

Proyecto Fin de Carrera
Ingeniería Industrial

Análisis heurístico de problemas de programación
de trabajos en intervalos

Autor: Julio Torralba Rubio

Tutor: José Manuel García Sánchez

Dpto de Organización Industrial y Gestión de
Empresas I
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla

Sevilla, 2016



Proyecto Fin de Carrera
Ingeniería Industrial

Análisis heurístico de problemas de programación de trabajos en intervalos

Autor:

Julio Torralba Rubio

Tutor:

José Manuel García Sánchez

Dpto de Organización Industrial y Gestión de Empresas I

Escuela Técnica Superior de Ingeniería

Universidad de Sevilla

Sevilla, 2016

Proyecto Fin de Carrera: Análisis heurístico de problemas de programación de trabajos en intervalos

Autor: Julio Torralba Rubio

Tutor: José Manuel García Sánchez

El tribunal nombrado para juzgar el Proyecto arriba indicado, compuesto por los siguientes miembros:

Presidente:

Vocales:

Secretario:

Acuerdan otorgarle la calificación de:

Sevilla, 2016

El Secretario del Tribunal

Índice	i
Índice de Tablas	iii
Índice de Figuras	v
1 Introducción y objetivos del proyecto	1
2 Programación de trabajos con intervalos	3
2.1 Características de los trabajos o actividades	3
2.2 Características de los recursos o máquinas	4
2.3 Restricciones de procesamiento	5
2.4 Tipos de planificación	5
2.4.1 Clasificación	6
2.5 Tipos de problema objetivo	8
3 Batería de problemas	13
3.1 Trabajos	13
3.2 Recursos	14
3.3 Matriz de Compatibilidad L	14
3.4 Parámetros de definición de cada problema	15
4 Estrategias de resolución	19
4.1 Esquema básico de resolución	19
4.1.1 Problema flujo a coste mínimo	20
4.2 Estrategias de priorización	23
4.2.1 Estrategias tipo 1	23
4.2.2 Estrategias tipo 2	27
4.3 Particularidades de heurística de problemas VSP	29
5 Programación en Visual Basic	33
5.1 FSP	33
5.1.1 Función Main()	33
5.1.2 Función leer archivo de entrada	34
5.1.3 Segmentación de trabajos	35
5.1.4 Función generar modelo FCM y resolución	36
5.1.5 Análisis de resultados FCM	37
5.2 Estrategias tipo 1	38
5.3 Estrategias tipo 2	38
5.4 Estrategias de control	38
5.5 VSP	39
6 Resultados computacionales	41
6.1 FSP	41
6.1.1 Observación de tendencias generales en respuesta a cada parámetro	42
6.1.2 Comparación entre estrategias	47
6.2 VSP	52
6.2.1 Observación de tendencias generales en respuesta a cada parámetro en VSP	53
6.2.2 Comparación entre estrategias	57

6.3	<i>Comparacion entre FSP y VSP</i>	61
7	Conclusiones y líneas futuras	63
8	Bibliografía	65
9	Anexos	67
9.1	<i>Código estrategia 2.1 FSP</i>	67
9.2	<i>Código estrategia 2.1 VSP</i>	79
9.3	<i>Código estrategia control 1 FSP</i>	91
9.4	<i>Mapa de distribución error relativo estrategias FSP</i>	105
9.5	<i>Mapa de distribución error relativo estrategias VSP</i>	114

ÍNDICE DE TABLAS

Tabla 1 Promedio error relativo por solape FSP	45
Tabla 2 Promedio error relativo por número de máquinas FSP	45
Tabla 3 Promedio error relativo por número de trabajos FSP	46
Tabla 4 Promedio error relativo por tipo matriz L FSP	47
Tabla 5 Promedio error relativo por distribución de máquinas FSP	47
Tabla 6 Promedio error relativo conjunto FSP	48
Tabla 7 Promedio error relativo por solape VSP	56
Tabla 8 Promedio error relativo por número de máquinas VSP	56
Tabla 9 Promedio error relativo por número de trabajos VSP	56
Tabla 10 Promedio error relativo por tipo matriz L VSP	57
Tabla 11 Promedio error relativo por distribución de máquinas VSP	57
Tabla 12 Promedio error relativo conjunto VSP	58
Tabla 13 Incremento valor función objetivo de FSP a VSP	61
Tabla 14 Incremento error relativo de FSP a VSP	62

ÍNDICE DE FIGURAS

Figura 1 Esquema de clasificación de los tipos de problema	6
Figura 2 Ejemplo FSP Operacional varias clases	9
Figura 3 Ejemplo asignación FSP Operacional varias clases	9
Figura 4 Ejemplo VSP Operacional varias clases	10
Figura 5 Ejemplo asignación VSP Operacional varias clases	11
Figura 6 Esquema de distribución de problemas de la batería utilizada	16
Figura 7 Esquema básico de resolución	20
Figura 8 Ejemplo de problema FSP Operacional de una clase	22
Figura 9 Ejemplo grafo de problema Flujo Coste Mínimo	23
Figura 10 Esquema de estrategias 1.1 y 1.2	25
Figura 11 Esquema de estrategias 1.3 y 1.4	27
Figura 12 Esquema estrategias 2.1 y 2.2	28
Figura 13 Esquema de trabajos ficticios correctamente creados	29
Figura 14 Grafo generado correctamente	30
Figura 15 Esquema de trabajos ficticios erróneamente creados	30
Figura 16 Grafo generado erróneamente	30
Figura 17 Efecto de la superposición	31
Figura 18 Promedio total de error relativo por estrategia FSP	41
Figura 19 Comportamiento ante cada solapamiento FSP	42
Figura 20 Comportamiento ante cada cantidad de máquinas FSP	43
Figura 21 Comportamiento ante cada cantidad de trabajos FSP	43
Figura 22 Comportamiento ante cada matriz de compatibilidad FSP	44
Figura 23 Comportamiento ante cada distribución de máquinas FSP	44
Figura 24 Tiempo medio simulación por estrategia FSP	49
Figura 25 Comportamiento disgregado Matriz L / Distribución máquinas FSP	50
Figura 26 Comportamiento disgregado Número de Maquinas / Número de Trabajos FSP	50
Figura 27 Comportamiento disgregado Solapamiento / Número de Máquinas FSP	51
Figura 28 Promedio total de error relativo por estrategia VSP	53

Figura 29 Comportamiento ante cada solapamiento VSP	53
Figura 30 Comportamiento ante cada cantidad de máquinas VSP	54
Figura 31 Comportamiento ante cada cantidad de trabajos VSP	54
Figura 32 Comportamiento ante cada matriz de compatibilidad VSP	55
Figura 33 Comportamiento ante cada distribución de máquinas VSP	55
Figura 34 Tiempo medio simulación por estrategia VSP	58
Figura 35 Comportamiento disgregado Tipo matriz L / Distribución de Máquinas VSP	59
Figura 36 Comportamiento disgregado Número de máquinas / Número de trabajos VSP	60
Figura 37 Comportamiento disgregado Solapamiento / Tipo de matriz L VSP	60
Figura 38 Comportamiento disgregado Solapamiento / Distribución máquinas VSP	61

1 INTRODUCCIÓN Y OBJETIVOS DEL PROYECTO

En este proyecto se ha realizado un estudio sobre el comportamiento de distintas estrategias heurísticas en la resolución de problemas de programación de trabajos en intervalos, tanto en su versión fija como variable.

Los problemas de programación de trabajos en intervalos tratan la asignación de máquinas a trabajos que deben procesarse dentro de una ventana temporal. Es una variante de los problemas clásicos de scheduling puesto que fuerzan el procesamiento dentro de unos límites temporales.

El análisis del problema se ha realizado desde un punto de vista heurístico, a través de estrategias de resolución que convierten el problema en un conjunto de problemas de flujo.

El estudio del comportamiento de las estrategias heurísticas se ha llevado a cabo mediante una amplia batería de problemas con diferentes parámetros de creación que nos permitirán obtener conclusiones sobre el comportamiento de dichas estrategias en relación a la variación de los parámetros.

El documento está estructurado del siguiente modo; En primer lugar se realiza una introducción en la cual se han mostrados los principales grupos de trabajos a los que nos enfrentamos y los tipos de secuenciación existentes.

A continuación se han explicado las distintas estrategias de resolución que se someterán a experimentación, para ello, se muestran los principales aspectos del código simulado en programación Visual Basic para continuar con los resultados de dicha experimentación.

Más adelante se muestran tanto los resultados para programación fija (FSP) como variable (VSP) y la comparación entre ambas.

Finalmente, se muestran las conclusiones obtenidas del estudio de los resultados conseguidos y se abre la puerta a líneas futuras en las que continuar estudiando.

2 PROGRAMACIÓN DE TRABAJOS CON INTERVALOS

En primer lugar, para facilitar la lectura de este capítulo definimos el tipo de problema al que nos enfrentamos cuando nos referimos a programación de trabajos en intervalos. Los elementos que componen este tipo de problemas son:

- Un conjunto de trabajos o actividades J con características definidas
- Un conjunto de recursos o máquinas M capaces de realizar dichos trabajos
- Restricciones de compatibilidad y temporales

Con estos elementos podremos definir distintos tipos de problemas en función de la clase de restricciones que presente nuestro problema, si las restricciones son abiertas o cerradas, o incluso del objetivo de nuestro problema como veremos más adelante en la clasificación.

Sin embargo, todos los problemas tendrán en común el objetivo de maximizar un determinado comportamiento, asignando en cada instante de tiempo nuestros recursos disponibles a los trabajos de la forma más conveniente.

2.1 Características de los trabajos o actividades

Este tipo de problemas, como ya nombrábamos más arriba puede ser aplicado a distintos sectores y ámbitos de la industria buscando siempre el mayor beneficio para la empresa, no obstante todos los trabajos a los que apliquemos esta metodología deberán cumplir una forma estándar de reflejar sus requisitos.

Dichos requisitos se verán traducidos en los siguientes valores:

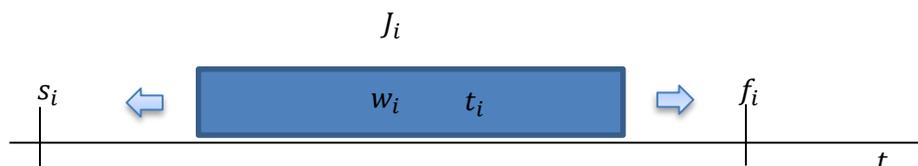
- Inicio del intervalo s_i
- Tiempo del proceso t_i
- Fin del intervalo f_i
- Peso del trabajo w_i
- Clase de trabajo d_i

Siendo el valor s_i el instante en el que comienza la ventana temporal donde será posible realizar el trabajo i , dicha ventana temporal quedará cerrada en el instante f_i , es decir, deberá comenzar después de s_i y finalizar antes de f_i . Dentro de este espacio debemos encontrar el momento de realizar el proceso con duración t_i . Además, para que sea factible dicha realización siempre se debe cumplir que $t_i < f_i - s_i$.

La última variable que nos determinará la factibilidad del trabajo es d_i , esta variable junto con la matriz de compatibilidad que definiremos más adelante servirá para la correcta asignación recurso-actividad.

Por otro lado, una vez que determinamos el trabajo J_i como factible pasamos a definir cómo de rentable es para nosotros realizar dicho trabajo, esta característica queda definida con el peso del trabajo w_i . Esta variable es traducida en el coste de operación de la maquinaria o en el beneficio obtenido por la empresa al realizar el trabajo, en función de si nuestra función objetivo resulte a maximizar beneficios o minimizar costes. Más adelante veremos cómo se comporta este parámetro en nuestra resolución.

Con las características descritas en este punto podemos representar cada trabajo J_i en un diagrama que nos servirá como herramienta visual de la siguiente manera



En cada problema, estas variables deberán quedar definidas para cada uno de los n trabajos del conjunto $J = \{J_1, J_2, \dots, J_i, \dots, J_n\}$

2.2 Características de los recursos o máquinas

En este conjunto de elementos máquina deberán quedar reflejados todos los recursos de los que disponemos para realizar los trabajos o actividades definidos antes en el conjunto J . En este conjunto estamos suponiendo que son elementos ideales que serán capaces de empezar un nuevo proceso justo al terminar el anterior, aunque siempre teniendo en cuenta las restricciones de dicho proceso.

Los recursos de los que disponemos serán reflejados en el problema mediante los siguientes valores en la clase de problema general:

- Clase de la máquina c_j
- Coste de uso de la máquina u_j
- Intervalo de disponibilidad (a_j, b_j)

Significando dichos parámetros las características necesarias para fijar el resultado de nuestro problema. La factibilidad de uso del recurso para realizar una actividad vendrá definida en este caso por el intervalo de disponibilidad y la clase de máquina; debiendo ser solapados los intervalos de disponibilidad de la máquina y de realización del trabajo en un espacio suficiente como para realizar el trabajo como veremos más adelante. La clase de máquina en su caso, junto con la matriz de compatibilidad, nos determinará qué actividades pueden ser realizadas por cada recurso. Esta limitación se traduce en la realidad con limitaciones físicas o por lo contrario intercambiabilidad entre máquinas que puedan realizar varios tipos de actividades.

Finalmente, una vez determinada la posibilidad de realizar el trabajo, la conveniencia de realizar dicho trabajo quedará influenciada por el coste de uso de dicho recurso. La equivalencia la podemos ver como costes de operación de un equipo o salario de un operario para realizar una operación.

En cada problema, estas variables deberán quedar definidas para cada uno de los m recursos del conjunto $M = \{M_1, J_2, \dots, M_j, \dots, M_m\}$

2.3 Restricciones de procesamiento

Teniendo definidos los conjuntos de actividades y recursos el siguiente paso lógico es tratar la existente interrelación entre ellos, que será lo que realmente nos limite a la hora de realizar la asignación y encontrar la manera más conveniente de resolver el problema.

Sin embargo, no solo debemos restringir la interrelación actividad-recursos sino también limitar la simultaneidad de trabajos realizados por una sola máquina. En nuestro problema podrán existir varias máquinas en paralelo de distintas características pero siempre debemos especificar la incapacidad de realizar varios trabajos en el mismo instante de tiempo por una de las máquinas. Si por las características de nuestra planta el elemento máquina que queremos modelar fuera capaz de realizar varias actividades en el mismo instante de tiempo bastaría con añadir una máquina más de dichas características.

La principal restricción que avanzábamos antes, es la restricción de compatibilidad entre los conjuntos actividad-recursos, la cual define la relación existente mediante una matriz de compatibilidad (L). Dicha matriz determina los trabajos que será capaz de realizar una máquina y de igual modo las máquinas que son capaces de realizar un determinado trabajo.

Estas relaciones se pueden representar fácilmente mediante una matriz binaria con el siguiente esquema:

$$\left. \begin{array}{l} \text{D: Número de clases de trabajos} \\ \text{C: Número de clases de máquinas} \end{array} \right\} L_{D \times C} \left\{ \begin{array}{l} 1 \text{ Relación de compatibilidad} \\ 0 \text{ Relación de incompatibilidad} \end{array} \right.$$

Esta matriz de compatibilidad debe cumplir el requisito de ser irreducible, por lo que no puede cumplir ninguna de las condiciones de matriz reducible:

- Contener una fila o columna completa de ceros
- Contener dos filas o columnas idénticas
- Poder ser representada mediante permutaciones de la siguiente forma:

$$L = \begin{pmatrix} L_x & 0 \\ 0 & L_y \end{pmatrix}$$

En el caso de que pudiera ser reducible significaría que una de las clases de trabajos o una de las clases de máquinas es igual a otra.

El tamaño de dicha matriz será dependiente de la variedad de trabajos y máquinas disponibles en la planta, no obstante, la compatibilidad reflejada en esta matriz no debe ser relacionada con el número de máquinas o trabajos de cada clase, ya que este valor es totalmente independiente a la matriz de compatibilidad.

Además, si el modelo lo requiere podemos trabajar con una tercera dimensión de la matriz L en la cual reflejar parámetros como la operación en la que se realiza en el caso de que hubiera varias operaciones.

Con estas restricciones quedan determinadas las relaciones actividades-recursos existentes en el entorno de trabajo.

2.4 Tipos de planificación

A la hora de abordar el problema definido anteriormente existen dos principales objetivos a perseguir según

las que sean nuestras necesidades en el planteamiento.

Según la visión que tengamos del problema, se llevarán a cabo distintos métodos de resolución implicando distintos resultados.

En primer lugar, pasamos a definir la planificación táctica.

Esta planificación tiene como meta principal calcular el número de máquinas que serían necesarias para realizar todas las actividades que estamos proponiendo. Esta cantidad calculada sería siempre imponiendo la minimización de costes de uso de maquinaria que garantice la realización del grupo completo de trabajos. Este tipo de planificación se realiza cuando tratamos las tareas envueltas en el camino crítico de una planificación, es decir, cuando todos los trabajos deben ser completamente realizados al acabar la operación.

Por otro lado, existe la llamada planificación operacional.

La planificación operacional está enfocada en sacar el máximo partido de las máquinas disponibles para la realización de los trabajos de forma que obtengamos los mejores resultados en cuanto a peso total de trabajos realizados. Este tipo de planificación es utilizada a la hora de afrontar una producción cuando no es posible conseguir más medios y no podremos garantizar la realización de todos los trabajos aunque sí la solución más ideal.

2.4.1 Clasificación

A la hora de clasificar los diferentes escenarios con los que nos podemos encontrar lo podemos hacer en base a diferentes criterios, obteniendo con cada uno de ellos distintas propiedades que puede tener nuestro escenario.

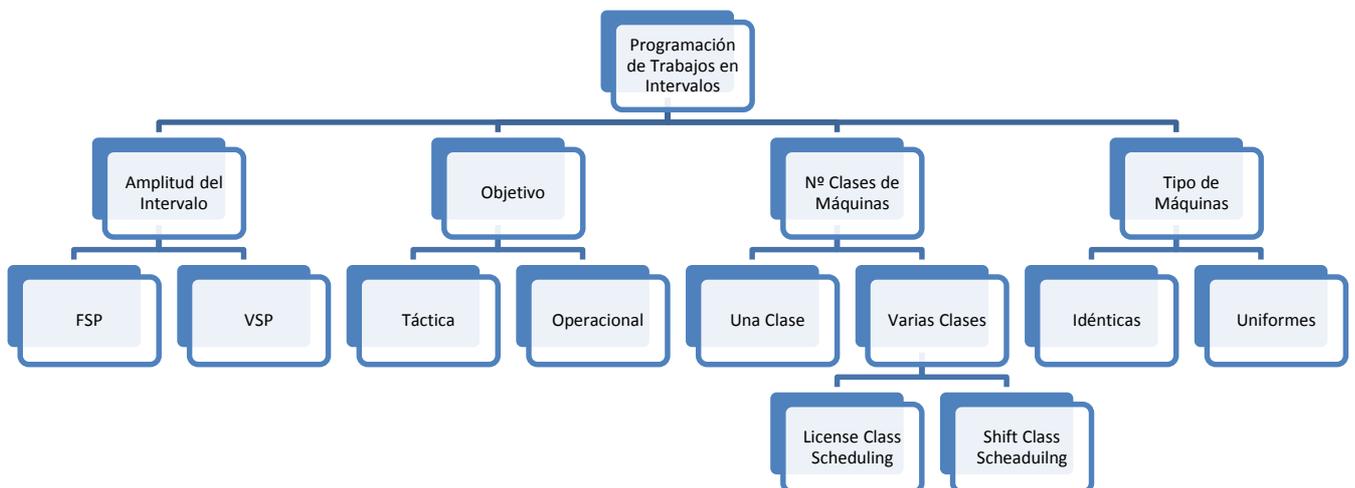


Figura 1 Esquema de clasificación de los tipos de problema

El primero de los criterios y el más lógico es el criterio en base a las restricciones en las que se encierra nuestro

trabajo para ser procesado, por lo que nos referiremos a este criterio como amplitud de procesamiento. Dentro de la amplitud de procesamiento nos encontramos dos grandes grupos de escenarios de trabajo

- **Fixed Scheduling Problem (FSP):** En este tipo de problemas se nos presenta una ventana temporal en la que puede ser realizado en el trabajo que es exactamente la misma a la duración de dicho proceso, por lo que el tiempo para realizarlo es fijo.
- **Variable Scheduling Problem (VSP):** Por lo contrario, los problemas de programación variable nos permiten realizar el proceso dentro de una ventana temporal en la que es admisible su realización. En dicho intervalo, la realización del proceso se puede desplazar permitiendo una mayor flexibilidad y por lo tanto siempre mejores resultados.

Sin embargo, aunque la programación VSP siempre dé como resultado mejores valores de la función objetivo, en la vida real no suele estar en nuestra mano elegir entre FSP o VSP. Este tipo de restricciones suelen venir dadas por otros procesos paralelos de la factoría. Un ejemplo de programación FSP lo podemos encontrar en la asignación de habitaciones en un hotel durante temporada alta, mientras que para programación VSP podría ser la producción de una cadena de montaje a pulsos, en la que todos los trabajos deben ser realizados antes de que el producto pase a la siguiente estación, pero en ese espacio de tiempo se pueden repartir las líneas de trabajo para obtener la planificación óptima.

El segundo de los criterios que distinguimos para clasificar los problemas es el objetivo principal con el que los resolvemos, dentro de este criterio podemos distinguir:

- **Planificación táctica:** En este tipo de planificación el objetivo que perseguimos es la realización de todos los trabajos adjudicando los recursos que sean necesarios para dicho ánimo. Siempre eso sí, de la manera más eficiente posible pero asegurando la plenitud de los trabajos.
- **Planificación operacional:** En este caso, la planificación la realizamos partiendo de la disponibilidad de una serie de recursos con los que el objetivo de nuestro trabajo será realizar las tareas que mejor resultado den a nuestra función objetivo.

Al realizar esta clasificación, en la planificación operacional partimos de la base de que no disponemos de recursos suficientes para acometer el total de las tareas, ya que en ese caso la solución de planificación táctica y operacional nos daría los mismos resultados.

En una situación del mercado, esta clasificación queda reflejada en función del momento en el que nos encontramos. Por norma general, si nos encontramos en el arranque de la producción, se realiza una planificación táctica que nos dará como resultado una cuantificación de los recursos necesarios para la completa acometida de la producción. Sin embargo, si no es posible conseguir todos los recursos necesarios o vamos a tener que acoplar la producción a lo que tenemos a nuestra disposición, el objetivo deberá ser operacional.

El siguiente criterio de clasificación de nuestro entorno es el número de clases de recursos que vamos a utilizar

- **Una clase:** Todas máquinas son iguales y por lo tanto todas pueden hacer los mismos tipos de trabajos.
- **Varias clases:** A la hora de asignar los recursos a los trabajos no todos tienen la misma aplicabilidad por lo que hay que tener en cuenta cómo realizamos la asignación.

En este criterio entra en juego la matriz L de compatibilidad. En esta matriz deberá quedar definida la relación entre recursos y tareas de forma inequívoca, es decir, cada tipo de recurso podrá realizar ciertos tipos de

trabajos. De esta forma, el entorno en el que todos los trabajos pueden ser realizados por todas las máquinas se podría entender como un caso puntual dentro del global de las combinaciones recursos-máquinas y podría quedar reflejado en una matriz de compatibilidad que determine dicha relación de intercambiabilidad.

Por último, la última clasificación que vamos a hacer para enmarcar este proyecto va a ser dentro del ambiente marcado por recursos de varias clases haciendo la diferenciación entre:

- License Class Scheduling: El cual se caracteriza por mostrar disponibilidad de los recursos abierta durante todo el horizonte de trabajo, en el que la compatibilidad está definida únicamente por aspectos técnicos.
- Shift Class Scheduling: Donde cada recurso solamente está disponible en un rango determinado de tiempo, la compatibilidad entre máquina y trabajo está afectada por la disponibilidad de la máquina.

Nuevamente esta determinación será fijada por el entorno al que nos enfrentemos y aunque siempre exista la posibilidad de mejorar nuestra capacidad de recursos solemos encontrarnos la limitación impuesta por la configuración ya predispuesta. También se puede enfocar una planificación de LCS para nuestro proceso formado por un conjunto de tareas esenciales y acoplar el resto de procesos suplementarios a una planificación SCS en los rangos temporales que permite el proceso crítico.

2.5 Tipos de problema objetivo

Después de mostrar en este capítulo tanto los elementos clave que marcan el tipo de problema al que nos enfrentamos como la clasificación que podemos hacer del marco de problemas, debemos definir y profundizar un poco más en los problemas en los que se centra este proyecto.

Los tipos de problemas en los que nos hemos centrado, tanto por su complejidad como por su mayor parecido con situaciones reales han sido los problemas FSP y VSP, ambos en su versión operacional de varias clases. Esto quiere decir que son problemas en los que partimos de una situación inicial en la que disponemos de un cierto número de máquinas de varias clases, las cuales mediante una matriz de compatibilidad están relacionadas con los trabajos a los que nos enfrentamos. En los problemas FSP además estará fijada la ventana temporal en la que podemos realizar el trabajo, que será igual a la duración del mismo; mientras que en los problemas VSP esta ventana temporal será mayor por lo que podremos desplazar el trabajo.

Pasamos a explicarlos en mayor profundidad.

En primer lugar, mostraremos el modelo teórico del problema FSP de varias clases, ya que aunque el principal objetivo del proyecto es el análisis heurístico, para la obtención de los óptimos con los que comparar el modelo heurístico se necesita este modelo.

$$\begin{aligned}
 & \text{Max} \sum_{i=1}^n w_i x_i \\
 & \sum_{c/L(d_i,c)=1} x_{ic} = x_i \quad \forall i \\
 & \sum_{i/L(d_i,c)=1 \& s_i \leq t \leq f_k} x_{ic} \leq m_c \quad c = 1 \dots C; t = s_k \forall J_k \in J/L(d_k, c) = 1
 \end{aligned}$$

En este modelo x_i indicará si el trabajo se procesa y x_{ic} indicará además que clase de máquina ha realizado el trabajo. En un gráfico podríamos ver un ejemplo simple de la siguiente manera.

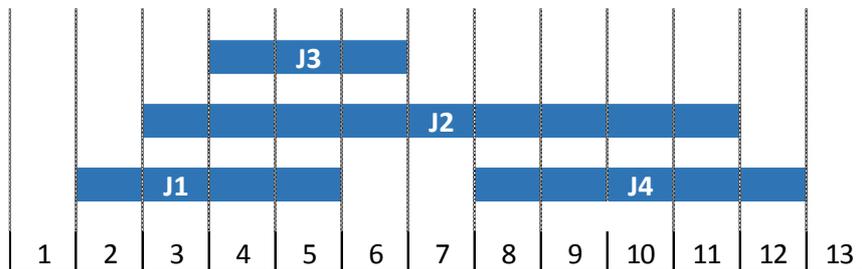


Figura 2 Ejemplo FSP Operacional varias clases

L	1	2
1	1	0
2	1	1
3	0	1

J_i	d_i	Maq. Comp	w_i
1	1	1	7
2	2	1,2	8
3	3	2	4
4	1	1	3

Teniendo una máquina de cada clase, esta distribución de trabajos en el tiempo y la matriz de compatibilidad podemos ver fácilmente cómo la asignación más adecuada sería la siguiente.

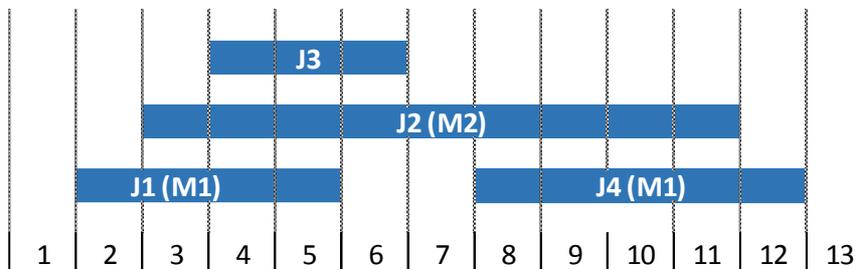


Figura 3 Ejemplo asignación FSP Operacional varias clases

Este simple ejemplo nos permite ver en qué consiste nuestro problema y nos permitirá también remarcar la diferencia de complejidad entre los problemas FSP y VSP ambos operacionales de varias clases.

