236
50	50	6	5	383
50	100	3	1	263
50	100	3	2	247
50	100	3	3	273
50	100	3	4	216
50	100	3	5	307
50	100	6	1	263
50	100	6	2	247
50	100	6	3	273
50	100	6	4	216
50	100	6	5	307
50	100	10	1	263
50	100	10	2	247
50	100	10	3	273
50	100	10	4	216
50	100	10	5	307
50	200	3	1	325
50	200	3	2	297
50	200	3	3	305
50	200	3	4	256
50	200	3	5	387
50	200	6	1	325
50	200	6	2	297

50	200	6	3	305
50	200	6	4	256
50	200	6	5	387
50	200	10	1	325
50	200	10	2	297
50	200	10	3	305
50	200	10	4	256
50	200	10	5	387

- 100 precedencias:

N. PRECEDENCIAS	N. TRABAJOS	N. MÁQUINAS	CASO	TIEMPO
100	100	3	1	248
100	100	3	2	351
100	100	3	3	345
100	100	3	4	326
100	100	3	5	332
100	100	6	1	248
100	100	6	2	351
100	100	6	3	345
100	100	6	4	326
100	100	6	5	332
100	100	10	1	248
100	100	10	2	351
100	100	10	3	345
100	100	10	4	326
100	100	10	5	332
100	200	3	1	356
100	200	3	2	428
100	200	3	3	396
100	200	3	4	432
100	200	3	5	396
100	200	6	1	356
100	200	6	2	428
100	200	6	3	396
100	200	6	4	432
100	200	6	5	396
100	200	10	1	356
100	200	10	2	428
100	200	10	3	396
100	200	10	4	432
100	200	10	5	396

Resultado de las heurísticas:

- 25 precedencias:

PRECEDENCIAS	TRABAJOS	MÁQUINAS	EJEMPLO	CP	CP*	LNS	LNS*
25	25	3	1	401	381	477	503
25	25	3	2	545	510	323	322
25	25	3	3	453	433	483	470
25	25	3	4	491	461	528	539
25	25	3	5	445	445	476	483
25	50	3	1	944	905	906	902
25	50	3	2	807	786	1188	1209
25	50	3	3	814	837	615	626
25	50	3	4	1010	1001	974	960
25	50	3	5	826	806	1080	1066
25	50	6	1	492	464	481	476
25	50	6	2	420	396	607	614
25	50	6	3	428	417	311	328
25	50	6	4	537	515	500	494
25	50	6	5	433	411	561	558
25	100	3	1	1732	1717	1669	1666
25	100	3	2	1562	1551	2106	2087
25	100	3	3	1731	1723	2727	2742
25	100	3	4	1804	1786	1876	1891
25	100	3	5	1598	1547	860	860
25	100	6	1	900	862	855	846
25	100	6	2	794	781	1057	1057
25	100	6	3	896	864	1391	1396
25	100	6	4	930	896	944	949
25	100	6	5	819	783	439	434
25	100	10	1	554	518	530	527
25	100	10	2	501	471	649	649
25	100	10	3	568	519	842	867
25	100	10	4	586	542	592	593
25	100	10	5	530	472	272	272
25	200	3	1	3307	3289	2120	2116
25	200	3	2	3196	3180	3225	3236
25	200	3	3	3392	3371	5214	5234
25	200	3	4	3387	3368	4973	5002
25	200	3	5	3505	3471	3609	3608
25	200	6	1	1680	1645	1060	1066
25	200	6	2	1632	1591	1626	1626

25	200	6	3	1727	1686	2635	2640
25	200	6	4	1724	1688	2521	2514
25	200	6	5	1784	1736	1821	1819
25	200	10	1	1021	988	646	648
25	200	10	2	1010	955	989	984
25	200	10	3	1069	1012	1582	1589
25	200	10	4	1047	1012	1527	1524
25	200	10	5	1101	1043	1104	1099

- 50 Precedencias:

PRECEDENCIAS	TRABAJOS	MÁQUINAS	EJEMPLO	CP	CP*	LNS	LNS*
50	50	3	1	828	817	762	766
50	50	3	2	823	810	804	791
50	50	3	3	873	858	675	688
50	50	3	4	783	783	1055	1073
50	50	3	5	805	763	1028	1020
50	50	6	1	458	443	401	401
50	50	6	2	438	411	402	419
50	50	6	3	472	441	339	352
50	50	6	4	405	409	562	566
50	50	6	5	457	403	561	561
50	100	3	1	1721	1708	1607	1613
50	100	3	2	1697	1671	2323	2321
50	100	3	3	1794	1777	2661	2644
50	100	3	4	1769	1753	2472	2445
50	100	3	5	1743	1739	1025	1027
50	100	6	1	888	856	820	823
50	100	6	2	859	840	1185	1187
50	100	6	3	923	889	1336	1325
50	100	6	4	911	878	1246	1246
50	100	6	5	928	872	534	529
50	100	10	1	553	529	507	508
50	100	10	2	536	509	732	722
50	100	10	3	582	539	824	822
50	100	10	4	557	529	757	752
50	100	10	5	579	526	325	325
50	200	3	1	3456	3446	2854	2856
50	200	3	2	3281	3254	3472	3476
50	200	3	3	3352	3324	5937	5932
50	200	3	4	3433	3398	3952	3965
50	200	3	5	3573	3561	3971	3948

50	200	6	1	1744	1724	1445	1439
50	200	6	2	1661	1634	1756	1755
50	200	6	3	1706	1663	2996	3017
50	200	6	4	1743	1699	1988	2007
50	200	6	5	1792	1781	1987	1985
50	200	10	1	1061	1036	875	875
50	200	10	2	1004	984	1060	1052
50	200	10	3	1041	999	1815	1806
50	200	10	4	1060	1020	1213	1209
50	200	10	5	1097	1070	1212	1203

-100 precedencias:

PRECEDENCIAS	TRABAJOS	MÁQUINAS	EJEMPLO	CP	CP*	LNS	LNS*
100	100	3	1	1651	1609	1117	1125
100	100	3	2	1898	1848	1097	1097
100	100	3	3	1778	1761	1370	1405
100	100	3	4	1599	1590	2084	2083
100	100	3	5	1808	1822	1659	1645
100	100	6	1	853	805	563	586
100	100	6	2	962	950	558	581
100	100	6	3	932	882	701	704
100	100	6	4	830	805	1074	1075
100	100	6	5	927	905	850	848
100	100	10	1	522	489	347	357
100	100	10	2	605	583	357	412
100	100	10	3	594	544	432	451
100	100	10	4	517	485	636	664
100	100	10	5	572	545	503	517
100	200	3	1	3455	3425	2347	2340
100	200	3	2	3318	3301	5329	5348
100	200	3	3	3295	3267	4095	4122
100	200	3	4	3401	3392	4295	4314
100	200	3	5	3371	3339	2431	2461
100	200	6	1	1751	1718	1178	1178
100	200	6	2	1673	1655	2701	2692
100	200	6	3	1669	1636	2066	2076
100	200	6	4	1745	1693	2170	2180
100	200	6	5	1717	1672	1226	1227
100	200	10	1	1082	1032	709	721
100	200	10	2	1034	996	1641	1650
100	200	10	3	1017	989	1261	1264
100	200	10	4	1071	1019	1307	1322
100	200	10	5	1053	1005	741	746