Para ello mostramos aquí la misma información para el problema VSP.

El modelo matemático de esta clase de problemas es el siguiente.

$$\begin{aligned}
 & \text{Max} \sum_{i=1}^n w_i x_i \\
 & \sum_{c/L(d_{i,c})=1} \sum_{k=a_i}^{b_i-t_i+1} x_{ikc} = x_i \quad \forall i
 \end{aligned}$$

$$\sum_{i/L(d_i,c)=1, k \in [a_i, b_i] / k \leq t \leq k+t_i-1} x_{ikc} \leq m_c \quad c = 1 \dots C, t \in T / t = a_r \dots b_r - t_i + 1 \forall J_r \in J / L(d_r, c) = 1$$

El significado de las incognitas es el mismo aunque añadiendo en este caso la variable k que nos indicará el instante en el que se realiza el trabajo en cuestión.

Como podemos observar, el modelo es mucho más complejo que el FSP, lo que nos acarreará problemas en la resolución como comprobaremos más adelante.

Finalmente añadimos un ejemplo de este tipo de problema para dar una visión más completa de los campos que pretendemos abordar.

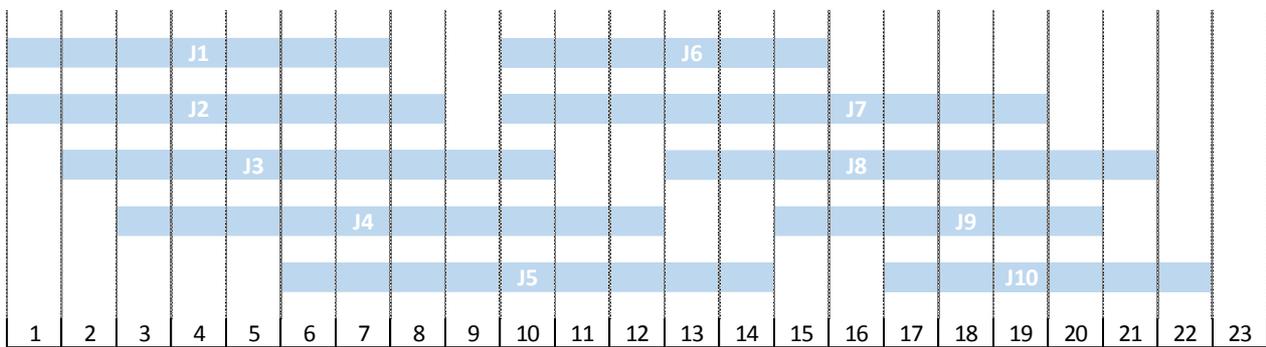


Figura 4 Ejemplo VSP Operacional varias clases

L	1	2
1	1	0
2	1	1
3	0	1

J_i	a_i	b_i	t_i	w_i	d_i
1	0	7	4	1	1
2	0	8	4	2	2
3	1	10	7	3	1
4	2	12	9	4	1
5	5	14	7	3	2
6	9	16	6	3	2
7	9	19	5	2	1
8	12	20	6	2	2
9	14	19	3	4	1
10	14	21	5	4	1

De igual manera, teniendo una máquina por cada clase y con los datos dados podemos llegar a una programación como la vemos en la figura siguiente.

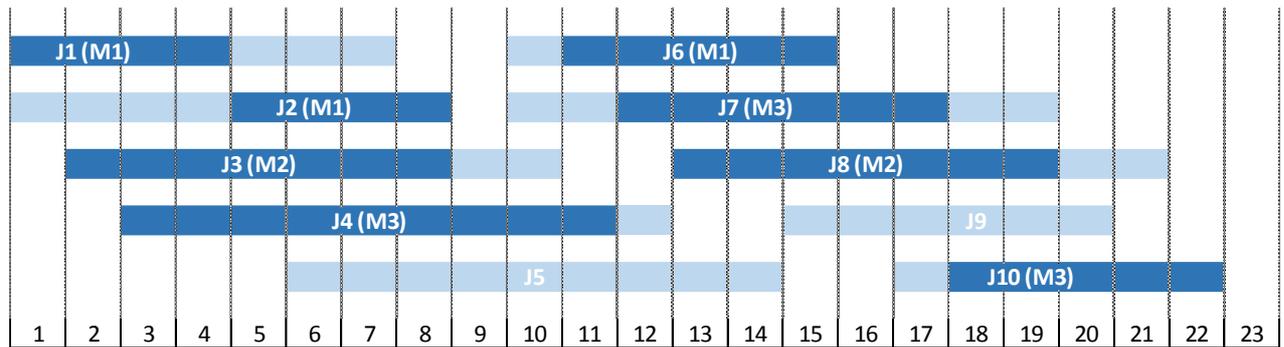


Figura 5 Ejemplo asignación VSP Operacional varias clases

En esta figura podemos observar cómo gracias a la ventana temporal que nos permite el desplazamiento de los trabajos, se pueden llevar a cabo gran cantidad de ellos obteniendo un mayor valor de nuestra función objetivo. Este efecto de incremento lo veremos también en los resultados del proyecto más adelante.

3 BATERÍA DE PROBLEMAS

Cada problema de la batería que vamos a estudiar podría representar por ejemplo una situación de asignación de turnos de trabajo o un flujo logístico en la cual como hemos descrito anteriormente, el horizonte temporal durante el cual puedo realizar los trabajos es exactamente igual al tiempo que tardo en realizar dichos trabajos. Existen distintas máquinas que son capaces de realizar sólo algunos de los trabajos según la matriz de compatibilidad aplicable a cada problema; los recursos que tenemos son limitados en número aunque estarán a nuestra disposición durante todo el horizonte temporal. Finalmente el objetivo es completar los trabajos que nos den un mayor beneficio determinado por el peso de cada uno de los realizados.

Esta situación de trabajo se podría dar por ejemplo a la hora de organizar la producción de una fábrica de coches donde los recursos que tratamos son operarios capaces de realizar varias tareas. Ya tenemos contratados los recursos actuales y queremos dejar realizadas las operaciones de mayor carga de trabajo.

Toda esta planificación la haremos también contando con la premisa de que los recursos de los que disponemos no son capaces de realizar varias tareas simultáneamente.

Al ser este un estudio de comportamiento de varias estrategias heurísticas, hemos compilado una batería de problemas sobre la cual trabajar para obtener conclusiones de cuál será la que mejor resultado nos consigue para aplicarla a nuestro problema.

La batería de problemas desarrollada consta de un total de 4320 problemas en los que variamos distintos parámetros representativos de cada problema, y que nos permitirán estudiar el comportamiento de las estrategias de resolución en función de cada uno de ellos. Todo esto además del estudio de los resultados de la función objetivo.

Para mayor representatividad, dentro de cada una de las combinaciones de parámetros existe un grupo de 10 problemas que hacen que los resultados del estudio sean siempre más significativos al ser la muestra mayor.

Cada problema de esta batería estará determinado por varios parámetros que determinarán las características del problema.

A su vez cada problema estará compuesto de la información que se detalla en los siguientes puntos

3.1 Trabajos

Cada uno de los n trabajos de cada problema contiene los datos necesarios para definirlo:

- s_i : Instante de comienzo
- f_i : Instante de finalización
- t_i : Tiempo necesario para su realización

- w_i : Peso del trabajo
- d_i : Clase de trabajo

Leyendo esta información del archivo de cada problema tendremos totalmente definidos los trabajos a realizar.

Según la estrategia de resolución que se le aplicara cobrarían mayor importancia unos datos u otros. Por ejemplo, si buscamos una aproximación táctica el peso no sería importante ya que tendríamos que realizar todos los trabajos, pero si fuera un objetivo FSP se debe cumplir que la ventana temporal durante la cual podemos realizar el trabajo sea de la misma duración que el tiempo que tardamos en realizarlo, etc...

Esta información de trabajos nos sirve para todas las clases que hemos definido anteriormente en la planificación.

3.2 Recursos

Respecto a los recursos, lo primero que debemos saber es si el problema que estamos afrontando es de una o varias clases. En nuestra batería de problemas en concreto, todos los problemas que encaramos son de varias clases, es por eso, que la información que recibimos de cada uno de los problemas es:

- Número de clases de recursos (C)
- Cantidad total de recursos
- Grado de distribución de los recursos
- Cantidad de recursos de cada tipo

Con esta información quedan totalmente definidos el conjunto de recursos para abordar cualquiera de los tipos de problemas que hemos definido previamente, siempre de varias clases. El número de clases de recursos como veremos en el siguiente punto nos influirá en la matriz de compatibilidad. El grado de distribución de recursos junto con la cantidad de recursos total nos permitirá estudiar cómo se comporta ante cada problema nuestro método de resolución. Finalmente la cantidad de recursos de cada tipo es la que nos permite solucionar exactamente el problema al que nos enfrentamos como veremos más adelante.

3.3 Matriz de Compatibilidad L

Como ya avanzábamos anteriormente, siempre que se trate de un problema de varias clases debemos añadir a nuestro problema la correspondiente matriz de compatibilidad entre recursos y tareas.

Esta matriz requiere un número de columnas igual a la cantidad de clases de máquinas y un número de filas igual a la cantidad de tipos de trabajos:

$$\begin{array}{l}
 \text{D: Número de clases de trabajos} \\
 \text{C: Número de clases de máquinas}
 \end{array}
 \left. \vphantom{\begin{array}{l} \text{D: Número de clases de trabajos} \\ \text{C: Número de clases de máquinas} \end{array}} \right\} L_{D \times C} \left\{ \begin{array}{l} 1 \text{ Relación de compatibilidad} \\ 0 \text{ Relación de incompatibilidad} \end{array} \right.$$

Como ya hemos dicho, cada uno de los trabajos de la batería incluye su propia matriz de **L** pero son de 3 tipos:

$$L_1 = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix}$$

$$L_2 = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix}$$

$$L_3 = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{pmatrix}$$

Como podemos observar fácilmente, estas matrices cumplen el requisito que indicamos en la introducción de no ser matrices reducibles.

3.4 Parámetros de definición de cada problema

Para facilitar el efecto de las características del problema hemos definido los siguientes parámetros que nos permitirán obtener conclusiones.

Cada problema se identifica por una combinación de:

- Solapamiento
- Número de trabajos
- Tipo de matriz de compatibilidad
- Cantidad de recursos
- Grado de distribución de los recursos

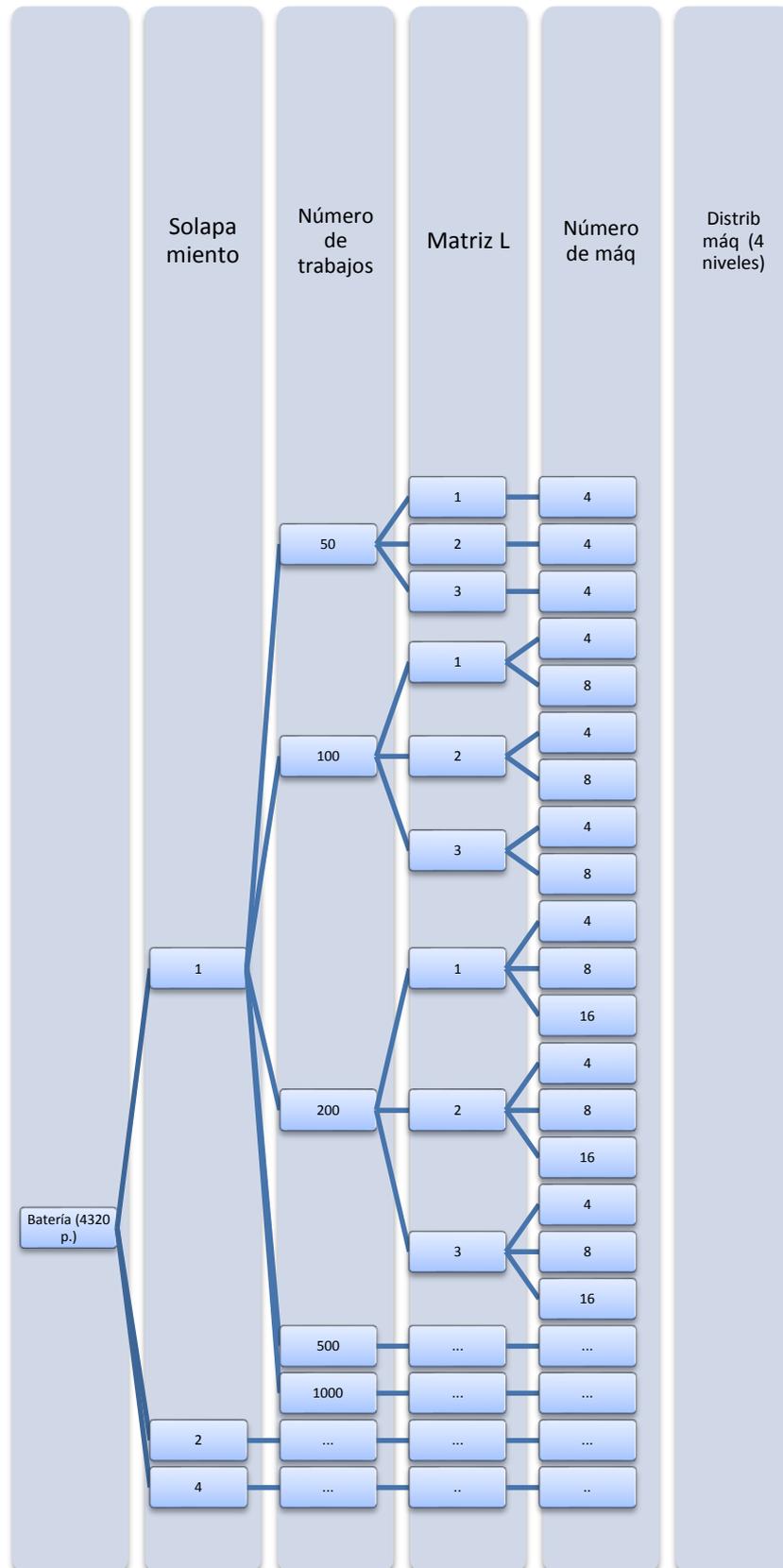


Figura 6 Esquema de distribución de problemas de la batería utilizada

Cada uno de estos parámetros nos permitirá caracterizar cómo se comporta cada una de las estrategias propuestas para el estudio en función de ellos y sus combinaciones. A continuación, pasamos a describir cada

uno de ellos para poder entender más adelante los resultados cuando veamos los efectos causados.

- Solapamiento (3 grados): Este primer parámetro nos da información sobre cómo de solapados se encuentran las tareas que debemos cubrir. Es decir, un grado mayor de solapamiento significará que en cada uno de los instantes de estudio encontraremos más trabajos simultáneos y por lo contrario un grado de solapamiento menor significará que los trabajos están más esparcidos en la línea temporal.
- Número de trabajos [50 100 200 500 1000]: Cada uno de los problemas de la batería contiene un número distinto de trabajos siendo el menor de 50 y el mayor de 1000. El efecto de este parámetro se podría predecir que afectará al tiempo de ejecución por la complejidad del problema en conjunto con un grado de solapamiento grande.
- Tipo de matriz de compatibilidad [1 2 3]: Hemos definido tres tipos de matriz de compatibilidad. Ya hemos mostrado las matrices de compatibilidad que serán usadas en la batería, se puede comprobar por ejemplo que la matriz número 2 tiene muy marcadas las compatibilidades lo que hará que en conjunto con el parámetro de distribución de recursos condicione el resultado. Además, dependiendo del tipo de matriz podremos distinguir los problemas que tienen en un mayor o menor número de clases de trabajo existentes.
- Cantidad de recursos [4 8 16]: Este parámetro muestra la cantidad de máquinas totales disponibles para realizar los trabajos. Lógicamente cuanto mayor sea el número de máquinas mayor será la cantidad de trabajos realizados pero nos permitirá estudiar como de eficaces son nuestras estrategias ante la escasez o abundancia de recursos. Además como podemos observar para los problemas de la batería con menor número de trabajos las mayores cantidades de trabajos se han omitido porque el resultado no sería significativo.
- Grado de distribución de los recursos (4 grados): Este parámetro nos muestra el reparto de máquinas en cada una de las 3 clases. Cuanto mayor sea el grado de distribución, las máquinas estarán concentradas en la última clase de máquina, mientras que el menor grado las aglutina en la primera clase y los grados centrales hace un reparto más equitativo. Este parámetro resulta más revelador en conjunto con el parámetro de matriz de compatibilidad tipo 2 ya que es el caso en el que se observa gran variación de compatibilidad entre las máquinas.
- Contador de problema (1 a 10): Este contador nos permite hacer el estudio más representativo ya que hemos generado 10 problemas de cada una de las combinaciones de los parámetros anteriores para evitar efectos negativos de la aleatoriedad a la hora de generar los problemas.

Finalmente la combinación de los parámetros en la manera explicada concluye como resultado en una batería de 4320 problemas.

4 ESTRATEGIAS DE RESOLUCIÓN

A continuación vamos a definir cada una de las estrategias de resolución que vamos a aplicar a la batería de problemas ya descrita. Cada una de ellas será simulada como explicaremos más adelante para obtener una solución de estos problemas de scheduling.

Aunque el número total de estrategias desarrolladas sea mayor, las podemos englobar en dos grandes bloques diferenciados por el sentido principal que toman a la hora de la toma de decisiones. A grandes rasgos estos dos grandes módulos se caracterizan de la siguiente manera:

- Estrategias tipo 1: Priorización en función de la compatibilidad de los trabajos
- Estrategias tipo 2: Priorización en función de la compatibilidad de las máquinas

Las estrategias del primer tipo están caracterizadas por asignar las prioridades en función de cómo de fácil será realizar una clase de trabajo es decir, si tomamos una clase de trabajo como la más prioritaria nos podremos asegurar la realización del trabajo más difícil de realizar de forma que no desaprovechemos las máquinas que son capaces de hacerlos en realizar otros trabajos más compatibles.

Como veremos más adelante habrá varios niveles de decisión ya que este único criterio de decisión no era suficiente para definir claramente la estrategia a seguir. Sin embargo, todas las estrategias tipo 1 tendrán en común este propósito a alto nivel.

Las estrategias del tipo 2 por lo contrario basarán su decisión directamente sobre la utilidad de las máquinas, de forma que tomen como nivel superior o inferior de prioridad la máquina más o menos compatible con los trabajos a realizar, sin influir a alto nivel de decisiones cómo de compatibles son los trabajos que realiza dicha máquina. De igual manera, este nivel de decisión no era suficiente por lo que se han creado varias estrategias para decisiones de nivel inferior.

4.1 Esquema básico de resolución

Antes de continuar con la explicación más en profundidad de las estrategias seguidas, debemos explicar el resultado que buscamos con dichas estrategias y por qué son necesarias en la resolución de problemas de scheduling de manera heurística.

Cómo ya hemos explicado, el marco donde englobamos nuestros problemas es del tipo operacional de varias clases, tanto FSP como VSP.

Para resolver dicho problema empíricamente el flujo a seguir es el siguiente:

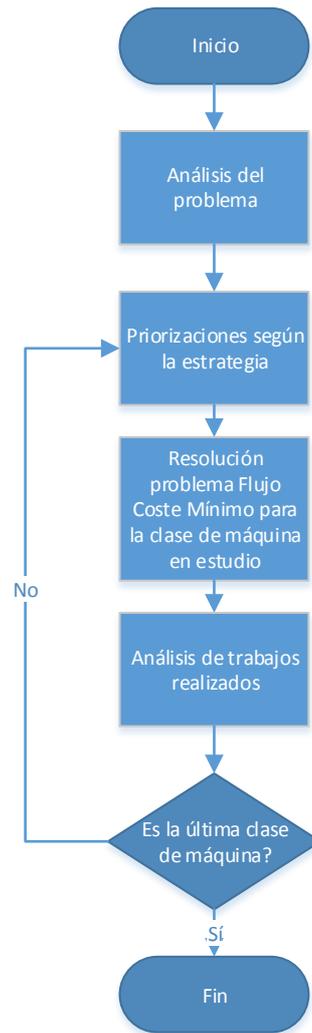


Figura 7 Esquema básico de resolución

La resolución de cada uno de los problemas de forma heurística comienza con la lectura de todos los trabajos, compatibilidad, máquinas disponibles, etc...

Posteriormente en función de los datos leídos debemos pasar a la priorización en función de los criterios de cada una de las estrategias.

Esta priorización es necesaria ya que el diagrama de flujo pasa por la resolución de un problema flujo coste mínimo con las características que explicamos más adelante. A grandes rasgos debemos tener en cuenta que deberá realizarse uno por cada clase de máquina. Este es el motivo de la necesidad de priorización e iteración.

En el caso de los problemas VSP explicaremos más adelante la solución planteada para la simulación y veremos cómo influye esta clase de problemas en la resolución de este diagrama.

4.1.1 Problema flujo a coste mínimo

Como hemos visto anteriormente, la resolución de dicho problema es parte del diagrama de flujo a alto nivel de este estudio, pero debemos dedicar un punto específico al problema flujo coste mínimo y a las características que lo definen.

Las características originales de cada uno de nuestros problemas ya han sido descritas, y como ya sabemos

este estudio se centra en los problemas de tipo operacional de varias clases, sin embargo, a continuación veremos las consecuencias que tiene esto sobre la resolución.

Para ello vamos a observar el caso más simple, que sería la resolución de un problema de asignación de una clase. La solución a un problema de este tipo como sabemos se puede realizar mediante un algoritmo de solución de Flujo Coste Mínimo con las siguientes características:

- Grafo Dirigido y Conexo $G(N,A)$: Donde N sería el conjunto de instantes de tiempo a estudio y A el conjunto de arcos que podrán ser de tipo trabajo o simplemente transporte
- Flujo de entrada y salida en cada nodo debe ser igual. Excepto en los nodos de inicio y fin donde el flujo será igual al número de máquinas (en el caso que estamos nombrado, una sola clase de máquina de forma que este será el número total de máquinas)
- c_{ij} : El coste de cada arco dependerá de si se trata de un arco trabajo o transporte:
 - Los arcos trabajo deberán tener un coste igual al peso del trabajo en negativo
 - Los arcos transporte tendrán peso igual a 0
- k_{ij} : La capacidad de cada arco también dependerá del tipo de arco:
 - Los arcos tipo trabajo deben tener capacidad limitada a 1
 - Los arcos tipo transporte tendrán capacidad ilimitada (fijada a un valor elevado)

El problema flujo coste mínimo se define así de forma que al ejecutar el algoritmo de resolución de los problemas FCM los arcos prioritarios por los que pasará el flujo con el objetivo de minimizar el valor del coste serán los arcos trabajo de mayor peso (más negativos). La capacidad de estos arcos ha sido fijada a valor 1 de forma que nos aseguramos que solo puedan ser realizados una vez. Además el flujo de entrada y salida del primer y último nodo es la cantidad de máquinas de forma que nuestro problema queda completamente definido.

El grafo de resultado del FCM nos permite interpretarlo de forma que los arcos trabajo por los que haya pasado el flujo son aquellos trabajos que se ha decidido que se ejecuten por ser los que más mejoran nuestra función objetivo.

Una vez explicado el algoritmo de resolución para un problema operacional de una clase es fácil ver la analogía para nuestro caso operacional de varias clases con la ayuda del diagrama de flujo antes mostrado.

En nuestro sistema, al coexistir varias clases de máquinas y trabajos, sólo podremos realizar un grafo que pueda ser resuelto con el algoritmo FCM para cada una de las clases de máquina. De esta forma el grafo que construimos es análogo teniendo en cuenta las siguientes consideraciones:

- El flujo que inyectamos y requerimos en los nodos de origen y fin respectivamente debe ser igual a la cantidad de máquinas disponibles de la clase que estemos estudiando
- En cada iteración se debe construir un grafo específico que recoja los trabajos compatibles con la máquina en cuestión
- Al realizar el análisis de resultados del FCM debemos tener en cuenta qué trabajos han sido ya realizados para construir los siguientes grafos

Estas consideraciones son fáciles de comprender sabiendo la limitación existente para aprovechar la resolución de grafos FCM de realizar uno por cada clase de máquina. Es muy importante ver cómo después de cada iteración de FCM se debe reconstruir el grafo de entrada para tener en cuenta tanto las compatibilidades de la máquina en estudio como los trabajos que ya han sido realizados por otras clases de máquinas previas.

Podemos adelantar al observar los procedimientos del algoritmo que la solución no hará fácil alcanzar el

óptimo ya que atacar el conjunto de problemas de forma separada por la compatibilidad de cada máquina lo segmenta y puede hacer que no queden reflejadas las prioridades del conjunto total. Aun así esto se verá con más detalle al realizar el análisis de los resultados.

Todo esto se verá con más detalle a la hora de detallar la estructura del código de simulación, pero a continuación vamos a mostrar un ejemplo de resolución de un problema de programación FSP operacional de una clase para facilitar la comprensión de este capítulo.

El modelo de problema es el siguiente:

$$\begin{aligned}
 & \text{Max } \sum_{i=1}^n w_i x_i \\
 & x_i = \begin{cases} 1 & \text{si } J_i \text{ se realiza} \\ 0 & \text{si } J_i \text{ no se realiza} \end{cases} \\
 & \sum_{\{i/s_i \leq t \leq f_i\}} x_i \leq m \quad t \in T / t = s_k \forall J_k \in J
 \end{aligned}$$

Este modelo de problema hace referencia a un problema de asignación para un escenario como el que mostramos en el siguiente gráfico:

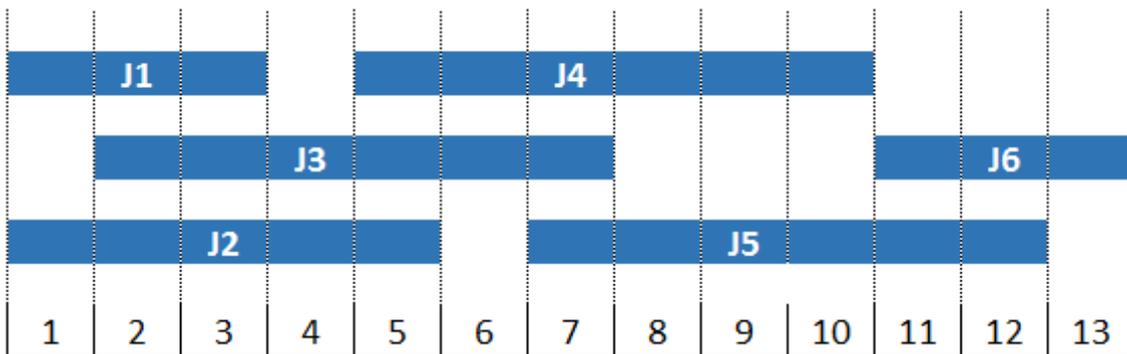


Figura 8 Ejemplo de problema FSP Operacional de una clase

Este ejemplo quedaría reflejado en el modelo mediante las siguientes ecuaciones tomando la cantidad de máquinas como dos y que sólo existe una clase de trabajo.

$$\begin{aligned}
 & \text{Max } w_1 x_1 + w_2 x_2 + w_3 x_3 + w_4 x_4 + w_5 x_5 + w_6 x_6 \\
 & x_1 + x_2 \leq 2 \quad t = 1 \\
 & x_1 + x_2 + x_3 \leq 2 \quad t = 2 \\
 & x_2 + x_3 + x_4 \leq 2 \quad t = 5 \\
 & x_3 + x_4 + x_5 \leq 2 \quad t = 7 \\
 & x_5 + x_6 \leq 2 \quad t = 11
 \end{aligned}$$

Sin embargo, en este proyecto estamos utilizando un método alternativo mediante el cual construimos un grafo con las propiedades que indicábamos un poco más arriba. En este ejemplo, el grafo en cuestión quedaría representado de la siguiente manera.

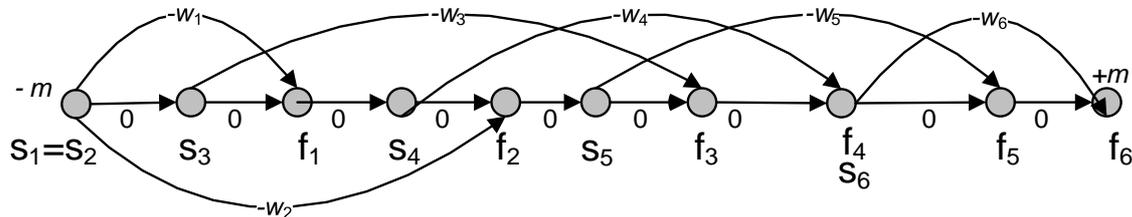


Figura 9 Ejemplo grafo de problema Flujo Coste Mínimo

Este sería el grafo en el que se estudia una sola clase de máquina ya que no podemos crear diferenciación de clases en el FCM, el valor m es la cantidad de máquinas de la clase estudiada.

En nuestro caso, el problema que queremos estudiar se lleva a cabo con la utilización de varias clases de recursos compatibles únicamente con algunas clases de trabajos por lo que se construirá un grafo por clase de máquina. Como el grupo de trabajos es compartido es ahí cuando al resolver el problema de forma heurística, surge la necesidad de priorizar cuál de las clases de máquinas será la primera en enfrentarse al grupo de trabajos.

En los siguientes puntos explicamos cómo se han llevado a cabo estas estrategias de priorización.

4.2 Estrategias de priorización

4.2.1 Estrategias tipo 1

Como ya hemos adelantado anteriormente este conjunto de estrategias se caracteriza por tomar como máxima a la hora de asignar las prioridades, cómo de compatible es cada trabajo.

Antes de nada debemos destacar que en este tipo de estrategia surge el papel de un segundo nivel de decisión bien diferenciado. Este segundo nivel aparece debido a que una vez que decidimos en un primer nivel de decisión cual será la clase de trabajo prioritaria, deberemos decidir de igual manera, cuál de las clases de máquinas capaces de realizar dicho trabajo según la matriz de compatibilidad será la encargada de realizarlo.

Esto es debido a que el problema Flujo Coste Mínimo siempre debe realizarse para una clase de máquina en concreto, así que no nos basta con priorizar la clase de trabajo que queremos que sea más relevante, sino que habrá que decidir con qué clase de máquina pasarla al algoritmo de Flujo Coste Mínimo.

De esta forma comienza la ramificación de estrategias, comenzando en su primera división por la decisión de priorizar los trabajos más compatibles o los menos compatibles. Esta decisión equivaldría a:

- Priorizar la realización de los trabajos más difíciles de hacer para asegurarnos que quedan realizados en vez de desaprovechar las máquinas especialistas en estos trabajos usándolas para hacer trabajos más compatibles con las demás máquinas
- Priorizar la realización de los trabajos más compatibles. Esta decisión se podría traducir en hacer los trabajos más comunes para asegurarnos que realizamos el grueso de los trabajos

En este marco de estrategias como veremos a continuación, priorizamos los trabajos menos compatibles las estrategias 1.1 y 1.3; mientras que priorizando los trabajos más compatibles hemos realizado las estrategias 1.2 y 1.4.

El segundo nivel de decisión del que hablábamos y es el que hace que haya 2 estrategias de cada uno de los tipos, es la decisión de prioridad entre las máquinas capaces de realizar el trabajo prioritario.

De igual manera, el criterio en esta ocasión será: el grado de compatibilidad de las máquinas, haciendo que el desglose se pueda traducir de la siguiente manera. Siempre tendremos en cuenta que la decisión se realiza de entre el conjunto de clases de máquinas que la matriz de compatibilidad dicte como compatibles con el trabajo decidido a priorizar. Los dos caminos posibles son:

- Priorizar la utilización de la máquina menos compatible, de las capaces de hacer el trabajo prioritario escogemos las que son por decirlo de alguna manera más especializadas en poca cantidad de trabajos como el que hemos priorizado
- Priorizar la utilización de la máquina más compatible, de las capaces de hacer el trabajo prioritario usaremos las que sean capaces de hacer mayor cantidad de clases de trabajos

En este escalón de decisión podremos marcar como estrategias encaminadas a escoger la máquina menos compatible las estrategias 1.1 y 1.2; y las encaminadas a escoger la máquina más compatible serán las estrategias 1.3 y 1.4.

Además, cabe destacar que debido a las matrices de compatibilidad con las que estamos trabajando, al ser de un tamaño pequeño, es decir, bajo número de clases de trabajos y de máquinas, existirían muchas ocasiones en las que las decisiones no serían únicas por lo que nos vemos obligados a imponer más subniveles de decisión de carácter general con el objetivo de maximizar los resultados.

Estos subniveles se aplicarán en los casos en los que los niveles anteriores de decisión den lugar a varias clases de trabajos que priorizar o varias clases de máquinas igual de prioritarias.

En el caso de que varios trabajos sean igual de compatibles, es decir, el primer nivel de decisión proponga varias clases de trabajos, tomaremos como trabajo prioritario el que sume mayor valor de sumatorio de pesos. Este subnivel se aplicará teniendo en cuenta solo aquellos trabajos que no hayan sido realizados aún.

De igual manera, cuando varias máquinas resulten como consecuencia de la utilización del segundo nivel de decisión, tomaremos como prioritaria la que tenga menor número de máquinas para las estrategias 1.1 y 1.2; y mayor número de máquinas para las estrategias 1.3 y 1.4.

Los niveles de decisión aquí descritos pueden verse reflejados en los siguientes diagramas de flujo.

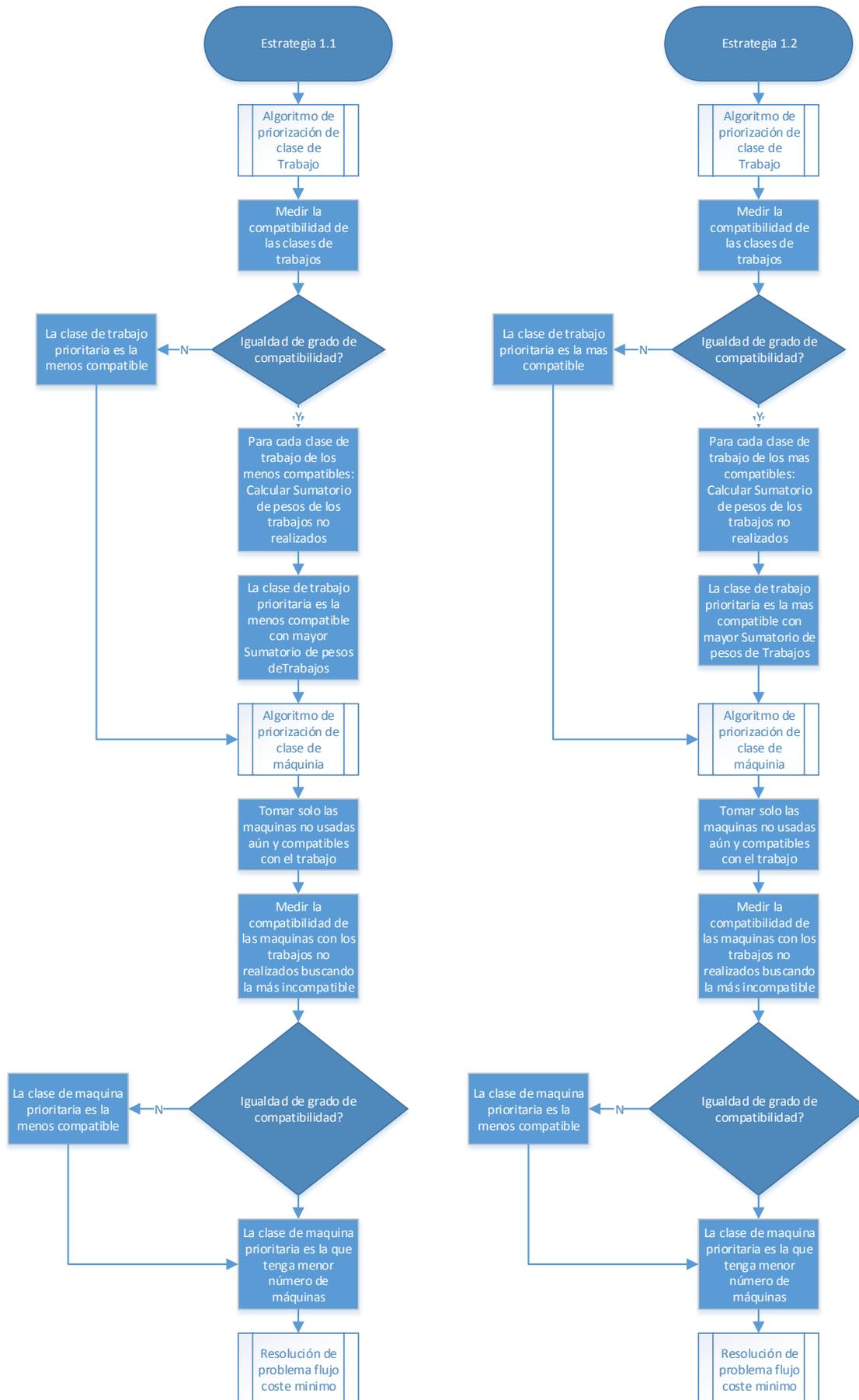


Figura 10 Esquema de estrategias 1.1 y 1.2

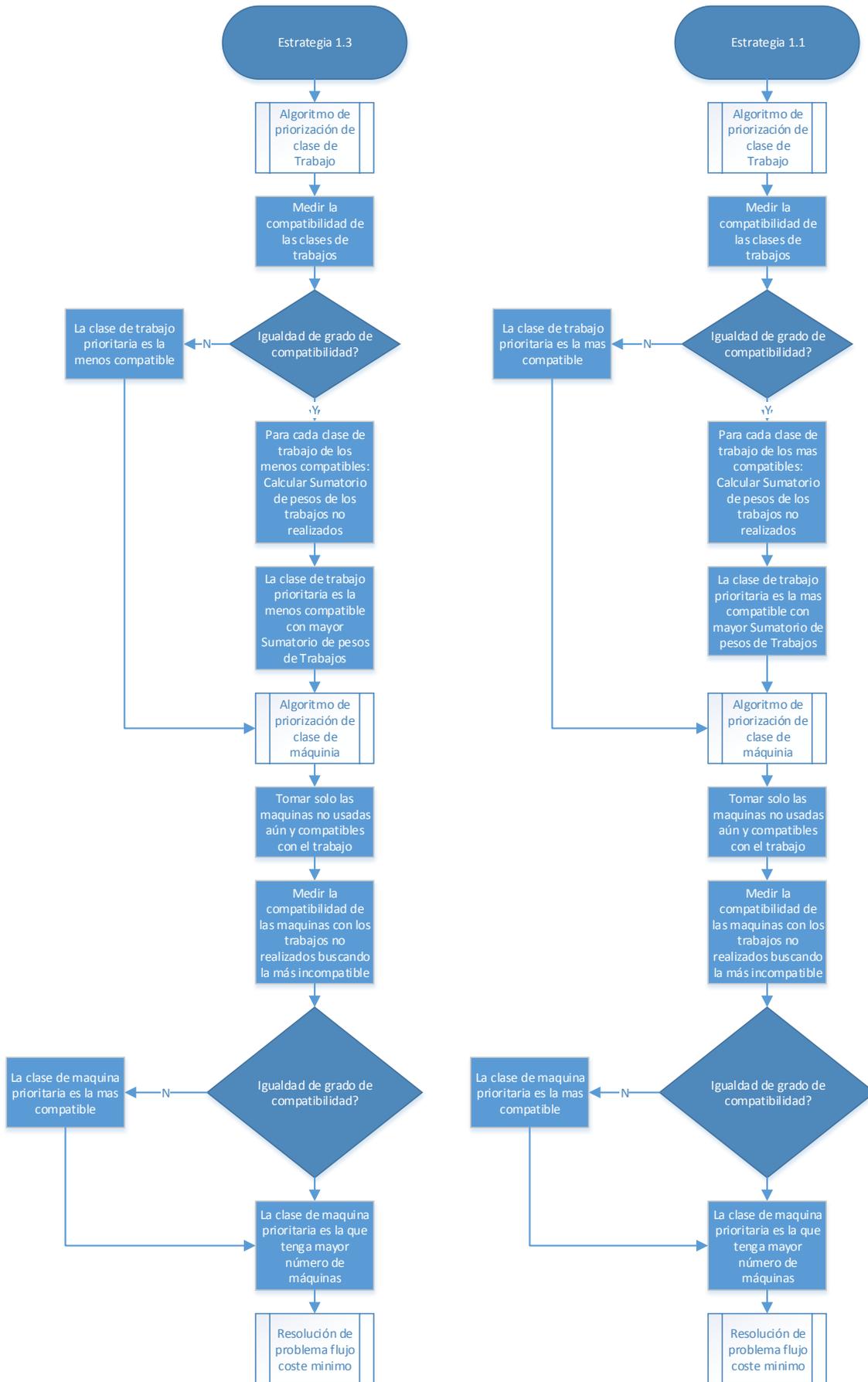


Figura 11 Esquema de estrategias 1.3 y 1.4

4.2.2 Estrategias tipo 2

Igual que hemos adelantado anteriormente, las estrategias tipo 2 están centradas en realizar la priorización en función de la compatibilidad de las máquinas del problema. Así que este será el nivel superior a tener en cuenta a la hora de la toma de decisiones.

En este caso, las estrategias tipo 2 resultan ser más simples y fáciles de implementar que las estrategias tipo 1 ya que al ser directamente la clase de máquina la que provoca la decisión de la prioridad y la que va a ser priorizada, hace que el algoritmo de implantación se acorte.

En las estrategias tipo 2, el primer nivel de toma de decisión se realiza midiendo la compatibilidad en la matriz L de cada una de las clases de máquinas disponibles. Según los resultados de esta medición, surgen 2 caminos posibles:

- Priorización de las máquinas menos compatibles: nos aseguramos de asignar en primer lugar las máquinas menos útiles o más específicas de forma que se ataca al conjunto de trabajos más específico
- Priorización de las máquinas más compatibles: comenzamos la asignación con las máquinas que podrán realizar el mayor grueso de las clases de trabajo

Al realizar esta división de caminos, se generan las dos nuevas estrategias del tipo 2.1 para la clase de máquina menos compatible y la 2.2 para la clase de máquina más compatible

De igual manera que las estrategias 1, surgen más subniveles de decisión cuando se produce una igualdad en el nivel superior. En esta ocasión, los subniveles van encaminados a buscar la clase de trabajo más compatible y de igual manera, en el caso de igualdad, la clase de trabajo que tenga mayor sumatorio de pesos de los trabajos no realizados aún. Esto se podría traducir en buscar la máquina que realiza el mejor beneficio en nuestra función objetivo.

Es fácil ver, la mayor simplicidad de las estrategias 2 ya que se basan directamente en las características de las máquinas para priorizar una de ellas.

Los niveles de decisión ya explicados se reflejan en el siguiente diagrama de flujo.

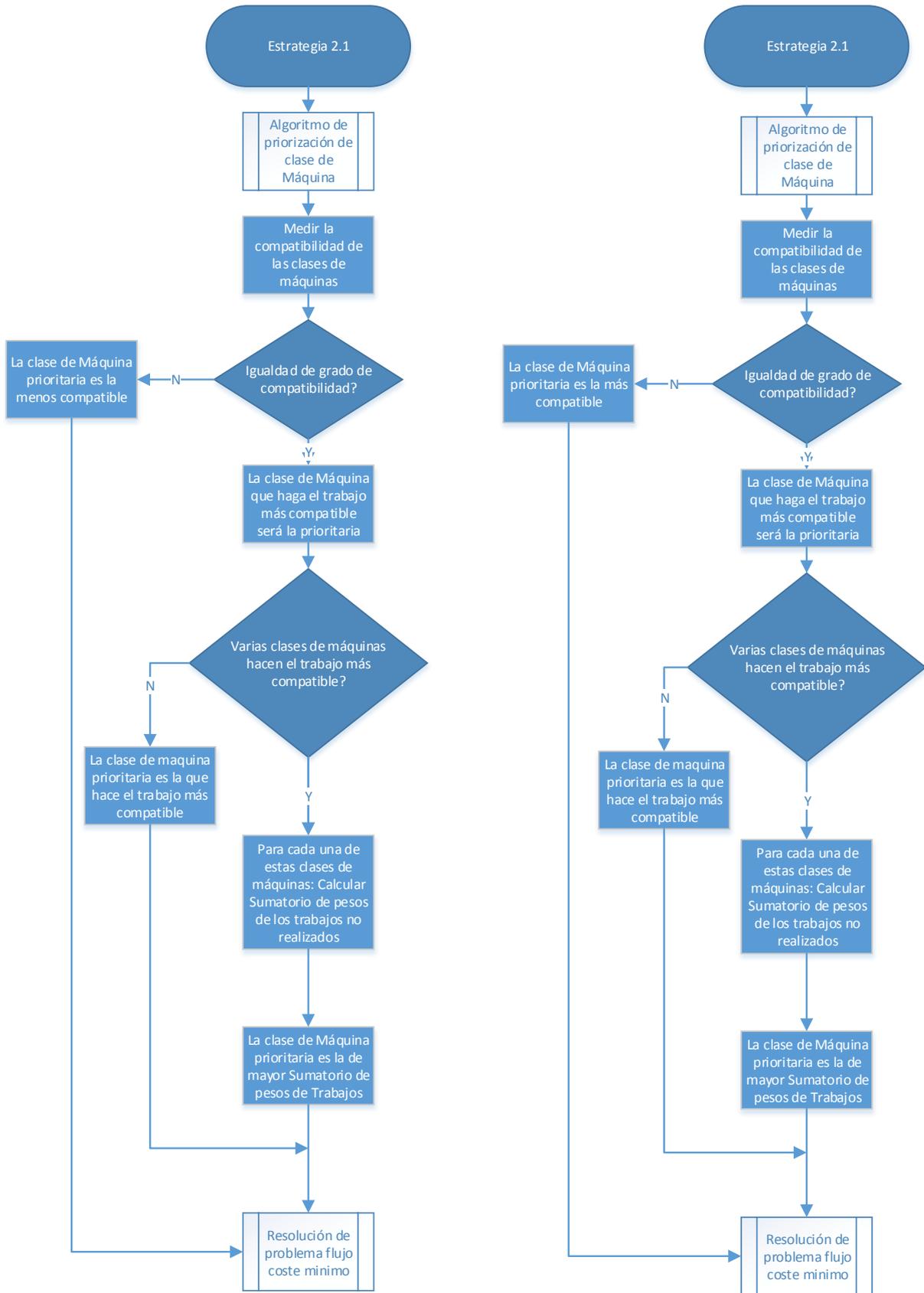


Figura 12 Esquema estrategias 2.1 y 2.2

Una vez definidos los trabajos a realizar, los recursos de los que disponemos y las estrategias de resolución que vamos a implementar llega el momento de definir el tipo de restricción temporal que vamos a aplicar.

La primera de las restricciones temporales que vamos a aplicar será Fixed Scheduling Problem (FSP) en el que el tiempo disponible para realizar el trabajo coincide exactamente con el tiempo que necesitamos para realizarlo por lo que no existe la posibilidad de desplazar temporalmente la realización del trabajo.

Este tipo de resolución es siempre más restrictiva que la Variable Scheduling Problem (VSP) que nos permitirá una variabilidad mínima de desplazamiento en el tiempo de los problemas. En la práctica podemos encontrarnos a menudo con restricciones fijas tipo FSP pero suele existir una ventana temporal mínima en la que podemos recolocar la realización del trabajo si al hacer esto consiguiéramos una mejora del aprovechamiento de la planta o el sistema que estemos explotando. Es por esto que estudiaremos también la posibilidad de un entorno VSP

Para realizar este tipo de problemas utilizaremos directamente los tiempos estipulados en la batería de problemas.

4.3 Particularidades de heurística de problemas VSP

Como hemos dicho, hemos considerado que aunque los tiempos de ejecución de los trabajos pueden venir fijados por requisitos del proceso, también puede existir la posibilidad de la apertura de una ventana temporal en la que podamos desplazar la realización de algunos o todos los trabajos si siendo así obtenemos un mayor aprovechamiento de los recursos de los que ya disponemos.

Con este propósito pasamos a realizar también el estudio de los problemas tipo VSP Operacionales de varias clases.

Este tipo de aproximación nos impondrá varias restricciones como ya veremos con más detalle pero a grandes rasgos presentamos en este punto estas restricciones.

Como ya sabemos, la resolución heurística que tratamos se basa en la resolución del algoritmo de flujo coste mínimo por cada una de las clases. Para poder introducir la variable de que el problema sea VSP debemos abrir una ventana temporal de $\pm x$ segundos. Sin embargo, para poder asegurar que cumplimos el requisito de no permitir que se asignen varios trabajos simultáneamente a la misma máquina, dichos trabajos deben permanecer siempre solapados.

A continuación mostramos dos representaciones de trabajos con sus correspondientes grafos para ilustrar la necesidad de que en las resoluciones heurísticas, los problemas ficticios añadidos para permitir el desplazamiento de la resolución permanezcan siempre solapados.

El solapamiento debe producirse de la siguiente manera.

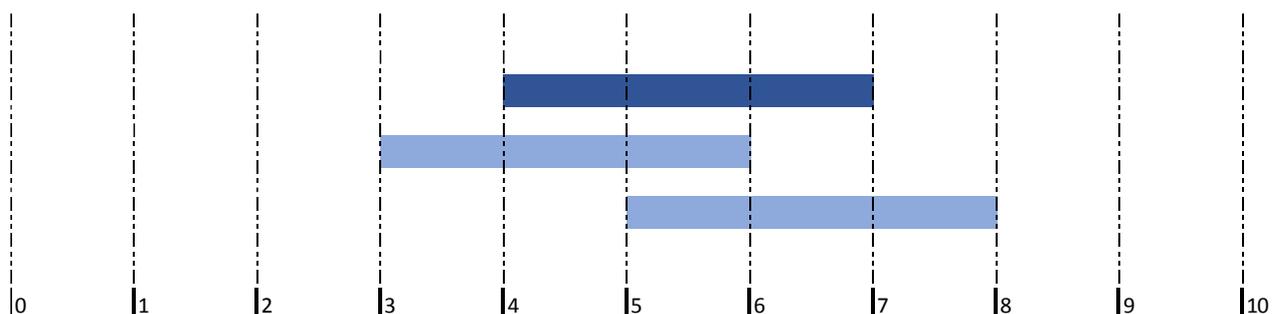


Figura 13 Esquema de trabajos ficticios correctamente creados

De esta forma, el grafo que se crea y que se resolverá mediante el algoritmo de flujo coste mínimo, es el siguiente.

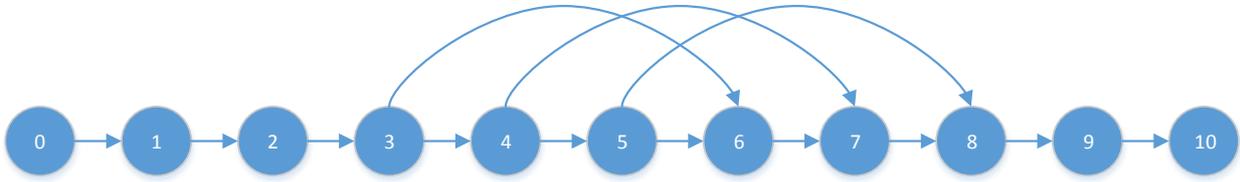


Figura 14 Grafo generado correctamente

Podemos ver como este grafo no permite la realización de más de uno de los arcos de transporte (cuando el flujo inyectado es de una máquina cómo hemos realizado en nuestras simulaciones). De esta manera, conseguimos evitar que se produzcan errores de asignar varias veces el mismo trabajo debido a que en el grafo se completaran varios de los arcos de trabajos ficticios.

Para mejor ilustración, mostramos a continuación cual sería la manera incorrecta de crear los trabajos ficticios, la cual sin lugar a dudas daría lugar a error.

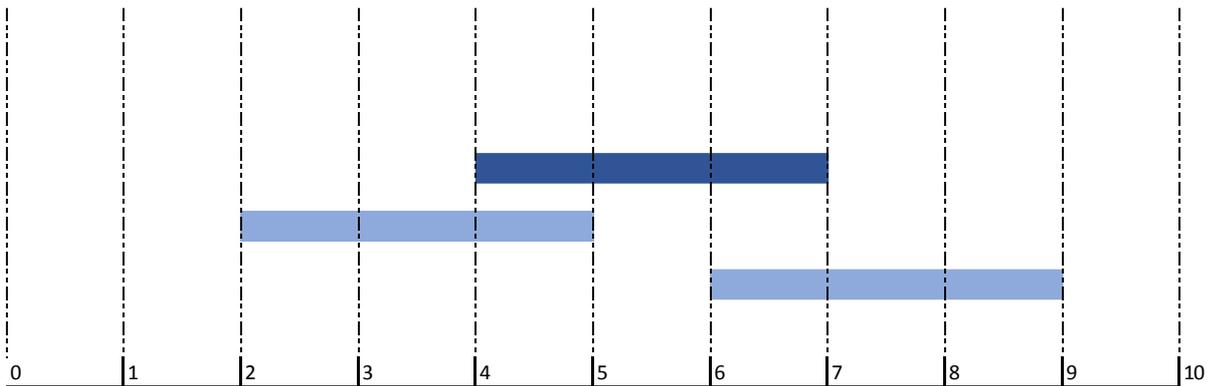


Figura 15 Esquema de trabajos ficticios erróneamente creados

En ella, los trabajos ficticios creados no están solapados correctamente, lo cual dará lugar al grafo que mostramos a continuación

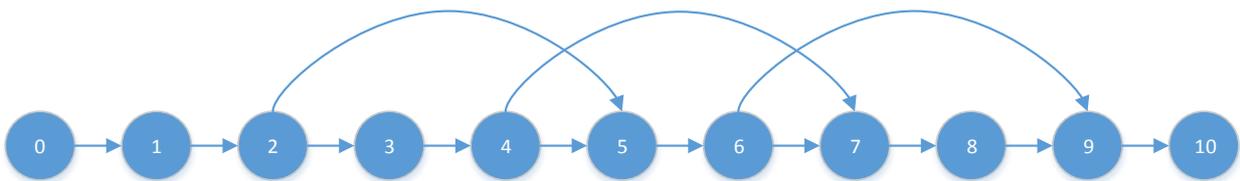


Figura 16 Grafo generado erróneamente

Con este grafo podemos ver cómo podrían realizarse por una sola máquina dos veces el mismo trabajo, lo cual duplicaría falsamente el beneficio reportado a la función objetivo.

Este efecto ha sido evitado usando la primera de las representaciones mostradas.

Como podemos observar, al realizar el grafo en el que cada nodo es un instante, intrínsecamente creamos una restricción que condiciona la cantidad de desplazamientos que pueden hacer nuestros trabajos en función de la longitud de los mismos.

De este modo, para poder crear desplazamientos temporales sin que se produzca un problema de doble asignación en nuestros trabajos, el menor tiempo de duración de nuestros trabajos debe ser de 3 segundos, es decir, 3 instantes de tiempo, con esto podremos desplazar los trabajos ± 1 segundo. Este tiempo de duración nos permite abrir dos posibilidades de desplazamiento; una comenzando y terminado un segundo antes y otra un segundo después.

Esta restricción ha sido tenida en cuenta a la hora de crear los trabajos de la batería aunque se podría haber esquivado simplemente dejando como FSP aquellos trabajos que sean de duración menor a 3 instantes de tiempo.

Por otro lado, debemos tener en cuenta también otro elemento restrictivo de la clase de problema VSP a la hora de afrontarlos con una resolución heurística de este tipo.

Al realizar el gráfico de entrada para el problema de FCM, estamos introduciendo inicialmente un grafo en el que el flujo de inyección en el primero de los nodos es la cantidad de máquinas de la clase que llevamos a estudio. Esto es así porque podemos asegurar que cada máquina va a realizar solo un trabajo simultáneamente al conducir su flujo por un arco u otro.

Sin embargo, si procediéramos así en los problemas con aproximación VSP estaríamos cometiendo un error porque el grafo creado muestra un trabajo por cada uno de los posibles desplazamientos temporales dentro de la ventana proporcionada. De esta forma, varias máquinas de una misma clase podrían conducir su flujo por uno de los trabajos originales y también por uno de los trabajos desplazados temporalmente incurriendo así en la asignación de un mismo trabajo dos veces.

Para mitigar este riesgo, como veremos en el código, el estudio se realiza haciendo un problema FCM de cada máquina de cada clase de trabajo, de una en una.

Sin embargo, realizar esta simplificación provoca un aumento el tiempo de ejecución de las simulaciones además de una pérdida de optimalidad ya que realizar el FCM con la cantidad de máquinas de una en una provoca que no siempre se escoja la asignación más conveniente.

Por ejemplo en este esquema, podemos ver como el grafo con una inyección de una máquina y posteriormente de otra, consigue un valor menor de función objetivo que la que obtendría la inyección inicial de dos máquinas de la misma clase.

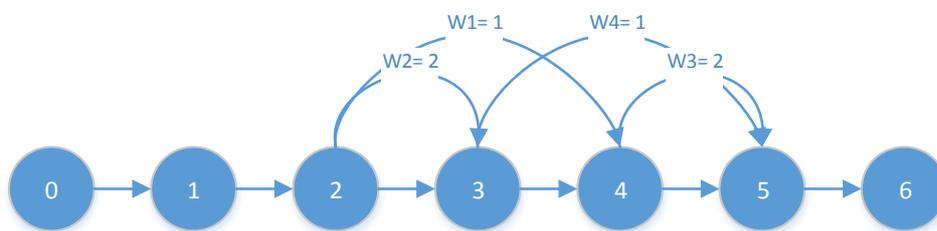


Figura 17 Efecto de la superposición

Como podemos observar, en este grafo se produce la siguiente asignación con un total de dos máquinas:

- Inyectar una máquina y después la otra nos daría un valor de función objetivo igual a 5 ya que la primera de ellas siguiendo el algoritmo FCM realizaría la asignación a los trabajos 2 y 3; mientras que la segunda solo podría realizar uno de los otros dos trabajos, o bien el trabajo 1 o el número 4.

- Inyectar directamente un flujo de 2 unidades de máquinas permitiría al algoritmo FCM obtener un mejor resultado de valor 6, ya que la asignación óptima del FCM permitiría realizar todos y cada uno de los trabajos.

Este efecto queda patente en el proyecto ya que en los problemas tipo VSP solo podemos realizar la asignación de las máquinas una a una. Más adelante podremos observar los resultados de dicho efecto en los resultados obtenidos.

5 PROGRAMACIÓN EN VISUAL BASIC

Llegados a este punto comentaremos a grandes rasgos el código utilizado para la realización de este proyecto, pasando en primer lugar por la función principal del código para los problemas tipo FSP y sus subrutinas incluidas la toma de decisiones, para después hacer el problema análogo en VSP resaltando las principales diferencias entre ambas formas de realizar el código del programa.

5.1 FSP

5.1.1 Función Main()

En los anexos podemos encontrar algunos de los códigos utilizados en los que mostramos el contenido exacto del programa con comentarios añadidos para facilitar su seguimiento. Para mayor facilidad y comprensión en la lectura del proyecto, en este capítulo comentaremos los grandes bloques de código que conforman el conjunto del programa.

Para la función principal Main() realizaremos una iteración para cada problema de la batería, es decir para cada archivo, en cada una de estas iteraciones encontramos los principales elementos de código entre los que cabe destacar los siguientes bloques que detallaremos:

- Lectura del problema
- Decisión de prioridades
- Asignación de prioridades
- Segmentación de los problemas factibles para la clase de máquina objetivo
- Generación del modelo de entrada al FCM
- Solución del FCM
- Análisis de resultado del FCM

Con estos elementos podremos resolver cada uno de los problemas. Debemos resaltar que en el apartado de toma de decisión influirá si nos encontramos con una estrategia tipo 1 o 2 ya que la estrategia tipo 2 solo deberá tener en cuenta la priorización de las máquinas. Sin embargo como ya sabemos, las estrategias tipo 1 primero deben realizar una priorización de trabajos y en consecuencia a dichos trabajos realizar una priorización de las máquinas. Más adelante podremos ver con más detalle cómo se realiza cada una de estas priorizaciones al centrarnos en cada una de las estrategias.

La información principal que debemos sacar de la función Main() es la forma en la que se estructura la resolución de problemas para después poder entender el objetivo de cada uno de los siguientes segmentos de código.

En el anexo correspondiente podemos observar que a la hora de realizar la lectura de los problemas se ha

tenido en cuenta como ya vimos más arriba que no existen ciertas combinaciones de problemas ya que no sería coherente por ejemplo problemas de 50 trabajos con 16 máquinas disponibles ya que estaría sobredimensionado.

Además, en este tramo de código vemos cómo al realizar la priorización de una máquina, se deja esto reflejado en la matriz `LModMach()` para que en la próxima iteración no se nos permita volver a utilizar la misma clase de máquina, ya que esto implicaría asignar por duplicado las máquinas de las que disponemos.

También podemos comprobar cómo se lleva a cabo la contabilización del tiempo de ejecución de cada problema con precisión de milisegundos, esto nos permitirá más adelante comparar las distintas estrategias no solo prestando atención al error relativo en el que incurren sino en el tiempo utilizado para llegar al resultado.

Pasamos a continuación a describir la funcionalidad de cada una de las funciones que se introducían anteriormente y que son llamadas desde la función `Main()`.

5.1.2 Función leer archivo de entrada

Podemos ver también el código utilizado por esta función que tiene como objetivo la lectura de cada uno de los archivos en los que se encuentran los problemas, recursos disponibles, etc... con toda la información que recogíamos en puntos anteriores.

La estructura de los archivos de problema para un pequeño ejemplo sería la siguiente:

```
10
2 32 30 253 1
2 7 5 348 2
3 10 7 1000 1
3 26 23 455 2
4 7 3 62 2
7 26 19 191 2
12 25 13 602 2
12 37 25 295 1
12 21 9 425 2
14 19 5 836 2

2
3
1 1 0
1 0 1
4
2
1
1
```

Equivalente a la siguiente información:

- Número de trabajos
- Datos temporales y tipo de cada trabajo: $s_i f_i t_i w_i d_i$
- Cantidad de clases de trabajos **D**
- Cantidad de clases de máquinas **C**
- Matriz de compatibilidad $L_{D \times C}$
- Cantidad de máquinas j
- Distribución de máquinas en cada una de las clases

El resultado de la función es:

- Estructura de información contiendo todos los trabajos del tipo:

```
Type DatosJobs
si As Long
fi As Integer
ti As Integer
wi As Integer 'peso inicial del trabajo
wwi As Integer 'creado para almacenar el peso original mientras modificamos
di As Integer 'añadido para especificar la clase de trabajo
ci As Integer 'añadido para almacenar la clase de máquina que ha resuelto este trabajo, útil para analizar los resultados en estrategia 2
li As Integer 'vector de ordenacion
mi As Integer 'añadido para asignar un nuevo indice de identificación en la estructura trabajosM que le quiero pasar a GenerarModelo
xi As Integer 'Indice que me dira si el trabajo esta realizado o no 1 realizado 0 no realizado
End Type
```

- Matriz de compatibilidad $L()$ recogiendo la relación entre recursos y trabajos
- Variables NumClasJobs y NumClasMachines que nos permiten conocer la cantidad de clases de recursos y tareas a las que nos enfrentamos, de forma que sabremos el grado de complejidad del problema.
- Vector Maquinas() con la cantidad de recursos disponibles de cada una de las clases

Al realizar la lectura de los trabajos, la estructura Trabajos() queda con la siguiente forma:

```
Trabajos().si: Instante de inicio
Trabajos().fi: Instante de fin
Trabajos().ti: Duración
Trabajos().wi: Peso
Trabajos().wwi: Peso auxiliar para poder modificar el peso del trabajo al priorizar
Trabajos().di: Clase de trabajo
Trabajos().ci: Vacio inicialmente, nos permitirá recoger qué clase de máquina lo realiza
Trabajos().li: Posición auxiliar para la trazabilidad con TrabajosM()
Trabajos().mi: Identificación auxiliar para TrabajosM()
Trabajos().xi: Marcador de trabajo realizado
```

Con esta información disponible ya podemos pasar a analizar el tipo de problema al que nos enfrentamos en esta iteración, lo cual nos permitirá pasar a la toma de decisiones de la estrategia. La información será fácilmente analizable ya que todos los trabajos están recogidos dentro de la misma estructura y con los mismos subíndices.

El siguiente punto por el que pasa la función Main() sería el de toma de decisiones pero como este tramo de código depende de cada una de las estrategias que estemos llevando a cabo preferimos mostrar inicialmente los tramos de código comunes a todas las formas de afrontar el problema. De esta forma el siguiente punto será la segmentación de los trabajos realizables por la máquina que resulta de la priorización hecha por la estrategia.

5.1.3 Segmentación de trabajos

En este tramo de código, el objetivo es realizar la segmentación de los problemas que son objetivo de la clase de máquina que estamos tratando en cada iteración del código.

Esto es así ya que una vez que hemos escogido la máquina priorizada debido a la estrategia seguida, dicha máquina solo podrá resolver cierto tipo de trabajos, dependiendo de la matriz de compatibilidad $L()$ de la que

disponga el problema en concreto.

Este paso es necesario ya que para la realización del algoritmo de flujo coste mínimo, no es posible la distinción de clases dentro del propio algoritmo.

Además, también se debe tener en cuenta la restricción de que cuando ya se hayan realizado más de una iteración dentro de un problema, existirán trabajos que ya han sido realizados, así que no deben ser tenidos en cuenta como parte de los trabajos que podríamos pasar al algoritmo flujo coste mínimo.

El código encargado de esta tarea sigue la siguiente lógica:

Recorrer la estructura de trabajos comprobando para cada uno de ellos si es compatible con la clase de máquina que este bajo estudio en esa iteración y si el trabajo no se ha realizado aún;

Si cumple ambas condiciones, el trabajo será añadido al conjunto de trabajos con los que crearemos el grafo objetivo del FCM.

```
' Qué clases de trabajos son compatibles con la máquina? -> Creo la matriz TrabajosM que será argumento de GenerarModelo
For IndiceOriginal = 1 To NumJobs          'Recorrer toda la matriz Trabajos() original
  If Trabajos(IndiceOriginal).xi = 0 Then  'Si el trabajo ya ha sido realizado no quiero que entre
    For d = 1 To NumClasJobs              'Recorrer la matriz L para todas las clases de trabajos
      If L(d, ClaseMaquina) = 1 Then      'Si la clase de trabajo es compatible con la clase de maquina
        If Trabajos(IndiceOriginal).di = d Then 'Si el trabajo es de esa clase de trabajo (compatible)
          IndiceM = IndiceM + 1           'Incremento el Indice que me va a decir cuantos trabajos tengo en TrabajosM
          Trabajos(IndiceOriginal).mi = IndiceM 'Almaceno cual va a ser su nuevo indice para poder hacer la conversion desde el archivo
        End If
      End If
    Next d
  End If
Next IndiceOriginal
```

Para la realización de este objetivo la estrategia ha sido crear una nueva estructura de trabajos llamada TrabajosM() en la cual solo pasan a estar los trabajos que cumplan la condición de no haber sido realizados ya y ser compatibles con la máquina objetivo del algoritmo de flujo coste mínimo.

La realización de un trabajo que marcada gracias al dato de la estructura Trabajos().xi que será valor 1 si el trabajo ha sido realizado y 0 si aún está por realizar.

Más adelante, veremos cómo a la hora de crear la estructura TrabajosM() en los problemas del tipo VSP debemos tener en cuenta más restricciones que las indicadas aquí.

5.1.4 Función generar modelo FCM y resolución

Una vez que tenemos segregados los trabajos que son objetivo de nuestro algoritmo FCM en la estructura TrabajosM() debemos pasar a reflejar dichos trabajos como un grafo.

En dicho grafo, representaremos como ya dijimos anteriormente los trabajos como arcos entre los nodos instante en los que deba ser realizado. El coste de dichos arcos variará según sea el peso de cada uno de los trabajos. La capacidad de los arcos trabajo estará fija a uno de forma que no pueda ser asignado varias veces el mismo trabajo.

Además, en dicho grafo también debemos crear los arcos transporte conectando cada uno de los nodos instante con las características de capacidad elevada (fijada a 1000) y coste 0 de forma que sean siempre

menos atractivos que los arcos trabajo.

El número de arcos y de nodos será en función del espacio de tiempo que sea necesario para realizar desde el primer trabajo al último y del número de trabajos contenidos en la estructura de información TrabajosM().

Un extracto de dicho grafo de entrada para la resolución del FCM seguiría la siguiente estructura.

```
100 442
1 7 1 -362
1 8 1 -256
...
97 99 1 -247
98 100 1 -163
1 2 1000 0
2 3 1000 0
...
98 99 1000 0
99 100 1000 0
-12
0
...
0
12
```

En ella podemos ver el siguiente patrón:

- Número de instantes y total de arcos (de transporte y de trabajo)
- Conjunto de arcos de trabajo: $s_i f_i k_i$ (*siempre* 1) $-w_i$
- Conjunto de arcos de transporte: $s_i f_i k_i$ (*siempre* 1000) $-w_i$
- Flujo inyectado en cada nodo (solo en inicial y final para FCM)

Por último debemos observar como el flujo inyectado en el primero de los nodos y demandado en el último de ellos debe ser la cantidad de máquinas disponibles del tipo de clase que estamos estudiando en cada iteración. Más adelante al realizar el estudio de los problemas VSP veremos cómo este punto también debe ser modificado.

5.1.5 Análisis de resultados FCM

Por último, la última de las funciones comunes a todas las estrategias será la llamada LeerArchivoSalida().

Dicha función, tiene como objetivo analizar los datos devueltos por la resolución del algoritmo FCM, en dichos datos aparecerá reflejado cual ha sido la asignación de trabajos que permite obtener el máximo peso de los que recibía de entrada a través de TrabajosM().

La estructura de información utilizada no es otra que un valor para cada nodo que contiene la información del flujo que transcurre por el en la solución de FCM.

Con dicha información deberemos pasar a marcar dichos trabajos como realizados a través del parámetro Trabajos().xi y sumar el valor obtenido a la función objetivo de nuestro problema.

El código desarrollado para esto podemos verlo en los anexos.

Una vez explicadas estas funciones podremos comprender mejor las referentes a la toma de decisiones de cada estrategia.

Debemos tener en cuenta también al revisar el código que existen funciones auxiliares como las encargadas de realizar impresiones del estado de la estructura Trabajos() que no serán descritas ya que su único objetivo es facilitar el desarrollo del código y aportar visibilidad mientras el programa es ejecutado.

5.2 Estrategias tipo 1

Como ya sabemos, la forma de priorización de las estrategias tipo 1 está condicionada de forma que el criterio principal será siempre tomando la decisión en función de la compatibilidad de los trabajos.

Debemos recordar que en este tipo de estrategias, el flujo a seguir fuerza que haya dos niveles de decisión claramente diferenciados. En primer lugar, qué clase de trabajo será prioritario y en segundo lugar, cuál de las clases de máquinas disponibles para realizarlo será la prioritaria.

Como consecuencia de esta forma de actuar, aparecen dos funciones extra muy simples en las cuales se modifica el peso de cada trabajo. Si un trabajo es prioritario, su peso será incrementado en 1000 unidades haciéndolo así más conveniente en el algoritmo FCM que cualquiera de los trabajos no priorizados (todos ellos con pesos inferiores a 1000).

Una vez realizada la iteración con el trabajo priorizado, otra función será encargada de devolver a los trabajos priorizados su peso original almacenado en una de las posiciones auxiliares de la estructura Trabajos().

Debido a la larga extensión de dichas funciones de toma de decisión, hemos decidido mostrarlas directamente en los anexos para mayor simplicidad de la información aquí recogida.

5.3 Estrategias tipo 2

Este tipo de estrategias como ya indicamos más arriba son más simples ya que se basan directamente en las características del conjunto de recursos para realizar la toma de decisiones sobre cuál de ellas será prioritaria.

De esta forma, todos los trabajos serán incluidos en la estructura TrabajosM() con su peso original y el objeto de la toma de decisiones será el orden con el que las máquinas son introducidas a la asignación.

5.4 Estrategias de control

Cabe recalcar que además de las 4 estrategias tipo 1 y las 2 estrategias tipo 2, hemos desarrollado una estrategia más de cada tipo de forma que esta nos sirva de control.

Estas 2 nuevas estrategias son llamadas de control ya que su objetivo es mostrar el máximo beneficio al que puede optar cada una de las 2 principales aproximaciones.

Es decir, gracias a ellas, podremos obtener una referencia de cómo de atractivas podrían llegar a ser cada una de las estrategias.

A grandes rasgos, lo que nos permitirán ver, como mostraremos en la sección de resultados es el objetivo ideal que podríamos alcanzar sacrificando siempre el tiempo tomado para ello, ya que la aproximación de estas estrategias de control no es otra que probar todas las combinaciones disponibles hasta llegar a la que nos reporta un mayor valor de la función objetivo.

Para llevar a cabo estas estrategias, hemos desarrollado dos archivos en los que se recogen todas las combinaciones posibles para cada estrategia.

Es decir, en las estrategias tipo 1, se abre un abanico de combinaciones muy grande, ya que debemos probar cada combinación de priorización de trabajos combinada con cada priorización de máquinas, haciendo que el número de veces que debemos ejecutar el algoritmo flujo coste mínimo sea muy elevado y por tanto el tiempo de ejecución también. El número de combinaciones posible dependerá de la matriz L de cada problema, ya que nos dirá cuántas clases de trabajos hay en cada uno de ellos.

En las estrategias tipo 2, deberemos realizar un abanico de combinaciones menos amplio aunque sigue siendo obviamente poco eficiente. Esto es así porque las combinaciones que debemos probar en esta aproximación serán todas las posibles en la ordenación de las máquinas, siempre sin repetir. Como en todos los problemas de la batería la cantidad de clases de máquinas es 3, en este caso si podemos concluir que el número de llamadas a la resolución del FCM para cada problema será 18 (6 combinaciones sin repetición posibles y 3 llamadas al FCM por cada combinación).

El código necesario para la realización de estas estrategias de control también se mostrará en los anexos ya que las funciones principales son similares a las de todo el conjunto y solo cambiará la parte de toma de decisiones. En estos dos casos no existe toma de decisión alguna, sino una pasada por todas las combinaciones posibles una por una buscando únicamente el aumento de la función objetivo con respecto a las anteriores.

5.5 VSP

El código mostrado hasta ahora es el referente a los problemas FSP, aunque la mayor parte aplica igual a los problemas VSP, tiene algunas modificaciones que debemos tener en cuenta y mostramos a continuación.

Lo primero que tenemos que tener en cuenta será la definición de la ventana temporal que queremos aportar a nuestros trabajos. Esta ventana temporal puede fijarse de varias maneras, mediante un tanto por ciento de la duración del trabajo, un valor fijo para todos los trabajos, un valor específico para cada trabajo... En este caso hemos tomado inicialmente la consideración de que todos los trabajos tengan una ventana temporal fija e igual, sin embargo, el código es abierto a la asignación de la ventana temporal de otra manera que fuera impuesta por la batería de problemas a la que nos enfrentemos.

Los principales cambios que debemos realizar para adaptar el código a los problemas tipo VSP son los siguientes:

- Creación de trabajos ficticios desplazados en el tiempo para simular la ventana temporal
- Evitar la realización de varios trabajos ficticios en la segregación
- Realizar el algoritmo FCM máquina a máquina para evitar duplicidades de asignación.

Como ya adelantamos anteriormente, para resolver problemas tipo VSP con la aproximación que estamos tomando, debemos crear trabajos ficticios desplazados en el tiempo de forma simétrica, es decir, si la ventana temporal es de ± 1 segundo; un trabajo ficticio empezará y acabará un segundo antes y el otro 1 segundo después.

Al realizar la asignación, debemos evitar que el resultado contenga más de uno de los trabajos así creados para cada trabajo original.

Para crear dichos trabajos añadiremos dos variables más a la estructura Trabajos() de forma que nos indique:

Type DatosJobs

xi As Integer 'Índice que me dirá si el trabajo está realizado o no 1 realizado 0 no realizado

ei As Integer 'Índice que me dirá si uno de los trabajos de la ventana de tiempo se ha realizado

ai As Integer 'Índice que indica en qué posición de la ventana temporal está este problema

ni As Integer 'Índice que indica cuantos trabajos he puesto en la ventana temporal, si es fija será el mismo número en todos los trabajos

End Type

Esta información será completada inicialmente al hacer la lectura de los trabajos y con ella, a la hora de crear TrabajosM(), podemos añadir una nueva condición que nos permite descartar aquellos trabajos ficticios de la ventana temporal para los que ya se haya realizado uno de los trabajos del grupo.

Además, al obtener los resultados del FCM deberemos marcar la componente Trabajos().ei de los demás trabajos de la ventana temporal para indicar que uno de ellos ha sido realizado y evitar que entren a TrabajosM() de nuevo.

InicioVentana = IndiceOriginal - Trabajos(IndiceOriginal).ai 'El primer trabajo de la ventana es el índice menos el contador de ventana
FinVentana = IndiceOriginal - Trabajos(IndiceOriginal).ai + Trabajos(IndiceOriginal).ni 'El último trabajo de la ventana temporal es el inicial de la ventana más el número de trabajos en la ventana

For iVentana = InicioVentana To FinVentana

Trabajos(iVentana).ei = 1 'Pongo a 1 para indicar que estos trabajos no han sido realizados pero no se deben tener en cuenta por ser de la misma ventana temporal

Next iVentana

Finalmente, al crear el archivo que contiene el grafo de FCM, la inyección de número de máquinas y la demanda en el último nodo debe ser de 1 así que habrá que realizar una llamada a dicho algoritmo por cada máquina de la clase bajo estudio. Esta restricción ya la explicamos en puntos anteriores y es fácilmente trasladable al código de resolución.

Los cambios en las estrategias de priorización no van más allá de encontrar una mayor cantidad de trabajos a la hora de afrontarlos y tener en cuenta si se ha realizado alguno de los correspondientes en la ventana temporal

6 RESULTADOS COMPUTACIONALES

Llegados a este punto pasamos a presentar los resultados de las simulaciones realizadas para comprobar cuál de las estrategias es la más atractiva y cuál será el punto de débil de la elegida.

Para ello vamos a diferenciar los dos grandes bloques de problemas con aproximación VSP o FSP

6.1 FSP

Comenzamos por la aproximación FSP por ser la de mayor simplicidad a la hora de realizar las simulaciones.

Para entender los resultados, antes de nada debemos saber que el parámetro a estudiar va a ser el error relativo entre el resultado de los problemas con la realización de nuestras estrategias y el resultado óptimo de cada problema, para poder así ver cuál de ellas es la más conveniente. Los resultados óptimos de los problemas han sido obtenidos en proyectos anteriores de esta misma línea de investigación.

También deberemos estudiar el tiempo medio empleado en resolver los problemas, ya que como veremos a continuación juega un papel importante a la hora de realizar la elección de la estrategia conveniente.

La primera visión que debemos plantear es el promedio de error relativo que nos reporta cada estrategia, dicho promedio queda representado en esta gráfica.

$$Error\ relativo = \frac{\acute{O}ptimo - Resultado\ heurístico}{\acute{O}ptimo}$$

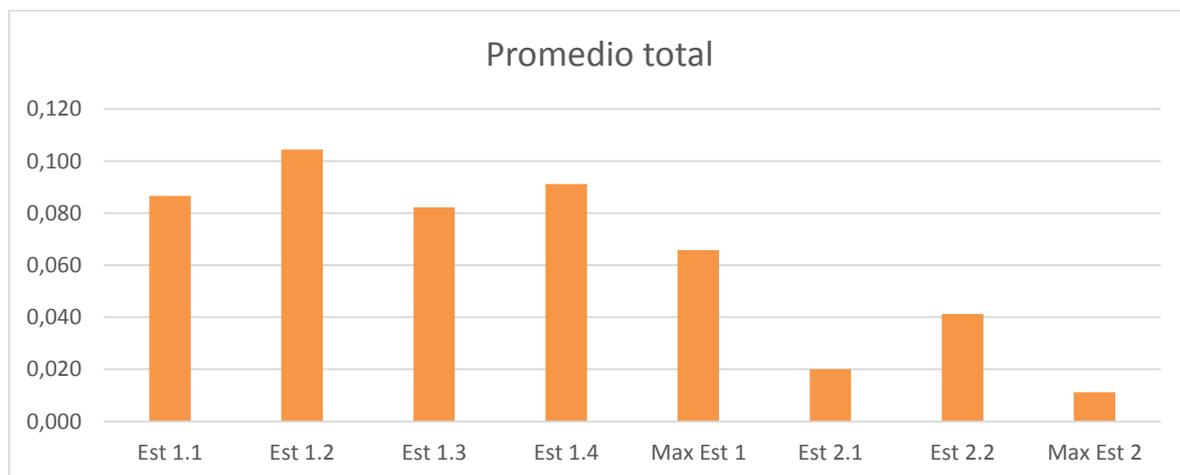


Figura 18 Promedio total de error relativo por estrategia FSP

6.1.1 Observación de tendencias generales en respuesta a cada parámetro

A primera vista podemos observar como las estrategias tipo 2 dan mejor resultado global que las estrategias tipo 1. Esta información nos la proporciona tanto cada una de las estrategias como los máximos de ellas. El mejor resultado que puede obtener una estrategia tipo 1 sería un error relativo promedio del 6,5%, mientras que el mejor resultado de la estrategia 2 podría llegar al promedio de 1,1% aproximadamente.

Sin embargo, para proporcionar la información con mayor detalle de cómo se comporta cada tipo de estrategia respecto a los parámetros de estudio vamos a realizar segregaciones de los tipos de problemas por cada uno de ellos.

Recordamos que los parámetros con los que podemos clasificar los problemas son:

- Solapamiento
- Número de máquinas
- Número de trabajos
- Tipo de matriz de compatibilidad
- Distribución de las máquinas

Presentamos el error relativo promedio de cada estrategia diferenciando el valor de cada uno de los parámetros de forma individual

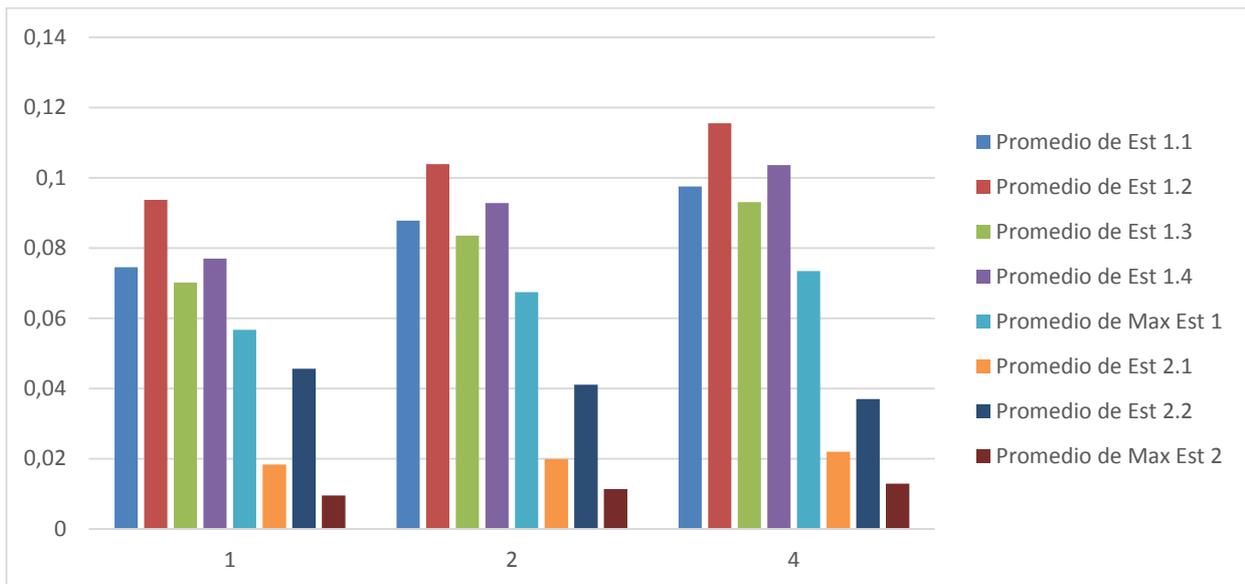


Figura 19 Comportamiento ante cada solapamiento FSP

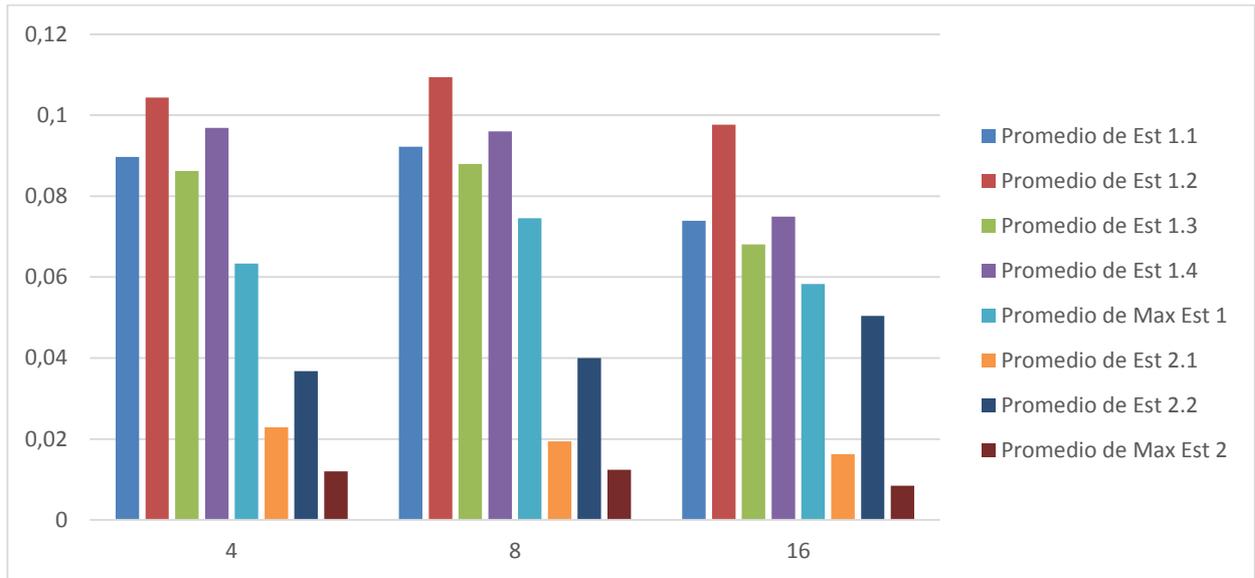


Figura 20 Comportamiento ante cada cantidad de máquinas FSP

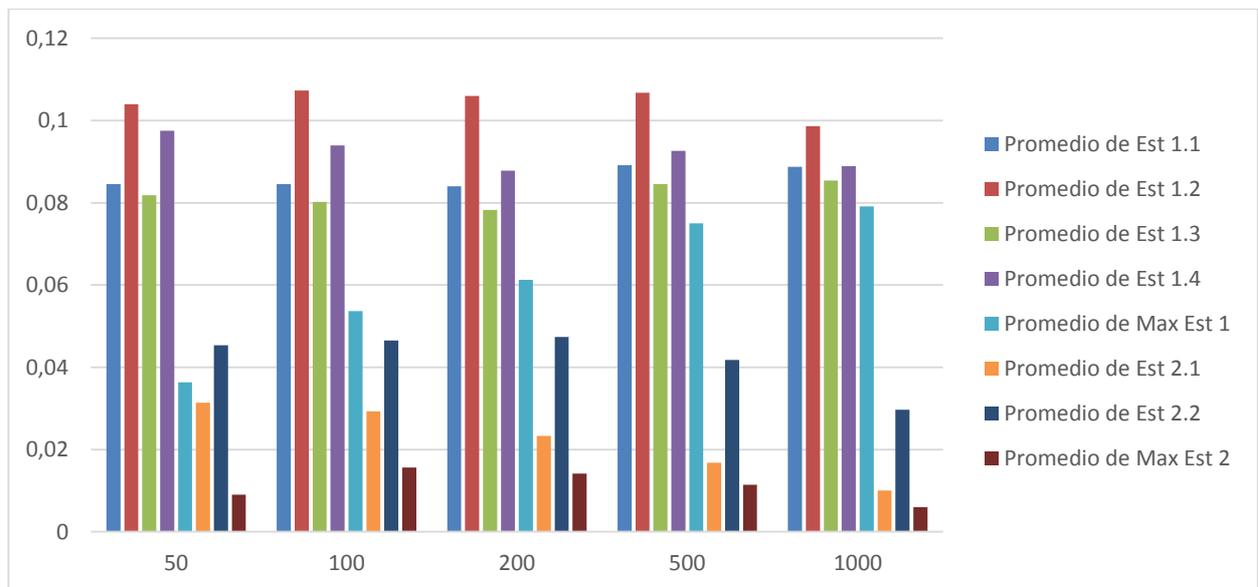


Figura 21 Comportamiento ante cada cantidad de trabajos FSP

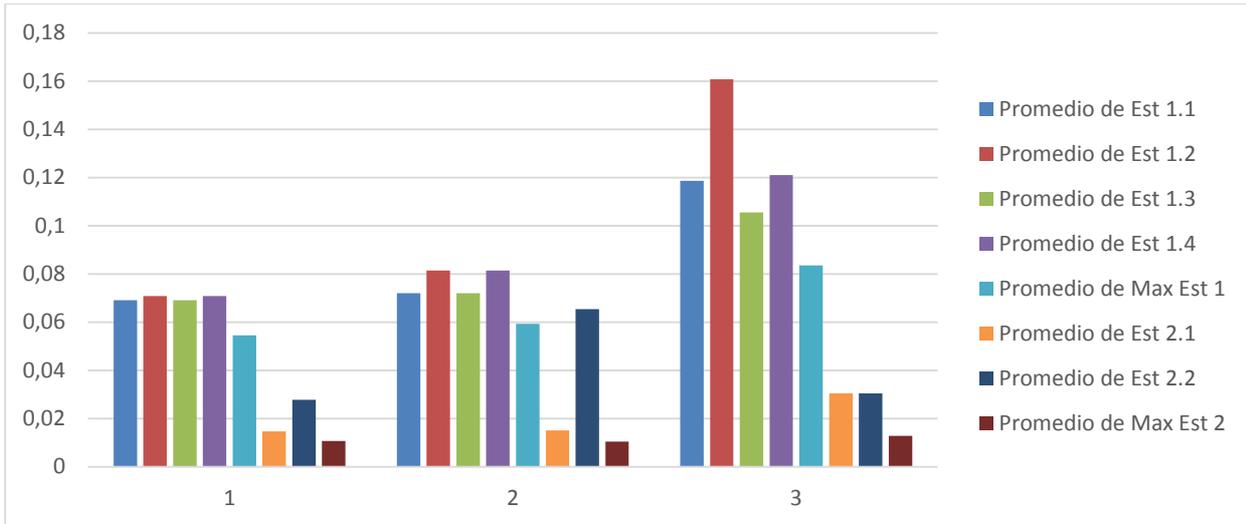


Figura 22 Comportamiento ante cada matriz de compatibilidad FSP

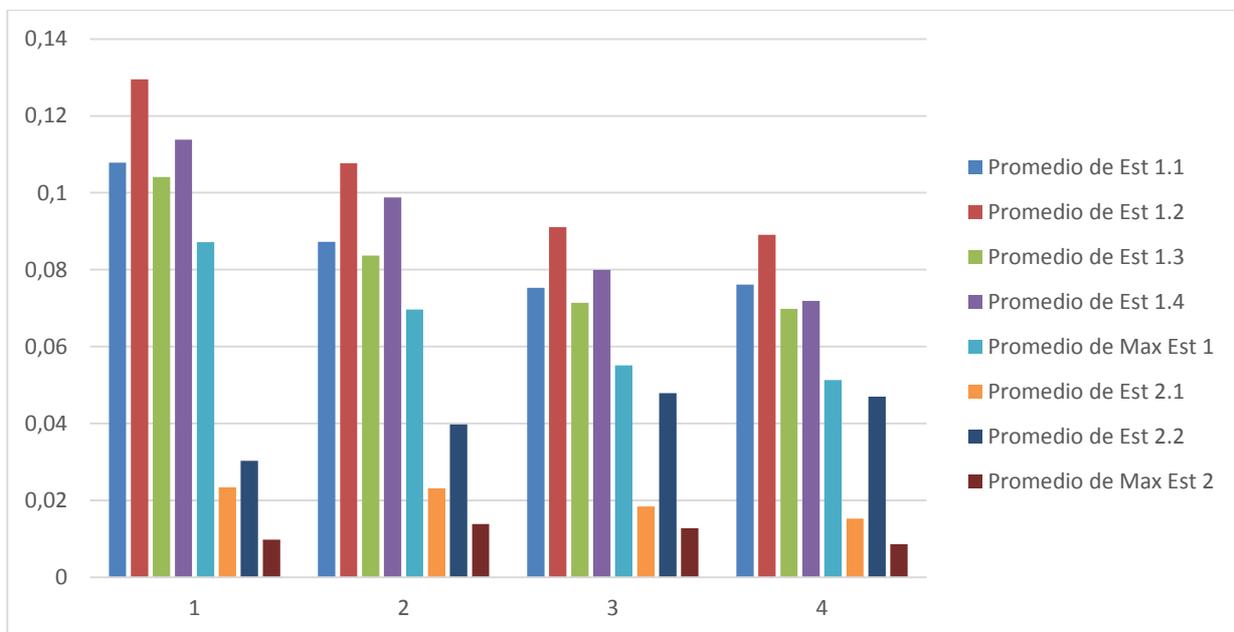


Figura 23 Comportamiento ante cada distribución de máquinas FSP

Antes de comenzar con la decisión sobre cuál de las estrategias responde mejor a la influencia de cada parámetro conviene observar las tendencias generales que se crean en todas las estrategias heurísticas frente a la variación de cada uno de los parámetros.

Como ya hemos introducido anteriormente, se muestran en los gráficos anteriores el efecto de cada parámetro en el promedio del error relativo en el que incurre cada estrategia de resolución.

6.1.1.1 Solapamiento

En primer lugar, se ha mostrado el efecto del solape, donde podemos apreciar fácilmente y como es lógico que cuanto mayor sea el valor del solape, es decir, cuanto más solapados se encuentran los trabajos del problema, las estrategias incurren en un error relativo mayor. Este resultado era de esperar ya que el grado de solape

está relacionado directamente con la complejidad del problema. Sin embargo, cabe apuntar que el incremento del promedio del error es menor de lo que se pudiera esperar. Llegando incluso a disminuir en la estrategia 2.2.

Solape	Est 1.1	Est 1.2	Est 1.3	Est 1.4	Max Est 1	Est 2.1	Est 2.2	Max Est 2
1	0,075	0,094	0,070	0,077	0,057	0,018	0,046	0,010
2	0,088	0,104	0,084	0,093	0,067	0,020	0,041	0,011
4	0,097	0,115	0,093	0,104	0,073	0,022	0,037	0,013

Tabla 1 Promedio error relativo por solape FSP

Como podemos observar en la tabla de valores, la estrategia 2.2 disminuye el error relativo al aumentar el solapamiento, sería objeto de estudio dicha disminución al enfrentarnos a un grado de solapamiento muy elevado, sin embargo, en los grados de solapamiento estudiados en nuestra batería de problemas, no llegamos a ver el cruce con el incremento de error de la estrategia 2.1.

Además, el solapamiento de nivel 4 usado en estos problemas es ya un solapamiento muy elevado por lo que pudiera pasar que nunca llegue a producirse ese cruce entre la estrategia 2.1 y la 2.2 en el que llegara a obtener mejores resultados la 2.2.

6.1.1.2 Número de máquinas y número de trabajos

El siguiente parámetro representado es el número de máquinas disponibles para el problema. En el número de máquinas, podemos observar que la tendencia general que describen las estrategias, denotada claramente por las estrategias de máximos es que cuanto mayor es el número de máquinas, mejores resultados obtenemos, lo cual es lógico ya que a pesar de que la resolución del problema sea más tediosa, para el programa será más fácil obtener la mejor solución debido a la mayor disponibilidad de soluciones.

De nuevo observamos que la tendencia de la estrategia 2.2 es contraria, dando lugar a mejores resultados cuando más complejo es el problema, es decir, cuando tenemos menos recursos máquina disponibles para asignarlos a los trabajos.

Num Maq	Est 1.1	Est 1.2	Est 1.3	Est 1.4	Max Est 1	Est 2.1	Est 2.2	Max Est 2
4	0,090	0,104	0,086	0,097	0,063	0,023	0,037	0,012
8	0,092	0,109	0,088	0,096	0,075	0,019	0,040	0,012
16	0,074	0,098	0,068	0,075	0,058	0,016	0,050	0,008

Tabla 2 Promedio error relativo por número de máquinas FSP

Mostramos después del efecto del número de máquinas, el efecto del número de trabajos, ya que hay una pequeña correlación entre ellos. Por la distribución de problemas que explicamos anteriormente, no tiene sentido hacer un estudio de problemas con bajo número de trabajos y gran número de máquinas por lo que no se han incluido en la batería de problemas propuesta a estudio. De esta forma, el promedio obtenido para los problemas con 16 máquinas está influenciado solamente por el promedio obtenido en los problemas de 200, 500 y 1000 trabajos. Este efecto se verá potenciado o contrarrestado por el efecto que tenga la variación del número de trabajos en la solución de cada estrategia.

Num	Est 1.1	Est 1.2	Est 1.3	Est 1.4	Max Est 1	Est 2.1	Est 2.2	Max Est 2
-----	---------	---------	---------	---------	-----------	---------	---------	-----------

Trab

50	0,085	0,104	0,082	0,098	0,036	0,031	0,045	0,009
100	0,085	0,107	0,080	0,094	0,054	0,029	0,047	0,016
200	0,084	0,106	0,078	0,088	0,061	0,023	0,047	0,014
500	0,089	0,107	0,085	0,093	0,075	0,017	0,042	0,011
1000	0,089	0,099	0,085	0,089	0,079	0,010	0,030	0,006

Tabla 3 Promedio error relativo por número de trabajos FSP

Respecto al número de trabajos, observamos tendencias cruzadas, donde podemos observar dos grandes grupos según el tipo de estrategia 1 o 2. Las estrategias tipo 1 no se ven muy afectadas por la variación en el número de problemas aunque si atendemos a la estrategia “Max Est 1” como control vemos que la tendencia general es a aumentar el error junto con el número de trabajos; sin embargo, las estrategias 2 tienen la tendencia contraria, siendo menor el error relativo cuanto mayor es el número de trabajos.

6.1.1.3 Tipo de matriz L y distribución de máquinas

El siguiente parámetro que vamos a estudiar, es la tendencia de las estrategias respecto a la matriz de compatibilidad L, antes de nada, recordar las matrices L que hemos utilizado.

$$L_1 = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix}$$

$$L_2 = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix}$$

$$L_3 = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{pmatrix}$$

En este punto es importante recordar también que el parámetro estudiado es el promedio de error relativo respecto al óptimo, ya que viendo las matrices de compatibilidad, cabe pensar que entre la matriz 2 y 3, la 3 proporcionará mejores resultados. Sin embargo, estos resultados son mejores en valor de la función objetivo, ya que la funcionalidad de las máquinas está repartida más equitativamente.

Siendo esto así, los resultados aquí reflejados, muestran que la tendencia general de las estrategias es conseguir resultados más cercanos al óptimo con la matriz de compatibilidad 1, por su simplicidad, error relativo medio en las matrices tipo 2 por la facilidad de priorizar las clases y peores resultados en la matriz de compatibilidad tipo 3 debido a la dificultad de definir una clase prioritaria entre ellas.

No obstante, podemos comprobar con los datos obtenidos, que el valor óptimo de la función objetivo es mayor en los problemas con matriz de compatibilidad tipo 1 y 3, que en los de tipo 2.

Tipo L	Est 1.1	Est 1.2	Est 1.3	Est 1.4	Max Est 1	Est 2.1	Est 2.2	Max Est 2
1	0,069	0,071	0,069	0,071	0,055	0,015	0,028	0,011
2	0,072	0,081	0,072	0,081	0,059	0,015	0,065	0,010
3	0,119	0,161	0,106	0,121	0,084	0,030	0,030	0,013

Tabla 4 Promedio error relativo por tipo matriz L FSP

Presentamos ahora los resultados obtenidos bajo el efecto de la distribución de máquinas en una clase u otra.

De igual manera, respecto al parámetro de distribución del número de máquinas en cada clase, viendo las matrices de compatibilidad, es seguro que los mejores resultados en la función objetivo la darán las distribuciones que aglomeren las máquinas entre las clases de máquinas 1 y 2 ya que en las matrices de compatibilidad 1 y 2, estas máquinas son más compatibles que las máquinas de clase 3.

Más adelante estudiaremos el efecto conjunto de la matriz de compatibilidad y la distribución de máquinas para ver esta diferencia.

Respecto al parámetro independiente de la distribución de las máquinas, hay una tendencia clara a obtener menor error conforme las máquinas se concentran en las últimas clases. Esta tendencia coincide con los problemas que obtienen menor valor de la función objetivo por la relación entre este parámetro y la matriz de compatibilidad.

Dist	Est 1.1	Est 1.2	Est 1.3	Est 1.4	Max Est 1	Est 2.1	Est 2.2	Max Est 2
Maq 1	0,108	0,130	0,104	0,114	0,087	0,023	0,030	0,010
2	0,087	0,108	0,084	0,099	0,070	0,023	0,040	0,014
3	0,075	0,091	0,071	0,080	0,055	0,018	0,048	0,013
4	0,076	0,089	0,070	0,072	0,051	0,015	0,047	0,009

Tabla 5 Promedio error relativo por distribución de máquinas FSP

6.1.2 Comparación entre estrategias

Visto el efecto de cada parámetro de forma general, pasamos ahora a comparar cuál de las estrategias heurísticas presenta mejores resultados.

Para ello vamos a presentar los datos anteriormente mostrados pero dando las distintas tonalidades teniendo en cuenta los valores de toda la tabla en vez de los de cada columna como hemos hecho anteriormente. Quedarán resaltadas en rojo las peores estrategias y en verde las mejores.

Solape	Est 1.1	Est 1.2	Est 1.3	Est 1.4	Max Est 1	Est 2.1	Est 2.2	Max Est 2
1	0,075	0,094	0,070	0,077	0,057	0,018	0,046	0,010
2	0,088	0,104	0,084	0,093	0,067	0,020	0,041	0,011
4	0,097	0,115	0,093	0,104	0,073	0,022	0,037	0,013

Num Trab	Est 1.1	Est 1.2	Est 1.3	Est 1.4	Max Est 1	Est 2.1	Est 2.2	Max Est 2
50	0,085	0,104	0,082	0,098	0,036	0,031	0,045	0,009
100	0,085	0,107	0,080	0,094	0,054	0,029	0,047	0,016
200	0,084	0,106	0,078	0,088	0,061	0,023	0,047	0,014
500	0,089	0,107	0,085	0,093	0,075	0,017	0,042	0,011
1000	0,089	0,099	0,085	0,089	0,079	0,010	0,030	0,006

Tipo L	Est 1.1	Est 1.2	Est 1.3	Est 1.4	Max Est 1	Est 2.1	Est 2.2	Max Est 2
--------	---------	---------	---------	---------	-----------	---------	---------	-----------

1	0,069	0,071	0,069	0,071	0,055	0,015	0,028	0,011
2	0,072	0,081	0,072	0,081	0,059	0,015	0,065	0,010
3	0,119	0,161	0,106	0,121	0,084	0,030	0,030	0,013
Num								
Maq	Est 1.1	Est 1.2	Est 1.3	Est 1.4	Max Est 1	Est 2.1	Est 2.2	Max Est 2
4	0,090	0,104	0,086	0,097	0,063	0,023	0,037	0,012
8	0,092	0,109	0,088	0,096	0,075	0,019	0,040	0,012
16	0,074	0,098	0,068	0,075	0,058	0,016	0,050	0,008
Dist Maq	Est 1.1	Est 1.2	Est 1.3	Est 1.4	Max Est 1	Est 2.1	Est 2.2	Max Est 2
1	0,108	0,130	0,104	0,114	0,087	0,023	0,030	0,010
2	0,087	0,108	0,084	0,099	0,070	0,023	0,040	0,014
3	0,075	0,091	0,071	0,080	0,055	0,018	0,048	0,013
4	0,076	0,089	0,070	0,072	0,051	0,015	0,047	0,009

Tabla 6 Promedio error relativo conjunto FSP

Viendo estos resultados es fácilmente identificable la estrategia 1.2 como la peor de ellas y la estrategia de control del tipo 2 como la mejor. Además, como comprobación podemos ver que ninguna de las estrategias es mejor que su respectivo control, lo que da validez a los datos.

Dentro de las estrategias tipo 2 aunque podemos ver que los resultados son similares en las estrategias 2.1 y 2.2, la estrategia tipo 2.1 siempre permanece con valores más cercanos al óptimo a pesar de que la estrategia 2.2 tiene tendencias que mejoran al aumentar ciertos parámetros nunca llega a producirse el momento en el que obtengamos mejores resultados con la estrategia 2.2 que con la 2.1.

Observando los datos obtenidos en tiempo medio de ejecución para llegar a estos resultados, la opción de escoger el camino de las estrategias de control se antoja inaceptable ya que en las estrategias tipo 1 el valor es excesivamente elevado y en las estrategias tipo 2, el tiempo de ejecución es cuatro veces mayor que en las ejecuciones con toma de decisión.

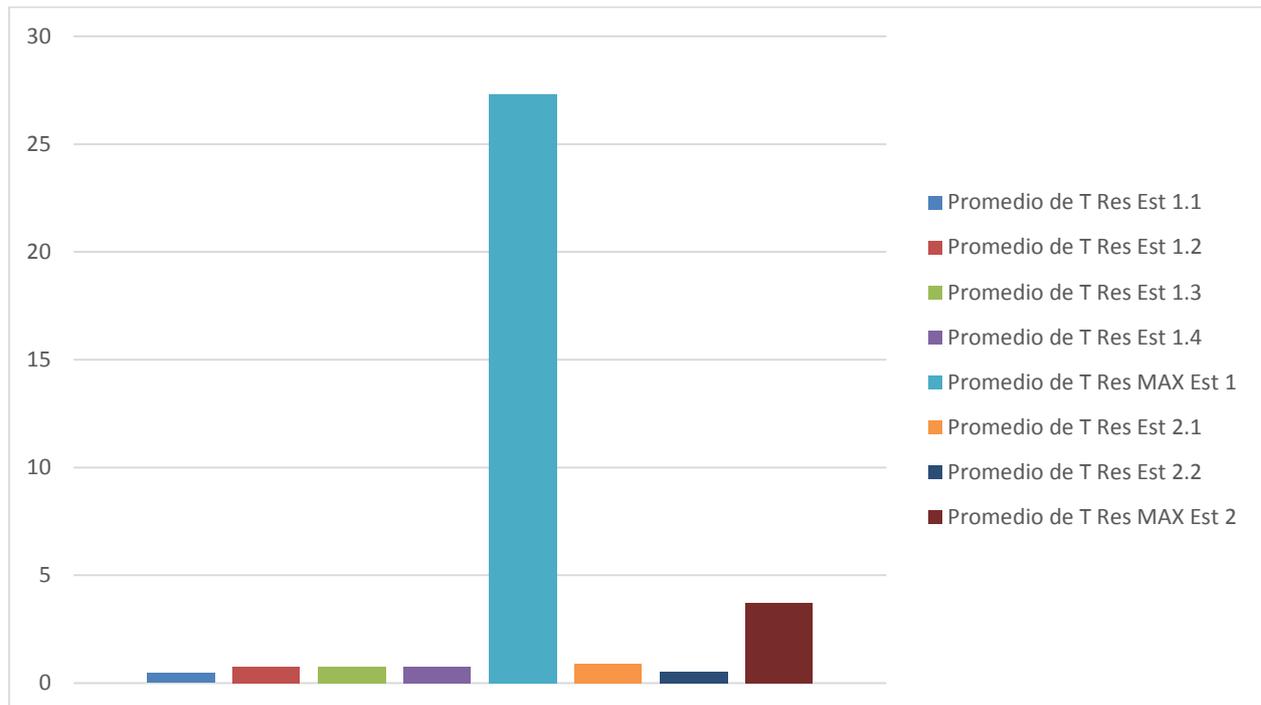


Figura 24 Tiempo medio simulación por estrategia FSP

Por ello, la estrategia que según los datos recogidos de forma independiente proporciona los mejores resultados será la estrategia 2.1.

Pasamos ahora a centrarnos en las estrategias tipo 2 y en concreto en la estrategia 2.1 para ver los efectos de cada parámetro con mayor detalle.

6.1.2.1 Efectos correlacionados de los parámetros para la estrategia 2.1

A continuación vamos a mostrar las gráficas de los resultados de algunos de los parámetros correlacionados. Para ello, veremos los valores del promedio del error relativo de la estrategia 2.1 y de su estrategia de control como referencia.

Antes de nada, cabe remarcar que los errores relativos de la estrategia 2.1 están siempre en un rango menor al 4% y muy cercanos al control por lo que aunque se analice el error que comete, los resultados son muy favorables.

En primer lugar, vamos a mostrar los efectos disgregados entre la matriz de compatibilidad del problema y la distribución de las máquinas (hacia la clase 1 o la clase 3, respectivamente).

Podemos observar, como ya vimos anteriormente que la matriz tipo 3 al ser la que tiene más repartido el grado de compatibilidad de las máquinas, es la que obtiene peores resultados en la estrategia heurística y por tanto la que mayores errores presenta. Además, podemos ver cómo el error es menor al hacer un reparto de máquinas mayor en la clase menos compatible (por ejemplo la clase 4 en la matriz L 2) lo cual muestra una buena respuesta por parte de la estrategia en situaciones desfavorables.

Por lo general, la tendencia de la estrategia 2.1 sigue la tendencia de la estrategia de control y se mantiene cercana por lo que está bien diseñada, el punto crítico donde se podría conseguir una mejora es al encontrar las compatibilidades de las máquinas muy repartidas (matriz L 3)

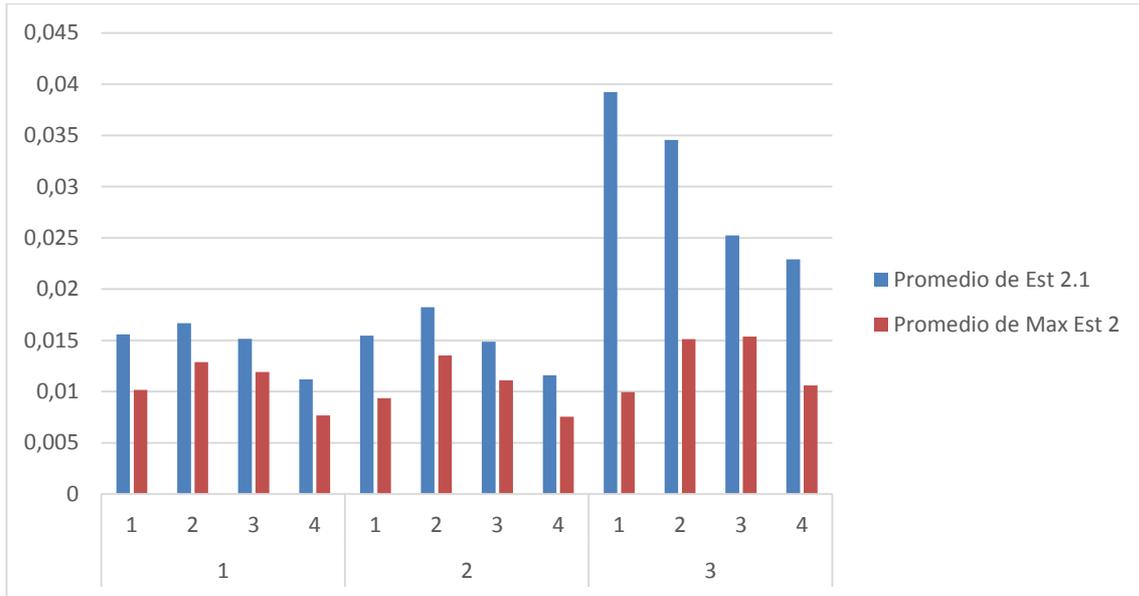


Figura 25 Comportamiento disgregado Matriz L / Distribución máquinas FSP

A continuación, observando los resultados disgregados respecto al número de trabajos y el número de máquinas disponibles, se observa la tendencia general anteriormente mostrada a disminuir el error mientras aumenta el número de máquinas y de trabajos.

Sin embargo, en la estrategia 2.1 al disgregar los efectos de cada parámetro podemos observar dos tendencias separadas. En los problemas con menos de 200 trabajos aumentar el número de máquinas es sinónimo de mejoría, sin embargo, al aumentar el número de máquinas en los problemas con gran número de trabajos, el error relativo aumenta. Este comportamiento es compartido por la estrategia 2.1 y por la estrategia de control por lo que es debido a las limitaciones inherentes al afrontar el problema de forma heurística y no de forma teórica.

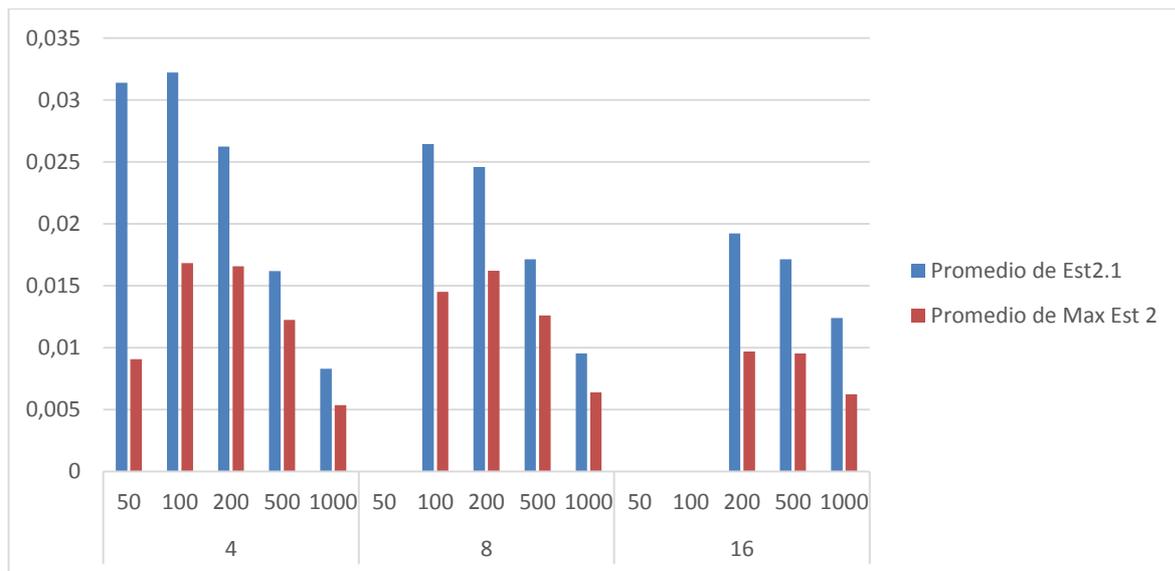


Figura 26 Comportamiento disgregado Número de Maquinas / Número de Trabajos FSP

Si atendemos al grado de solapamiento, los resultados son similares a los ya expuestos, el promedio general

de error tiende a aumentar conforme aumenta el solapamiento debido a su complejidad mientras que en cada grado de solapamiento, el error disminuye al aumentar el número de máquinas. Este efecto de disminución del error al aumentar el número de máquinas está relacionado con la no existencia de problemas con gran número de máquinas y bajo número de trabajos, ya que como vemos en la gráfica anterior, para bajo número de problemas el error es grande.

Visto estos resultados, se podrían tomar medidas para mejorar el error incurrido en los problemas de solape 1 y gran número de máquinas ya que hay una gran diferencia con el máximo posible marcado por su estrategia de control.

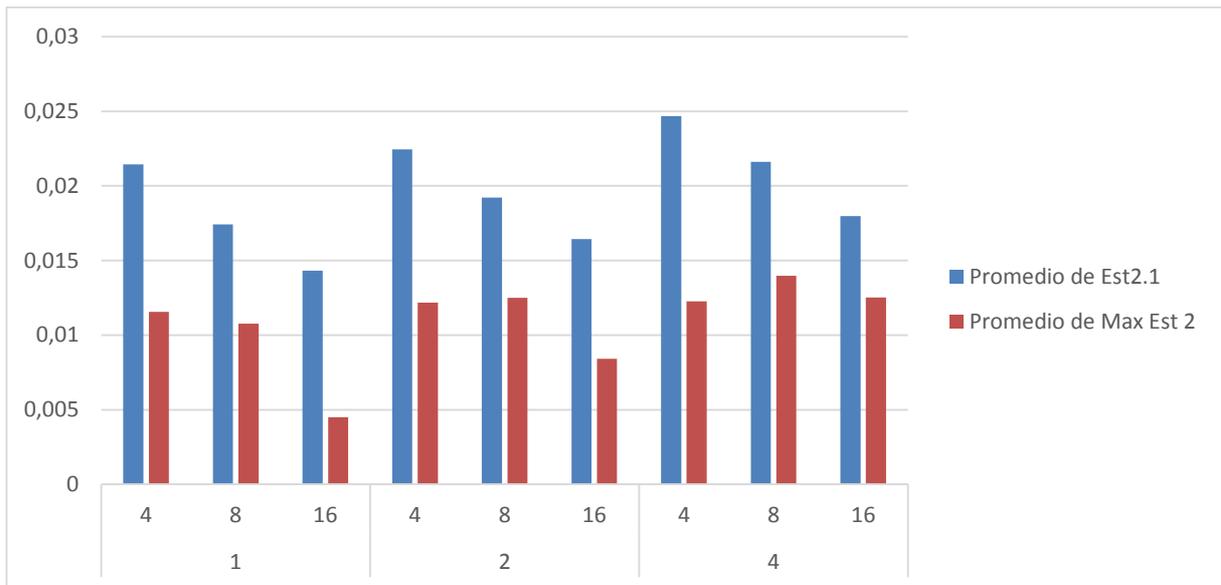


Figura 27 Comportamiento disgregado Solapamiento / Número de Máquinas FSP

6.2 VSP

A continuación, después de haber estudiado los resultados de las distintas estrategias en problemas del tipo FSP vamos a pasar a estudiar cómo se comportan en la modalidad de problema VSP.

Además de cómo se comporta la VSP de manera independiente, podremos estudiar también una comparación entre ambas.

La experimentación de problemas del tipo VSP siempre es más compleja por las limitaciones que ya hemos explicado más arriba. Lo que hace que las simulaciones tengan mayor duración y los efectos de cada parámetro más difíciles de apreciar.

Por otro lado, en la experimentación con los problemas VSP hemos tenido problemas al realizar las simulaciones de problemas complejos, en concreto, ha sido imposible conseguir las soluciones óptimas teóricas de los problemas con número de trabajos mayor a 100, por lo que el estudio que se va a mostrar a continuación será solo de los grupos de problemas de 50 y 100 trabajos. Esta limitación se ha visto solo afectada para la solución teórica, la solución heurística siempre ha sido posible llevarse a cabo gracias a su mayor simplicidad. Esta es una ventaja más por la que la resolución heurística de este tipo de problemas se ve como una buena alternativa.

Para la experimentación de los problemas VSP hemos utilizado la misma batería de problemas utilizada para los problemas FSP dejando una ventana temporal a cada problema del valor de ± 1 segundo de forma que aprovechando las simulaciones nos permita ilustrar el beneficio obtenido en nuestra función objetivo si conseguimos hacer que nuestro proceso nos permita abrir dicha ventana.

Como ya hemos dicho, en la práctica, los procesos suelen venir fijados con unos instantes de comienzo y fin inamovibles, sin embargo, buscar esta posibilidad puede reportar grandes mejoras.

Un buen punto a señalar, es la limitación impuesta de manera obligatoria a la hora de resolver problemas VSP de manera heurística con el algoritmo que estamos aquí probando. Esta limitación se observa en la diferencia innata entre los valores óptimos y los obtenidos a través de cada una de las estrategias que son a grandes rasgos mayores que las mismas diferencias en el caso de los problemas FSP, más adelante se mostrarán datos que respaldan este efecto de diferencia entre FSP y VSP.

Pasamos en primer lugar, en el mismo orden que hicimos con los problemas FSP a observar los valores del error relativo obtenido de la experimentación.

Comenzamos mostrando una visión general de los errores en los que incurre cada una de las estrategias respecto a los correspondientes óptimos.

$$\text{Error relativo} = \frac{\text{Óptimo} - \text{Resultado heurístico}}{\text{Óptimo}}$$

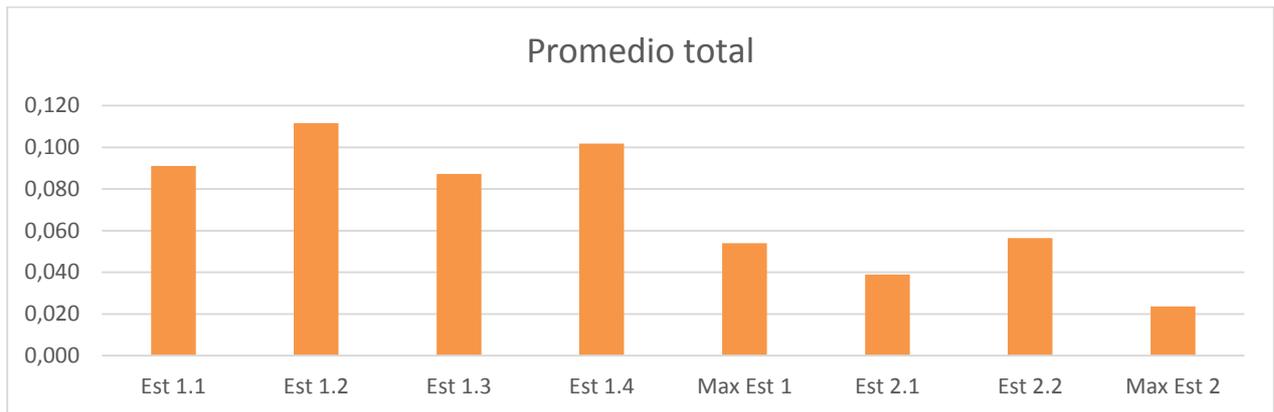


Figura 28 Promedio total de error relativo por estrategia VSP

6.2.1 Observación de tendencias generales en respuesta a cada parámetro en VSP

En esta visión general podemos ver que igual que sucedió con los problemas FSP, las estrategias tipo 1 obtienen peores resultados que las estrategias tipo 2. Sin embargo, la estrategia de control de tipo 1 supera a la estrategia tipo 2.2 aunque no llega a superar a la 2.1 ni a su respectivo control. Pasaremos a continuación al igual que hicimos en los FSP a mostrar los distintos gráficos que ilustran el comportamiento de cada estrategia respecto a la variación de los parámetros, para después centrarnos en la tendencia de la mejor de ellas.

El error relativo promedio de cada estrategia VSP respecto a los parámetros queda aquí reflejado:

- Solapamiento
- Número de máquinas
- Número de trabajos
- Tipo de matriz de compatibilidad
- Distribución de las máquinas

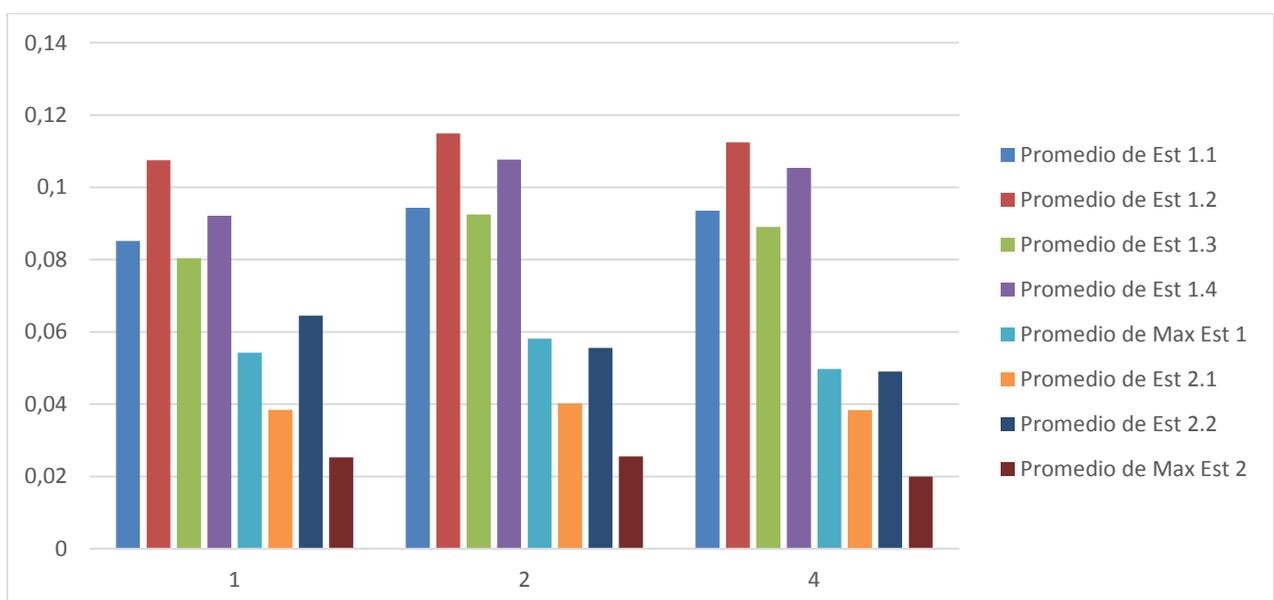


Figura 29 Comportamiento ante cada solapamiento VSP

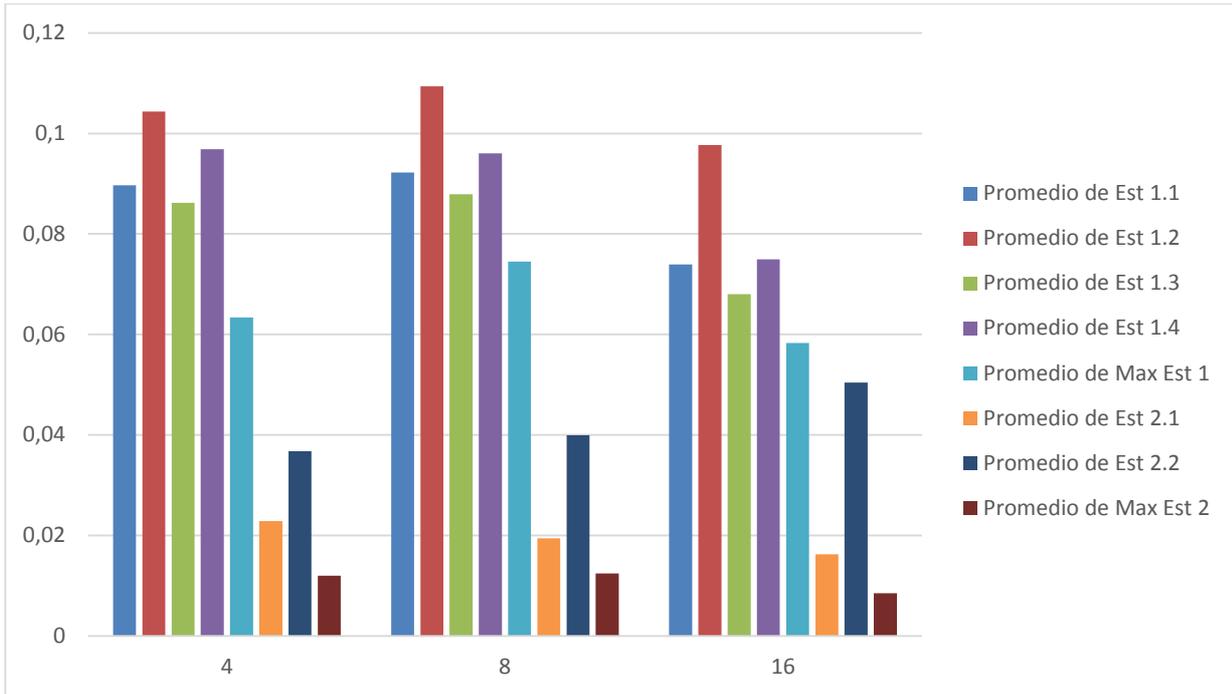


Figura 30 Comportamiento ante cada cantidad de máquinas VSP

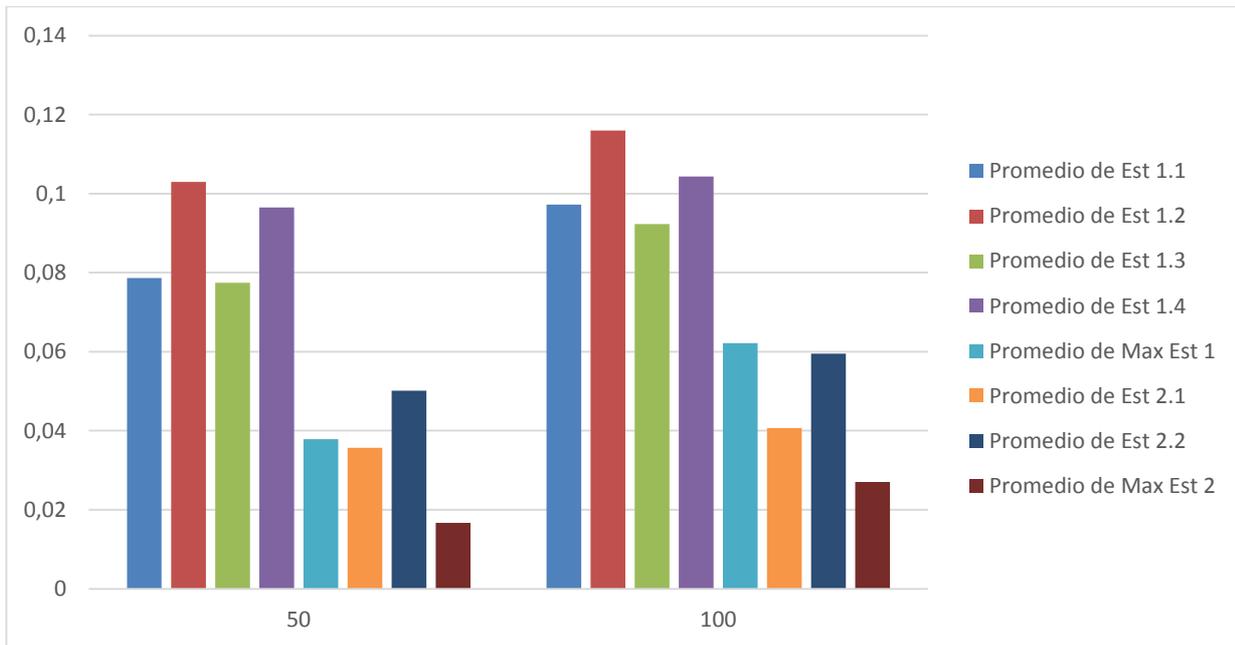


Figura 31 Comportamiento ante cada cantidad de trabajos VSP

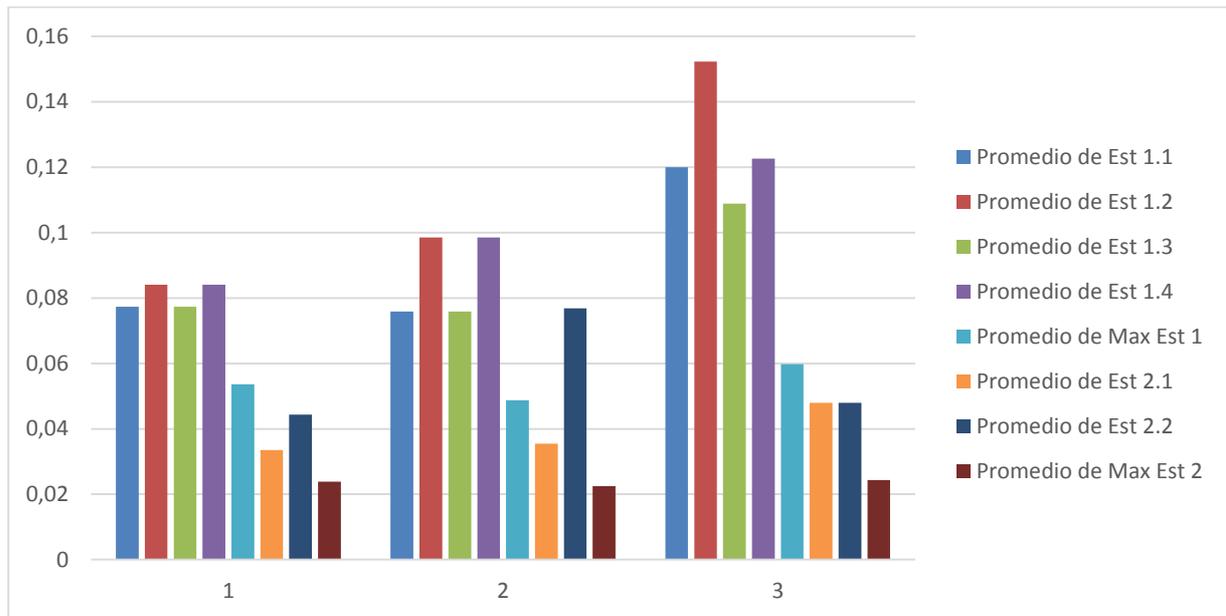


Figura 32 Comportamiento ante cada matriz de compatibilidad VSP

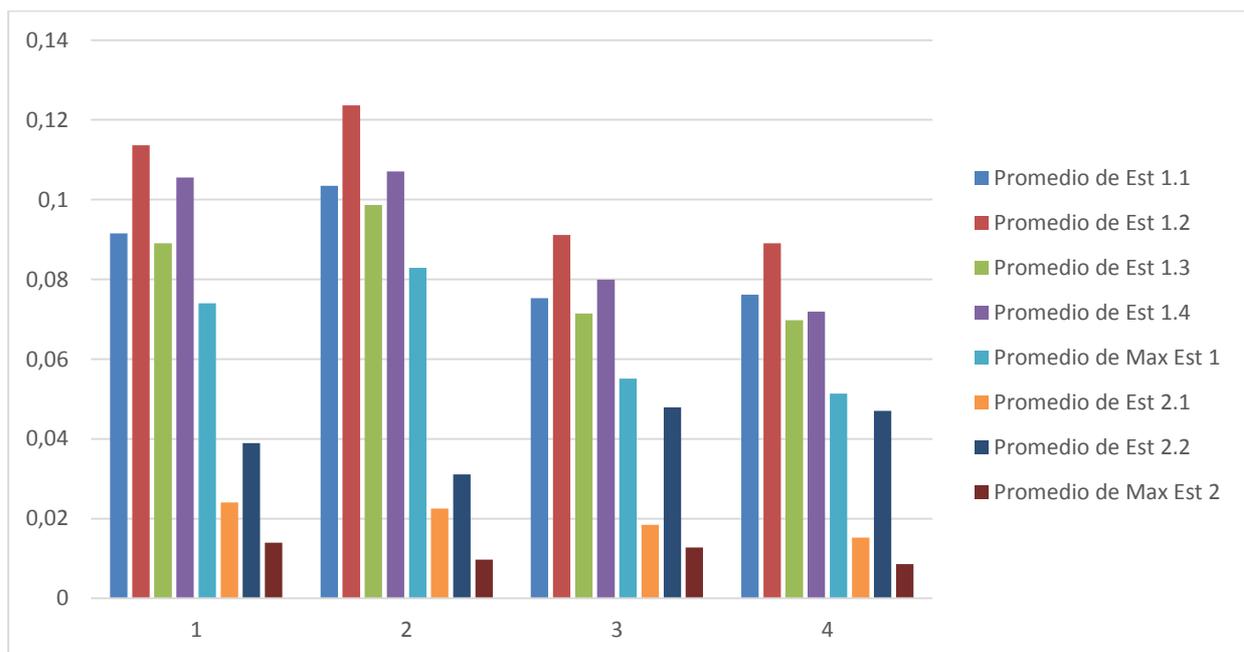


Figura 33 Comportamiento ante cada distribución de máquinas VSP

Al igual que hicimos para los problemas FSP vamos a analizar en mayor profundidad el efecto de cada uno de los parámetros individualmente, para ello, prestamos atención a las cifras de error relativo que producen los gráficos anteriormente mostrados.

A diferencia de los datos de los problemas tipo FSP, estos datos resultan ser menos representativos ya que solo han podido ser comparados con los correspondientes a los problemas de 50 y 100 trabajos. Esto tiene como consecuencia una disminución importante del número de problemas, pasando de 4320 a 1800.

En primer lugar, respecto al solape, podemos apreciar que al contrario de lo visto en los problemas anteriores, el efecto del solape no crea grandes variaciones en el error producido por las estrategias al aumentarlo. Sí

podemos observar sin embargo, las distintas tendencias entre las estrategias 1 y las estrategias 2, que focalizan los mejores resultados en los problemas con bajo grado de solapamiento y con gran grado de solapamiento respectivamente.

Además, es curioso observar como en este caso, a pesar de ser por una diferencia pequeña, los errores más grandes se producen en los problemas con solape de grado 2.

El solape es un parámetro que crea diferencia entre las distintas estrategias como veremos más adelante pero no crea tanta diferencia al aumentar dentro de cada una de ellas.

Solape	Est 1.1	Est 1.2	Est 1.3	Est 1.4	Max Est 1	Est 2.1	Est 2.2	Max Est 2
1	0,085	0,108	0,080	0,092	0,054	0,038	0,065	0,025
2	0,094	0,115	0,092	0,108	0,058	0,040	0,056	0,026
4	0,094	0,112	0,089	0,105	0,050	0,038	0,049	0,020

Tabla 7 Promedio error relativo por solape VSP

Siguiendo con el orden establecido en las estrategias FSP, pasamos a observar los resultados respecto al número de máquinas, donde como vemos solo se han podido estudiar los grupos de 4 y 8 máquinas. En este tipo de problemas vemos como la tendencia es bastante clara a obtener un peor resultado cuanto mayor es el número de máquinas.

Este efecto podemos predecir que tendrá una tendencia a aumentar al crecer el número de máquinas por el efecto que describimos anteriormente que se produce al tener que crear problemas flujo a coste mínimo máquina a máquina en los problemas VSP.

Num Maq	Est 1.1	Est 1.2	Est 1.3	Est 1.4	Max Est 1	Est 2.1	Est 2.2	Max Est 2
4	0,090	0,110	0,087	0,104	0,050	0,039	0,054	0,021
8	0,094	0,115	0,087	0,098	0,062	0,039	0,062	0,029

Tabla 8 Promedio error relativo por número de máquinas VSP

El siguiente parámetro a estudio es el número de trabajos de los problemas donde podemos observar una tendencia clara común en la que los problemas con mayor número de trabajos dan como resultado peores respuestas por todas las estrategias heurísticas. En esta ocasión, la diferencia que se crea sí resulta ser significativa.

Num Trab	Est 1.1	Est 1.2	Est 1.3	Est 1.4	Max Est 1	Est 2.1	Est 2.2	Max Est 2
50	0,079	0,103	0,077	0,096	0,038	0,036	0,050	0,017
100	0,097	0,116	0,092	0,104	0,062	0,041	0,060	0,027

Tabla 9 Promedio error relativo por número de trabajos VSP

Respecto a las diferentes matrices de compatibilidad, al igual que ocurría con los problemas del tipo FSP, la matriz de compatibilidad tipo 3 obtiene peores resultados en comparación con el óptimo teórico debido a que

es la matriz que presenta la compatibilidad de máquinas y trabajos más repartida, lo cual, a pesar de obtener mejores resultados en la función objetivo, resulta perjudicial en las estrategias heurísticas a la hora de decidir la prioridad de aplicación de las máquinas.

Tipo								
L	Est 1.1	Est 1.2	Est 1.3	Est 1.4	Max Est 1	Est 2.1	Est 2.2	Max Est 2
1	0,077	0,084	0,077	0,084	0,054	0,034	0,044	0,024
2	0,076	0,099	0,076	0,099	0,049	0,035	0,077	0,023
3	0,120	0,152	0,109	0,123	0,060	0,048	0,048	0,024

Tabla 10 Promedio error relativo por tipo matriz L VSP

Finalmente, la distribución de las máquinas al igual que en los problemas FSP provoca peores resultados al estar más centrada en las clases de máquinas más compatibles excepto en la estrategia del tipo 2.2 ya que la lógica de esta estrategia como ya describimos es priorizar las máquinas más compatibles. No obstante, como veremos a continuación, a pesar de tener la tendencia contraria, sigue obteniendo peores resultados que la estrategia 2.1 en todas las distribuciones posibles.

Dist								
Maq	Est 1.1	Est 1.2	Est 1.3	Est 1.4	Max Est 1	Est 2.1	Est 2.2	Max Est 2
1	0,098	0,127	0,094	0,113	0,062	0,044	0,053	0,025
2	0,097	0,119	0,093	0,106	0,059	0,042	0,053	0,026
3	0,086	0,102	0,082	0,098	0,050	0,037	0,061	0,024
4	0,083	0,099	0,080	0,090	0,045	0,033	0,059	0,020

Tabla 11 Promedio error relativo por distribución de máquinas VSP

6.2.2 Comparación entre estrategias

Una vez mostradas de manera independiente las tendencias que se crean en cada una de las estrategias al variar los parámetros de estudio, pasamos al igual que hicimos en los problemas VSP a la decisión de cuál de las estrategias obtiene mejores resultados.

Para ello mostraremos tanto las tablas de errores respecto a cada parámetro como los tiempos empleados en obtener estos resultados.

Solape	Est 1.1	Est 1.2	Est 1.3	Est 1.4	Max Est 1	Est 2.1	Est 2.2	Max Est 2
1	0,085	0,108	0,080	0,092	0,054	0,038	0,065	0,025
2	0,094	0,115	0,092	0,108	0,058	0,040	0,056	0,026
4	0,094	0,112	0,089	0,105	0,050	0,038	0,049	0,020

Num Trab	Est 1.1	Est 1.2	Est 1.3	Est 1.4	Max Est 1	Est 2.1	Est 2.2	Max Est 2
50	0,079	0,103	0,077	0,096	0,038	0,036	0,050	0,017
100	0,097	0,116	0,092	0,104	0,062	0,041	0,060	0,027

Tipo L	Est 1.1	Est 1.2	Est 1.3	Est 1.4	Max Est 1	Est 2.1	Est 2.2	Max Est 2
1	0,077	0,084	0,077	0,084	0,054	0,034	0,044	0,024
2	0,076	0,099	0,076	0,099	0,049	0,035	0,077	0,023
3	0,120	0,152	0,109	0,123	0,060	0,048	0,048	0,024

Num Maq	Est 1.1	Est 1.2	Est 1.3	Est 1.4	Max Est 1	Est 2.1	Est 2.2	Max Est 2
4	0,090	0,110	0,087	0,104	0,050	0,039	0,054	0,021
8	0,094	0,115	0,087	0,098	0,062	0,039	0,062	0,029
16								

Dist Maq	Est 1.1	Est 1.2	Est 1.3	Est 1.4	Max Est 1	Est 2.1	Est 2.2	Max Est 2
1	0,098	0,127	0,094	0,113	0,062	0,044	0,053	0,025
2	0,097	0,119	0,093	0,106	0,059	0,042	0,053	0,026
3	0,086	0,102	0,082	0,098	0,050	0,037	0,061	0,024
4	0,083	0,099	0,080	0,090	0,045	0,033	0,059	0,020

Tabla 12 Promedio error relativo conjunto VSP

Como podemos observar fácilmente, los mejores resultados para cada parámetro son obtenidos siempre por la estrategia de control tipo 2 seguida por la estrategia tipo 2.1. Solo existe un parámetro en el que los errores relativos de las estrategias 2.1 y 2.2 se igualan, este caso se da en los problemas con matriz tipo L = 3.

Por último mostramos los tiempos promedio invertidos en obtener estos resultados, donde no queda duda que la estrategia elegida que obtiene mejores resultados es la 2.1 al igual que en los problemas del tipo FSP.

Además, podemos observar como los tiempos de resolución son mayores debido a la limitación ya mencionada de tener que inyectar en los grafos de flujo coste mínimo las máquinas de una en una. Los tiempos son mucho mayores que los obtenidos en los problemas tipo FSP incluso

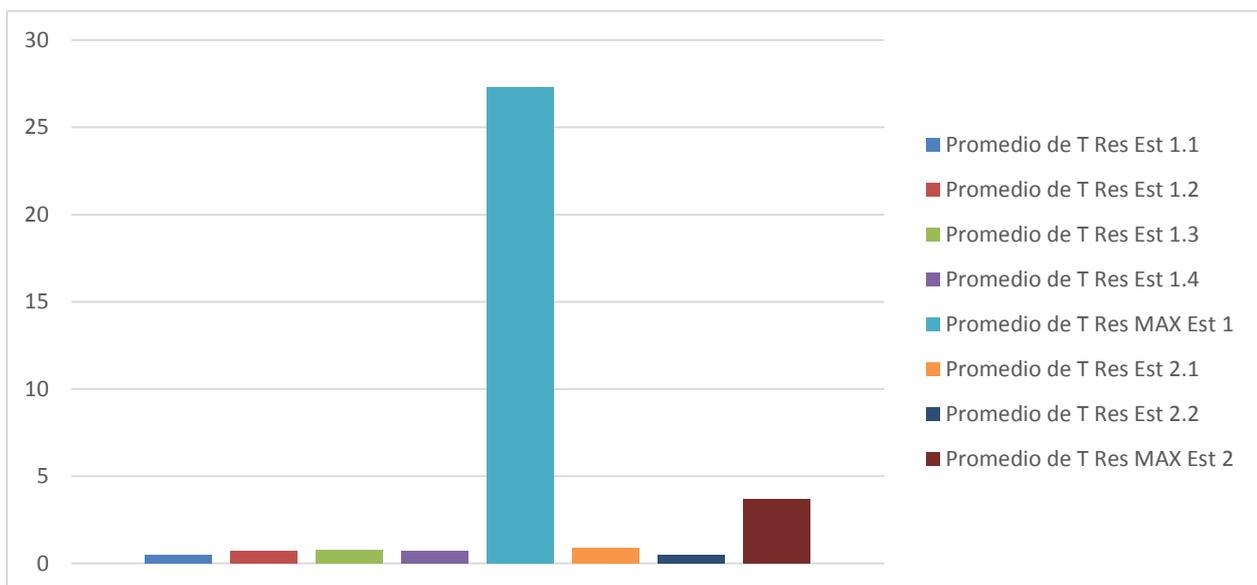


Figura 34 Tiempo medio simulación por estrategia VSP

Pasaremos a continuación a mostrar los efectos disgregados de los parámetros que mostramos para el caso FSP.

En primer lugar, para seguir con el mismo orden que en los problemas FSP vamos a observar la tendencia de la estrategia 2.1 frente a su estrategia de control respecto a la variación del tipo de matriz L y la distribución del número de máquinas.

Como podemos observar, la tendencia es parecida a la que veíamos en los problemas FSP, siendo la matriz tipo 3 la que obtiene mayor error relativo frente al óptimo. Esta tendencia coincide con los problemas en los que se obtiene mayor valor de la función objetivo, ya que la matriz 3 es la que presenta la compatibilidad de las máquinas más repartidas.

Además, podemos ver que en cada tipo de matriz, al distribuir las máquinas hacia la clase menos compatible, el error relativo disminuye, por lo que nuestras estrategias heurísticas obtienen resultados más cercanos al óptimo cuando la distribución de máquinas es menos propicia. Recaltar de nuevo que no debemos confundir la cercanía al óptimo que se representa en estas graficas con el valor de función objetivo conseguido en cada problema.

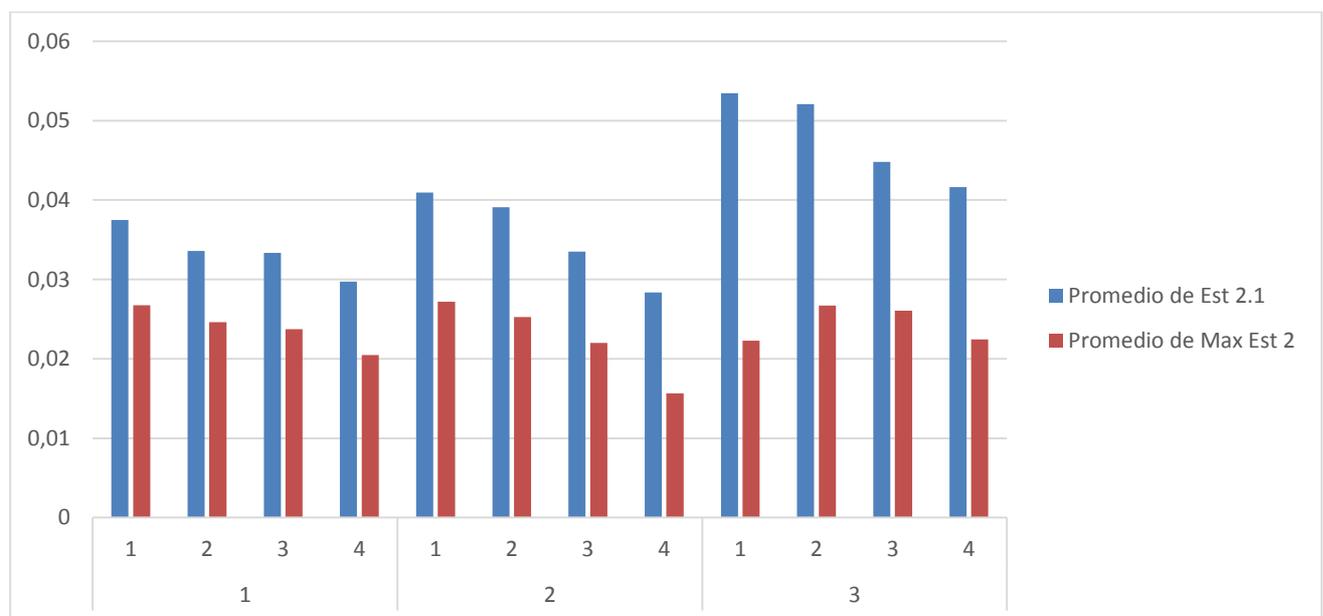


Figura 35 Comportamiento disgregado Tipo matriz L / Distribución de Máquinas VSP

El siguiente gráfico de estudio muestra los efectos cruzado del número de máquinas y el número de problemas, sin embargo, como podemos ver, en esta ocasión los datos no son representativos ya que estos dos parámetros no han permitido obtener suficientes datos.

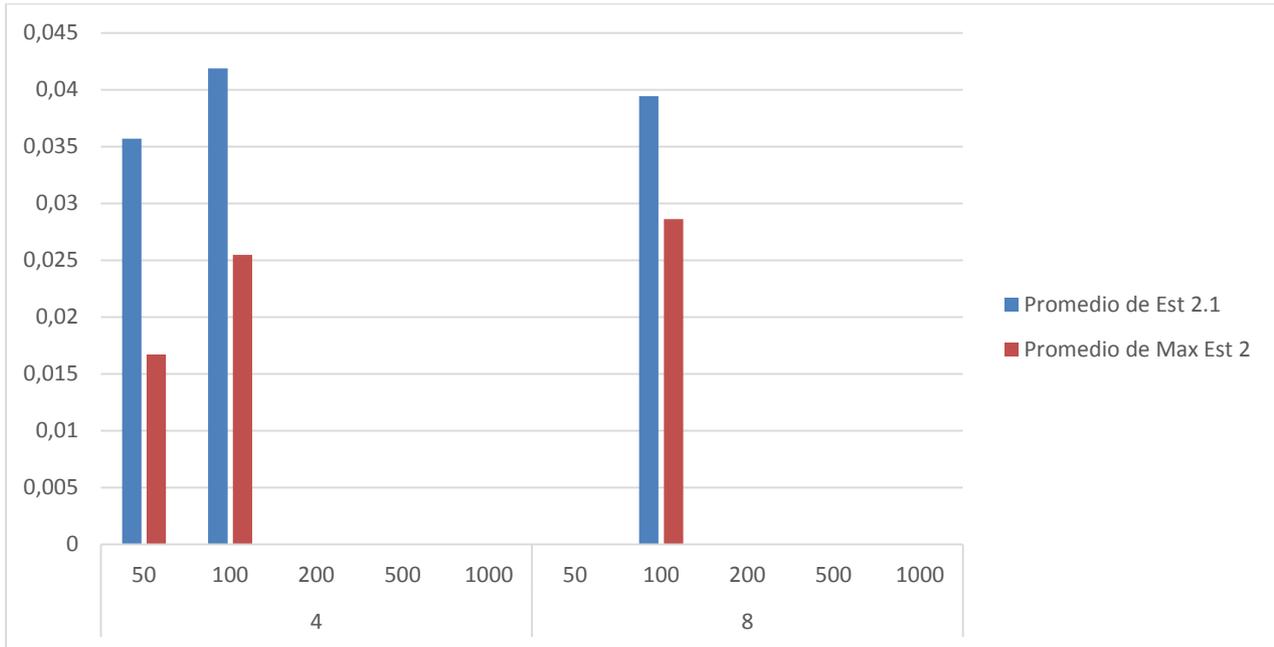


Figura 36 Comportamiento disgregado Número de máquinas / Número de trabajos VSP

Nos centramos entonces en los efectos del tipo de matriz de compatibilidad, solapamiento y distribución de las máquinas para el conjunto de problemas en los que sí se ha podido obtener el óptimo.

Dentro de los parámetros que sí podemos estudiar, vemos una tendencia clara disgregando el solapamiento y las matrices de compatibilidad, donde la matriz de compatibilidad donde es menos efectiva la estrategia heurística es la 3 pero cuanto mayor es el solapamiento su efecto mejora.

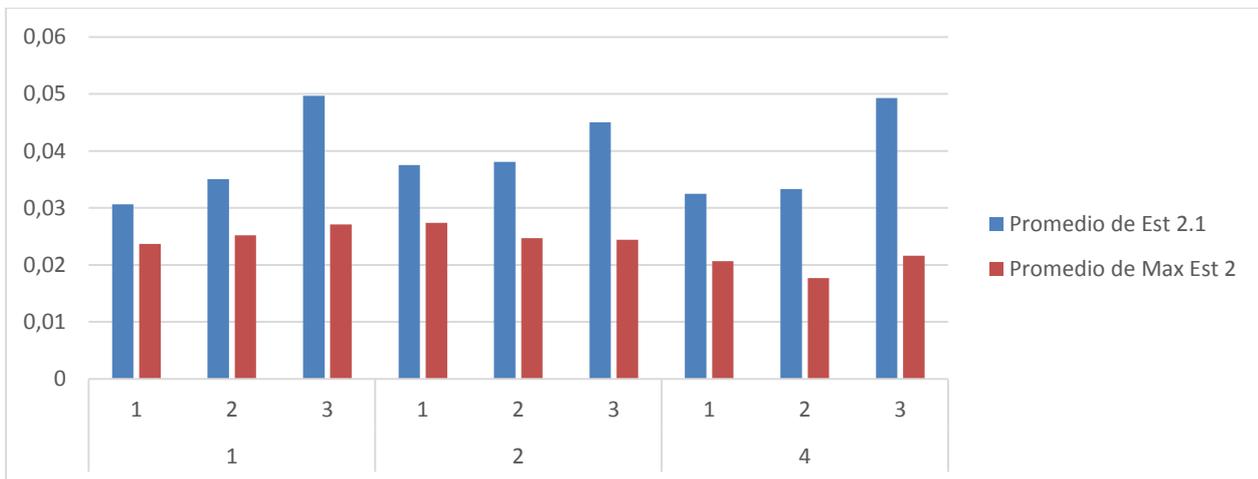


Figura 37 Comportamiento disgregado Solapamiento / Tipo de matriz L VSP

Finalmente, para completar el efecto de la distribución de las máquinas ya mostrado en conjunto con el tipo de matriz L, mostramos ahora el efecto agrupado por nivel de solapamiento para comprobar que el efecto es similar al ya mostrado pero en esta ocasión podemos ver pequeñas variaciones en el error relativo entre cada grado de solapamiento por lo que se cerciora que los efectos no están relacionados.

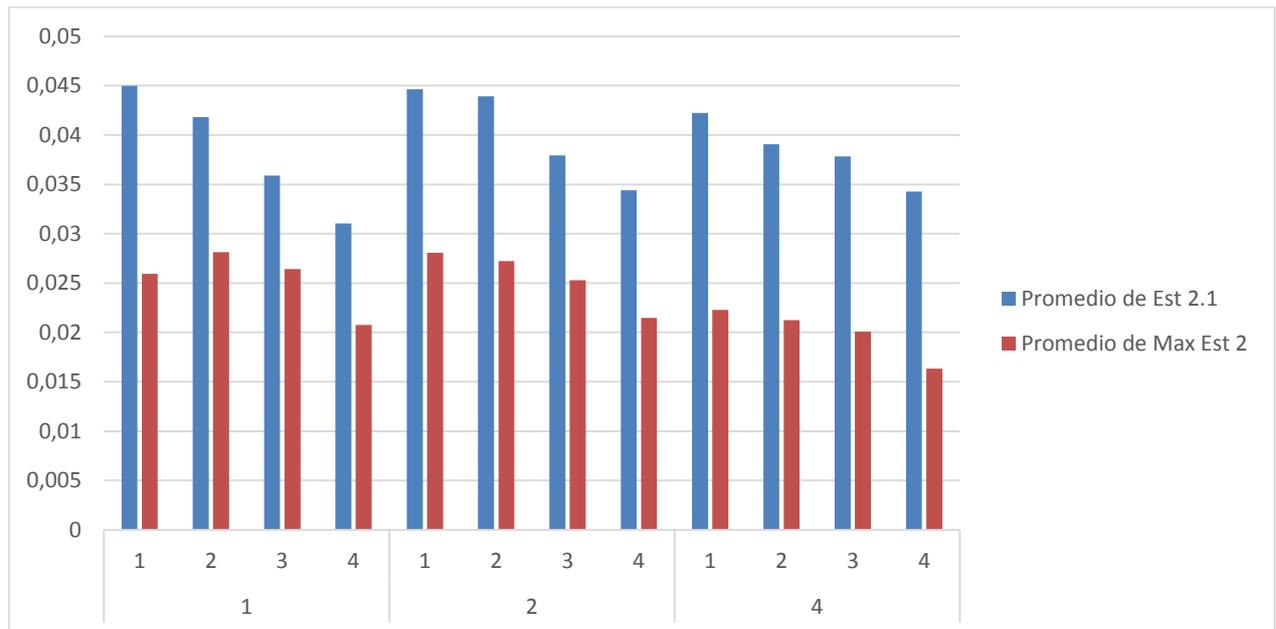


Figura 38 Comportamiento disgregado Solapamiento / Distribución máquinas VSP

6.3 Comparacion entre FSP y VSP

En este proyecto hemos presentado hasta ahora los resultados obtenidos resolviendo una batería de problemas siguiendo diferentes estrategias heurísticas y comparándolas con sus respectivos óptimos teóricos. Además, en primer lugar se resolvieron considerando los problemas FSP para después, permitir el desplazamiento de los trabajos con flexibilidad de un segundo para considerarlos VSP.

De esta manera, pasamos ahora a evaluar y estudiar la mejora que hemos conseguido en los resultados obtenidos de la batería si se diera la posibilidad de flexibilizar los trabajos como hemos indicado.

Para ello, en primer lugar, vamos a estudiar cómo se han incrementado los óptimos teóricos en el planteamiento VSP respecto al FSP; de forma general podemos observar que el promedio de mejora de los óptimos es del 7.02%, lo cual es una gran mejoría.

Además, vamos a mostrar cómo ha mejorado tanto el óptimo como cada una de las estrategias en cada uno de los grupos de problemas. En la siguiente tabla se muestra el % de crecimiento del valor de la función objetivo en los problemas VSP frente los FSP, siguiendo la siguiente fórmula:

$$\Delta = \frac{\text{Valor VSP} - \text{Valor FSP}}{\text{Valor FSP}}$$

Δ	Est 1.1	Est 1.2	Est 1.3	Est 1.4	Max Est1	Est 2.1	Est 2.2	Max Est2	Óptimos
Total	6,4%	6,5%	6,4%	6,4%	6,4%	6,0%	5,9%	5,9%	7,0%

Tabla 13 Incremento valor función objetivo de FSP a VSP

En esta tabla podemos comprobar la mejora que se produce en el valor final tanto en los óptimos como en las estrategias en valor absoluto de la función objetivo. Vemos que la mejora es muy significativa aplicando una flexibilidad pequeña, permitiendo tan solo la variación del instante de inicio un segundo antes o después.

Además, podemos observar los problemas que son más susceptibles y los que menos a esta mejora. Hay parámetros como el solape y el número de trabajos que marcan claramente la mejora posible al estar los trabajos más dispersos permiten aumentar de manera muy clara el resultado final. Mientras que el tipo de matriz de compatibilidad no tiene ningún efecto y la distribución de las máquinas tampoco.

No obstante, el parámetro con el que vamos a ser capaces de observar el cambio de comportamiento de nuestras estrategias al pasar de FSP a VSP es el error relativo, ya que deberemos comparar cómo se comporta cada estrategia respecto a su correspondiente óptimo en FSP y VSP.

Para ello presentamos la siguiente tabla donde mostramos la diferencia de errores relativos

$$\Delta = \text{Error relativo VSP} - \text{Error relativo FSP}$$

Δ	Est 1.1	Est 1.2	Est 1.3	Est 1.4	Max Est1	Est 2.1	Est 2.2	Max Est2
Total	0,6%	0,5%	0,7%	0,7%	0,6%	0,9%	1,0%	1,0%

Tabla 14 Incremento error relativo de FSP a VSP

Finalmente, en esta tabla podemos observar claramente que los errores relativos en los problemas VSP son en lo general un 1% mayores en las estrategias heurísticas que en los FSP. Queda así mostrada la influencia del efecto ya nombrado anteriormente sobre el efecto negativo que se crea al no poder inyectar directamente todas las máquinas disponibles de la clase en el problema flujo coste mínimo.

Sin embargo, debemos tener en cuenta que la complejidad del problema en los tipo VSP también será mayor por lo que solo es atribuible cierta parte del incremento del error a dicho efecto.

7 CONCLUSIONES Y LÍNEAS FUTURAS

Para concluir, mostraremos aquí las principales conclusiones obtenidas del análisis de resultados.

En primer lugar, ha quedado demostrado que las estrategias tipo 1 muestran peor comportamiento respecto a las tipo 2, siendo en FSP la media de los errores relativos de las primeras siempre mayores del 7% mientras que en el caso de la 2.1 se encuentra en el 2%. Esto demuestra que basar la decisión de la estrategia heurística en la compatibilidad de cada trabajo queda completamente contraindicado frente a basar dicha estrategia en la compatibilidad de cada recurso.

Además, dentro de basar la estrategia a gran nivel en la compatibilidad de las máquinas, ha quedado demostrado también que la mejor de las estrategias es la denominada 2.1. Esto implica que mejores resultados si a la hora de realizar la asignación de las máquinas, priorizamos en primer lugar las máquinas menos compatibles reservando el uso de las máquinas más compatibles.

Las estrategias heurísticas presentan sistemáticamente un error relativo respecto al óptimo debido a la imposibilidad de realizar el estudio de asignación con el conjunto total de trabajos y máquinas, siempre debiendo hacerse un problema flujo coste mínimo por cada clase de máquina. Este error ha quedado patente con las estrategias de control que siempre han mostrado el menor error relativo al que puede optar cada una de las estrategias, limitadas por este efecto. Este error relativo tiene como promedio en las estrategias tipo 2 un 1.13%. Siendo así, el promedio obtenido por la estrategia heurística del 2% es considerado un buen resultado.

Observando las tendencias de la mejor de las estrategias FSP, podemos concluir también que existe rango de mejora en los problemas con bajo número de trabajos y en los problemas en los que la matriz de compatibilidad reparte de manera muy equitativa las relaciones máquina trabajo como es la matriz 3.

Por otro lado, al realizar las simulaciones VSP, la primera conclusión que debemos mostrar es la imposibilidad de obtener los óptimos teóricos mediante el programa LINGO de forma automatizada debido a requisitos de hardware, mientras que los valores máximos de las estrategias heurísticas han sido fácilmente alcanzables con un ordenador personal común.

Con los resultados óptimos que sí hemos podido alcanzar, podemos concluir que las tendencias son muy similares a las observadas en los problemas FSP respecto a la clara diferencia entre las estrategias 1 y 2. Siendo la misma estrategia la que mejores resultados obtiene en los problemas FSP y VSP. Además, el potencial de mejora en los problemas con matriz de compatibilidad repartida equitativamente también ha quedado patente.

Respecto a la relación encontrada entre los problemas VSP y FSP ha quedado reflejada la gran mejora que podemos obtener en el entorno del 7%. El mayor rango de mejora observado al permitir una ventana temporal de ± 1 segundo de desplazamiento pasando así los problemas a VSP ha sido percibido como se esperaba en los problemas con bajos niveles de solapamiento y gran número de trabajos.

Además, respecto a la tendencia observada en los errores relativos de cada estrategia al pasar de FSP a VSP,

podemos observar como la tendencia general es de un aumento del 1% en el error relativo. Este 1% es respaldado entre otros efectos por el error en el que debemos incurrir al solucionar la clase de problemas VSP de manera heurística ya que no podemos inyectar en el grafo del problema flujo coste mínimo todas las máquinas disponibles de la clase bajo estudio como hemos explicado con anterioridad.

Con los resultados obtenidos en este proyecto, se abren líneas de trabajo por las que seguir estudiando el comportamiento de este tipo de problemas ante la resolución heurística. Entre las cuales cabría nombrar las siguientes:

- Mejorar la estrategia que mejores resultados ha obtenido para que resulte también aceptable en los problemas en los que ha aparecido con resultados más débiles.
- Obtener la manera de resolver la batería de problemas de forma exacta para tener los óptimos con los que comparar el resto de resultados heurísticos.
- Estudiar el efecto al crear una ventana temporal mayor en los problemas VSP.
- Estudiar el efecto al crear ventanas temporales variables en función de la duración o el peso del trabajo.
- Análisis de tiempos ociosos de las máquinas para evitar desaprovechamientos (Lean manufacturing).

Cabe destacar que los programas desarrollados en Visual Basic para las simulaciones de este proyecto se han desarrollado con vistas a facilitar su uso para futuras resoluciones haciendo el código fácilmente adaptable y genérico para distintas baterías de problemas.

8 BIBLIOGRAFÍA

- Visual Basic
Autor: José Eduardo Maluf de Cervalho
Editorial: McGraw-Hill
- Documentación de la asignatura Secuenciación
Profesor: José Manuel García Sánchez
Universidad de Sevilla
- Ingeniería de Organización: Modelos y aplicaciones
Pablo Cortés Achedad, Jose Manuel García Sanchez, José Guadix Martín.,
Universidad de Sevilla

9.1 Código estrategia 2.1 FSP

Se adjunta en este punto de anexo el código completo utilizado por la estrategia 2.1 para que sirva a la vez como ejemplo de programa en secuenciación FSP y como muestra de la mejor de las estrategias heurísticas FSP

```
Attribute VB_Name = "FSPNoclasesOperacional"
'ESTRATEGIA 2.1

Option Explicit
Option Base 1
Type DatosJobs
    si As Long
    fi As Integer
    ti As Integer
    wi As Integer
    di As Integer 'añadido para especificar la clase de trabajo
    ci As Integer 'añadido para almacenar la clase de máquina que ha resuelto este trabajo, útil para analizar los resultados en estrategia 2
    li As Integer 'vector de ordenacion
    mi As Integer 'añadido para asignar un nuevo indice de identificación en la estructura trabajosM que le quiero pasar a GenerarModelo
    xi As Integer 'Indice que me dira si el trabajo esta realizado o no 1 realizado 0 no realizado
End Type
Dim Trabajos() As DatosJobs
Dim TrabajosM() As DatosJobs 'DataJobs que solo va a contener los trabajos que quiero que haga ese modelo (argumento de GenerarModelo)
Dim NumJobs As Integer 'cantidad total de trabajos
Dim NumJobsM As Integer 'cantidad de trabajos que va a haber en TrabajosM() cada vez
Dim NumMachines As Integer 'cantidad total de máquinas
Dim NumClasJobs As Integer 'añadido para la cantidad de clases de trabajos
Dim NumClasMachines As Integer 'añadido para la cantidad de clases de máquinas
Dim NumArchivoR As Integer

Dim iSolape, iCantidadTrabajos, iTipoL, iCantidadMaquinas, iDistribucionMaquinas, iContador As Integer
Dim Solape(), CantidadTrabajos(), CantidadMaquinas() As Integer

Dim L() As Integer
Dim LModMach() As Integer 'Para almacenar la matriz L() y poder modificarla cuando hayamos usado una maquina

Dim PrioMach As Integer 'Almacenamos el valor de la maquina prioritaria en cada iteracion

Dim Maquinas() As Integer

Dim TipoClaseMaquina As Integer
```

```

Dim FOTotal As Double
Dim NumeroTotalTrabajosRealizados As Double

Dim IDProblema As Integer

Dim NumArchivo2 As Integer
Public Declare Function GetSystemMenu% Lib "user32.dll" (ByVal hwnd%)

Sub main()

Dim iter As Integer
Dim i As Integer
Dim j As Integer

Dim PrioMachines() As Integer 'Creo un vector para almacenar el orden en el que doy prioridad a las maquinas para comparar si es el
optimo

Dim NumArchivoRes As Integer
Dim BanderaExcepcion As Integer

Dim T1 As Double
Dim T2 As Double
Dim T As Double

ReDim PrioMachines(3)

ReDim Solape(3)
ReDim CantidadTrabajos(5)
ReDim CantidadMaquinas(3)

Solape(1) = 1
Solape(2) = 2
Solape(3) = 4
CantidadTrabajos(1) = 50
CantidadTrabajos(2) = 100
CantidadTrabajos(3) = 200
CantidadTrabajos(4) = 500
CantidadTrabajos(5) = 1000
CantidadMaquinas(1) = 4
CantidadMaquinas(2) = 8
CantidadMaquinas(3) = 16

'Archivo donde ir haciendo un registro de comprobaciones
NumArchivo2 = FreeFile
Open App.Path & "\Comprobacion.dat" For Output As NumArchivo2
Print #NumArchivo2, "Inicio"

NumArchivoRes = FreeFile
Open App.Path & "\Resultados_Est_2" & ".txt" For Output As NumArchivoRes

For iSolape = 1 To 3
For iCantidadTrabajos = 1 To 5
For iTipoL = 1 To 3
For iCantidadMaquinas = 1 To 3
For iDistribucionMaquinas = 1 To 4
For iContador = 1 To 10

'Recojo aqui las excepciones ya que no hay problemas para algunas combinaciones que carecen de sentido
If CantidadTrabajos(iCantidadTrabajos) = 50 Then
  If CantidadMaquinas(iCantidadMaquinas) > 4 Then
    BanderaExcepcion = 1
  End If

```

```

End If
If CantidadTrabajos(iCantidadTrabajos) = 100 Then
  If CantidadMaquinas(iCantidadMaquinas) > 8 Then
    BanderaExcepcion = 1
  End If
End If

If BanderaExcepcion = 0 Then

  T1 = Timer
  IDProblema = IDProblema + 1
  FOTotal = 0
  NumeroTotalTrabajosRealizados = 0

  LeerArchivoEntrada Solape(iSolape), CantidadTrabajos(iCantidadTrabajos), iTipoL, CantidadMaquinas(iCantidadMaquinas),
  iDistribucionMaquinas, iContador

  For iter = 1 To NumClasMachines

    AsignarPrioridadMaquinas

    PrioMachines(iter) = PrioMach

    For i = 1 To NumClasJobs 'Modifico la matriz L para hacer que no se vuelva a usar la misma maquina
      LModMach(i, PrioMach) = 0
    Next i

    GenerarTrabajosM PrioMach

    ImprimirTrabajos1 'Imprime en un nuevo archivo la matriz trabajos() antes de hacerle nada
    ImprimirTrabajosM 'Imprime en un nuevo archivo la matriz trabajosM() antes de hacerle nada

    GenerarModelo

    If NumJobsM > 0 Then 'Comprobación de que sigue habiendo trabajos disponibles para realizar
      ResolverFlujoCosteMinimo
      LeerArchivoSalida
      ImprimirTrabajos2 'Imprime en un nuevo archivo la matriz trabajos() despues de solucionar el flujo
    End If
  Next iter
  T2 = Timer
  T = T2 - T1
  ImprimirResultado
  Print #NumArchivoRes, (CStr(IDProblema) & " " & CStr(FOTotal) & " " & CStr(PrioMachines(1)) & " " & CStr(PrioMachines(2)) & " " &
  CStr(PrioMachines(3)) & " " & CStr(T))

End If 'if de no ejecutar si no existe el problema, para los de 50 y 100 que no tienen todas las combinaciones
BanderaExcepcion = 0

Next iContador
Next iDistribucionMaquinas
Next iCantidadMaquinas
Next iTipoL
Next iCantidadTrabajos
Next iSolape

End Sub

Sub LeerArchivoEntrada(ByVal Solape As Integer, ByVal NumTrab As Integer, ByVal TipoL As Integer, ByVal NumMaq As Integer, ByVal
DisMaq As Integer, ByVal Cont As Integer)

  Dim NumArchivo As Integer
  Dim Linea As String
  Dim Indice As Integer

```

```

Dim i As Integer 'añadido para leer L
Dim j As Integer 'añadido para leer L
Dim Pos As Integer
Dim Izq As String
Dim Der As String

NumArchivo = FreeFile
Open App.Path & "\bateria2015\Fichero_" & CStr(Solape) & "_" & CStr(NumTrab) & "_" & CStr(TipoL) & "_" & CStr(NumMach) & "_" & CStr(DisMach) & "_" & CStr(Cont) & ".txt" For Input As NumArchivo
Line Input #NumArchivo, Linea
Linea = Trim(Linea)
NumJobs = CInt(Linea)

If NumJobs <> NumTrab Then 'seguridad de que NumJobs coincide con el argumento NumTrab
    Print #NumArchivo2, "Error! NumJobs distinto a NumTrab"
End If

ReDim Trabajos(NumJobs) 'Dimensiona el struct según el número de trabajos

For Indice = 1 To NumJobs
    Line Input #NumArchivo, Linea

    Linea = Trim(Linea)
    Trabajos(Indice).li = Indice
    Pos = InStr(Linea, " ")
    Izq = Left(Linea, Pos - 1) ' lo que hay a la izquierda
    Trabajos(Indice).si = CInt(Izq) ' lee el instante de inicio
    Der = Right(Linea, Len(Linea) - Pos) ' lo que hay a la derecha

    Linea = Der
    Pos = InStr(Linea, " ")
    Izq = Left(Linea, Pos - 1)
    Trabajos(Indice).fi = CInt(Izq) ' lee el instante de fin del trabajo
    Der = Right(Linea, Len(Linea) - Pos) ' lo que hay a la derecha

    Linea = Der
    Pos = InStr(Linea, " ")
    Izq = Left(Linea, Pos - 1)
    Trabajos(Indice).ti = CInt(Izq) ' lee el tiempo del trabajo
    Der = Right(Linea, Len(Linea) - Pos) ' lo que hay a la derecha

    Linea = Der
    Pos = InStr(Linea, " ")
    Izq = Left(Linea, Pos - 1)
    Trabajos(Indice).wi = CInt(Izq) ' lee el peso del trabajo
    Der = Right(Linea, Len(Linea) - Pos) ' lo que hay a la derecha

    Trabajos(Indice).di = CInt(Der) 'lo que queda ya es el tipo de trabajo
    Trabajos(Indice).xi = 0 'inicio a cero todos los trabajos como no realizados

Next

Line Input #NumArchivo, Linea 'hay una linea vacía
Line Input #NumArchivo, Linea 'leo la cantidad de tipos de trabajos
NumClasJobs = CInt(Linea)

Line Input #NumArchivo, Linea 'leo la cantidad de tipos de maquinas
NumClasMachines = CInt(Linea)

ReDim L(NumClasJobs, NumClasMachines) 'Creo la matriz L de dimension en función de las variables leidas
ReDim LModMach(NumClasJobs, NumClasMachines)

For i = 1 To NumClasJobs 'bucles para leer la matriz L en funcion del número de clases de maquinas y de trabajos

```

```

Line Input #NumArchivo, Linea
  For j = 1 To NumClasMachines - 1
    Pos = InStr(Linea, " ")
    Izq = Left(Linea, Pos - 1)
    L(i, j) = Clnt(Izq)
    Der = Right(Linea, Len(Linea) - (Pos + 1)) ' Ojo: hay 2 espacios en la matriz L
    Linea = Der
  Next j
j = NumClasMachines
L(i, j) = Clnt(Der)
Next i

Line Input #NumArchivo, Linea 'leo el número total de maquinas
Linea = Trim(Linea)
NumMachines = Clnt(Linea)

ReDim Maquinas(NumClasMachines) 'Creo el vector de longitud acorde al número de clases de máquinas
For j = 1 To NumClasMachines 'leo cuantas máquinas de cada clase tengo
  Line Input #NumArchivo, Linea
  Maquinas(j) = Clnt(Linea)
Next j

LModMach() = L()
Close NumArchivo
End Sub

'Funcion para decidir la prioridad con la que vamos introduciendo las maquinas
Sub AsignarPrioridadMaquinas()

Dim AuxM() As Integer 'vector auxiliar donde almacenar la compatibilidad de cada maquina
Dim AuxJ() As Integer 'vector auxiliar donde almacenar la compatibilidad de cada trabajo

Dim i As Integer
Dim ii As Integer
Dim j As Integer
Dim jj As Integer

Dim Min As Integer
Dim MinimOs As Integer

Dim Max() As Integer
Dim Maxi As Integer
Dim Maximos As Integer

Dim SumatorioPesos() As Double
Dim MaxSumPesos As Double

ReDim AuxM(NumClasMachines)
ReDim AuxJ(NumClasJobs)

'1º Busco el grado de compatibilidad de cada maquina y el valor minimo
Min = 1000
For j = 1 To NumClasMachines
  For i = 1 To NumClasJobs
    If LModMach(i, j) = 1 Then
      AuxM(j) = AuxM(j) + 1
    End If
  Next i
  If AuxM(j) < Min Then
    If AuxM(j) > 0 Then
      Min = AuxM(j)
    End If
  End If
Next j

```

```

'Busco la/s maquina/s que tienen la compatibilidad minima

Minimos = 0
For j = 1 To NumClasMachines
  If AuxM(j) = Min Then
    Minimos = Minimos + 1
  End If
Next j

If Minimos = 1 Then
For j = 1 To NumClasMachines
  If AuxM(j) = Min Then
    PrioMach = j
  End If
Next j
End If
'2º Si hay varias maquinas con el minimo grado de compatibilidad
If Minimos > 1 Then
  ReDim Max(NumClasMachines) 'Necesito almacenar "Minimos" número de Maximos (uno por cada maquina de compatibilidad
minima
  Maxi = 0
  'De las maquinas que tienen el minimo de compatibilidad busco la que hace el trabajo más compatible
  For j = 1 To NumClasMachines
    Max(j) = 0
    If AuxM(j) = Min Then 'Las maquinas menos compatibles
      For i = 1 To NumClasJobs
        If LModMach(i, j) = 1 Then 'Para cada trabajo de los compatibles con la maquina
          For jj = 1 To NumClasMachines
            AuxJ(i) = AuxJ(i) + LModMach(i, jj) 'Almaceno la compatibilidad de los trabajos en AuxJ()
          Next jj
          If AuxJ(i) > Max(j) Then 'Almaceno el Maximo en Max(j) porque así tendré el número de compatibilidades del trabajo mas
compatible de cada maquina
            Max(j) = AuxJ(i)
          End If
          If Max(j) > Maxi Then 'Busco el maximo de los maximos para identificarlo luego
            Maxi = Max(j)
          End If
        End If
      Next i
    End If
  Next j

  Maximos = 0
  For j = 1 To NumClasMachines
    If Max(j) = Maxi Then 'Las maquinas menos compatibles
      Maximos = Maximos + 1
    End If
  Next j

  If Maximos = 1 Then 'Recorrer y acabo cuando encuentre dicho maximo asignando la prioridad
    For j = 1 To NumClasMachines
      If Max(j) = Maxi Then
        PrioMach = j
      End If
    Next j
  End If
  If Maximos > 1 Then
    '3º Grado, tienen misma compatibilidad las maquinas y misma compatibilidad el mas compatible de sus trabajos
    'Busco mayor sumatorio de trabajos
    ReDim SumatorioPesos(NumClasMachines)
    For j = 1 To NumClasMachines 'Busco el sumatorio de pesos de cada maquina y el Maximo
      SumatorioPesos(j) = 0
      If AuxM(j) = Min Then

```

```

    For ii = 1 To NumJobs
      If Trabajos(ii).xi = 0 Then
        If LModMach(Trabajos(ii).di, j) = 1 Then
          SumatorioPesos(j) = SumatorioPesos(j) + Trabajos(ii).wi
        End If
      End If
    Next ii
    If SumatorioPesos(j) > MaxSumPesos Then
      MaxSumPesos = SumatorioPesos(j)
    End If
  End If
Next j
For j = 1 To NumClasMachines 'Busco cual de las máquinas esta en ese máximo y esa será la prioritaria
  If SumatorioPesos(j) = MaxSumPesos Then
    PrioMach = j
  End If
Next j

End If

End If

End Sub

'**** Función creada por mí para hacer un TrabajosM con los trabajos para la clase de maquina que yo diga
'**** TrabajosM() quiero que contenga solo los trabajos que van a ser compatibles con la clase de maquina que recibe
'**** Además en TrabajosM() no van a entrar los trabajos que ya han sido realizados
Sub GenerarTrabajosM(ByVal ClaseMaquina As Integer)

Dim TrabajosMM() As DatosJobs 'Uso una variable local trabajosMM para que se reinicie cada vez que llame a esta función
Dim IndiceOriginal As Integer
Dim IndiceM As Integer
Dim d As Integer

IndiceM = 0
NumMachines = 0

' Qué clases de trabajos son compatibles con la máquina? -> Creo la matriz TrabajosM que sera argumento de GenerarModelo
For IndiceOriginal = 1 To NumJobs
  'Recorrer toda la matriz Trabajos() original
  If Trabajos(IndiceOriginal).xi = 0 Then
    'Si el trabajo ya ha sido realizado no quiero que entre
    For d = 1 To NumClasJobs
      'Recorrer la matriz L para todas las clases de trabajos
      If L(d, ClaseMaquina) = 1 Then
        'Si la clase de trabajo es compatible con la clase de maquina
        If Trabajos(IndiceOriginal).di = d Then
          'Si el trabajo es de esa clase de trabajo (compatible)
          IndiceM = IndiceM + 1
          'Incremento el indice que me va a decir cuantos trabajos tengo en TrabajosM
          Trabajos(IndiceOriginal).mi = IndiceM
          'Almaceno cual va a ser su nuevo indice para poder hacer la conversion desde el
          archivo Salida.dat
          ReDim Preserve TrabajosMM(IndiceM)
          'Hay que redimensionar cada vez la matriz de estructuras conservando los
          elementos anteriores
          TrabajosMM(IndiceM) = Trabajos(IndiceOriginal) 'Copio el trabajo a mi nueva matriz TrabajosMM
        End If
      End If
    Next d
  End If
End If
Next IndiceOriginal

TrabajosM() = TrabajosMM() 'Cuando termino almaceno en el TrabajosM que es el que voy a usar en los siguientes pasos; funciona si no
hay ningun trabajo compatible??
NumJobsM = IndiceM 'Almaceno una variable que me diga cuantos trabajos tengo en mi TrabajosM() (-1 porque si
NumMachines = Maquinas(ClaseMaquina) 'Almaceno el número de máquinas de esta clase para usarlo en GenerarModelo

End Sub

```

```

**** GenerarModelo Modificada, usa las variables globales TrabajosM() NumJobsM y NumMachinesM y crea el archivo Entrada.dat
Sub GenerarModelo()
  Dim NumArchivo As Integer
  Dim Linea As String
  Dim Cadena As String
  Dim Indice As Long
  Dim Pos As Integer
  Dim i As Integer
  Dim j As Integer
  Dim K As Integer
  Dim L As Integer
  Dim Max As Integer
  Dim M As Integer
  Dim Ind As Long
  Dim TrabajosEjecutados(2000) As Integer

  'OrdenarPorsi Trabajos() 'Mejor que vengan ordenados, podría alterar la conversión de índices?

  NumArchivo = FreeFile
  Open App.Path & "\Entrada.dat" For Output As NumArchivo

  ' Numero de nodos y de arcos porque hay tantos nodos como segundos máximos (ojo, el segundo 0 no existe); el total de arcos es la
  ' la suma de los arcos de transporte (segundos Max-1)+los arcos de trabajos (NumJobsM)

  Max = 0
  For M = 1 To NumJobsM
    If Max < TrabajosM(M).fi Then
      Max = TrabajosM(M).fi
    End If
  Next M

  Cadena = CStr(Max) & " " & CStr((NumJobsM) + (Max - 1))
  Print #NumArchivo, Cadena
  Cadena = ""

  ' lineas de arcos correspondientes a los trabajos

  For Indice = 1 To NumJobsM
    Cadena = CStr(TrabajosM(Indice).si) & " " & CStr((TrabajosM(Indice).si + TrabajosM(Indice).ti)) & " " & "1" & " " & "-" &
    CStr(TrabajosM(Indice).wi)
    Print #NumArchivo, Cadena
    Cadena = ""
  Next Indice

  ' lineas de arcos para el desplazamiento temporal

  For Indice = 1 To Max - 1
    Cadena = CStr(Indice) & " " & CStr(Indice + 1) & " " & 10000 & " " & 0
    Print #NumArchivo, Cadena
    Cadena = ""
  Next Indice

  ' Unidades de flujo cedidas y absorbidas por cada nodo

  Cadena = "-" & CStr(NumMachines)
  Print #NumArchivo, Cadena
  Cadena = ""
  For Indice = 1 To Max - 2
    Cadena = 0
    Print #NumArchivo, Cadena
    Cadena = ""
  Next Indice

```

```

Cadena = CStr(NumMachines)
Print #NumArchivo, Cadena
Cadena = ""
Close NumArchivo

'Damos una espera hasta que el archivo Entrada.dat este completamente cerrado

While Dir(App.Path & "\Entrada.dat") = " "
Indice = Indice + 1
Wend

End Sub

**** Llamada a la funcion Shell para ejecutar Flujo.exe; este programa toma Entrada.dat y devuelve Salida.dat con la solucion del FCM
Sub ResolverFlujoCosteMinimo()

Dim Salida As Integer
Dim i As Integer
Dim fs As New Scripting.FileSystemObject
Dim F As String

Salida = Shell(App.Path & "\FLUJO.EXE entrada.dat salida.dat comproba.dat", vbHide)

F = App.Path & "\comproba.dat"
While fs.FileExists(F) = False

Wend
fs.DeleteFile F

For i = 1 To 10000
Next i

End Sub

**** Función para interpretar la información de Salida.dat
Sub LeerArchivoSalida() 'Leemos la información del archivo salida (valor de la FO y arcos de trabajo usados solo, los arcos de transporte
no me interesan)

'Dim FO As String 'Para almacenar la cadena del valor de la función objetivo
Dim IndiceOriginal As Integer
Dim NumArchivo As Integer
Dim Linea As String
Dim Pos As Integer
Dim Indice As Integer
Dim FO As Double

NumArchivo = FreeFile
Open App.Path & "\Salida.dat" For Input As NumArchivo
Line Input #NumArchivo, Linea 'Linea de la función objetivo
Pos = InStr(Linea, ".")
FO = CDbI(Left(Linea, Pos - 1)) * -1 'Lo que hay a la izda del punto es el valor de la FO porque siempre es -xxx.00000
For Indice = 1 To NumJobsM 'Leo las primeras "NumJobsM" filas que van a ser las que me digan como estan los trabajos
Line Input #NumArchivo, Linea
If Cint(Linea) = 1 Then 'Si el trabajo se ha realizado aquí
IndiceOriginal = TrabajosM(Indice).li 'La reconversión de indice. El indice original esta intacto en .li
If Trabajos(IndiceOriginal).xi = 1 Then 'Por seguridad
Print #NumArchivo2, "ERROR! El trabajo con este indice ya había sido realizado" & CStr(IndiceOriginal)
End If
Trabajos(IndiceOriginal).xi = 1 'El trabajo ha sido realizado
Trabajos(IndiceOriginal).ci = PrioMach 'Almaceno aqui la clase de maquina que ha resuelto este trabajo
NumeroTotalTrabajosRealizados = NumeroTotalTrabajosRealizados + 1 'Comprobación simplemente
End If
Next Indice
Close NumArchivo

```

```

FOTotal = FOTotal + FO 'Vamos acumulando el valor de la FO
End Sub

'**** Función complementaria para comprobar el valor de la matriz Trabajos() antes de resolver el FCM
Sub ImprimirTrabajos1()
Dim i As Integer
Dim NumArchivoX As Integer
Dim Cadena As String
NumArchivoX = FreeFile
Open App.Path & "\Trabajos1.dat" For Output As NumArchivoX
For i = 1 To NumJobs
Cadena = ""
Cadena = CStr(Trabajos(i).si) & " " & CStr(Trabajos(i).fi) & " " & CStr(Trabajos(i).ti) & " " & CStr(Trabajos(i).wi) & " " & CStr(Trabajos(i).di)
& " " & CStr(Trabajos(i).li) & " " & CStr(Trabajos(i).mi) & " " & CStr(Trabajos(i).xi)
Print #NumArchivoX, Cadena
Next i
Close NumArchivoX
End Sub

'**** Función complementaria para comprobar el valor de TrabajosM() que van a entrar para generar el modelo que entrara a FCM
Sub ImprimirTrabajosM()
Dim i As Integer
Dim NumArchivoX As Integer
NumArchivoX = FreeFile
Open App.Path & "\TrabajosM.dat" For Output As NumArchivoX
For i = 1 To NumJobsM
Print #NumArchivoX, (CStr(TrabajosM(i).si) & " " & CStr(TrabajosM(i).fi) & " " & CStr(TrabajosM(i).ti) & " " & CStr(TrabajosM(i).wi) & " "
& CStr(TrabajosM(i).di) & " " & CStr(TrabajosM(i).li) & " " & CStr(TrabajosM(i).mi) & " " & CStr(TrabajosM(i).xi))
Next i
Close NumArchivoX
End Sub

'**** Función complementaria para comprobar el valor de Trabajos() después de resolver la iteración y habiendo interpretado salida.dat
Sub ImprimirTrabajos2()
Dim i As Integer
Dim NumArchivoX As Integer
NumArchivoX = FreeFile
Open App.Path & "\Trabajos2.dat" For Output As NumArchivoX
For i = 1 To NumJobs
Print #NumArchivoX, (CStr(Trabajos(i).si) & " " & CStr(Trabajos(i).fi) & " " & CStr(Trabajos(i).ti) & " " & CStr(Trabajos(i).wi) & " " &
CStr(Trabajos(i).di) & " " & CStr(Trabajos(i).li) & " " & CStr(Trabajos(i).mi) & " " & CStr(Trabajos(i).xi))
Next i
Close NumArchivoX
End Sub

'**** Función complementaria para imprimir los resultados
Sub ImprimirResultado()
Dim i As Integer
Dim NumArchivoX As Integer

NumArchivoX = FreeFile
Open App.Path & "\Resultados" & "\Res_Est_2_" & CStr(Solape(iSolape)) & "_" & CStr(CantidadTrabajos(iCantidadTrabajos)) & "_" &
CStr(iTipoL) & "_" & CStr(CantidadMaquinas(iCantidadMaquinas)) & "_" & CStr(iDistribucionMaquinas) & "_" & CStr(iContador) & ".txt"
For Output As NumArchivoX

Print #NumArchivoX, CStr(FOTotal) & " " & CStr(NumeroTotalTrabajosRealizados)

For i = 1 To NumJobs
Print #NumArchivoX, (CStr(Trabajos(i).si) & " " & CStr(Trabajos(i).fi) & " " & CStr(Trabajos(i).ti) & " " & CStr(Trabajos(i).wi) & " " &
CStr(Trabajos(i).mi) & " " & CStr(Trabajos(i).li) & " " & CStr(Trabajos(i).di) & " " & CStr(Trabajos(i).ci) & " " & CStr(Trabajos(i).xi))
Next i
Close NumArchivoX
End Sub

```

```
Sub AbrirR() ' Abrir archivo para la escritura de resultados
  NumArchivoR = FreeFile
  Open App.Path & "\Resultados.txt" For Output As NumArchivoR
End Sub
```

```
Sub CerrarR()
  Close NumArchivoR
End Sub
```


9.2 Código estrategia 2.1 VSP

Se adjunta en este punto de anexo el código completo utilizado por la estrategia 2.1 para que sirva a la vez como ejemplo de programa en secuenciación VSP y como muestra de la mejor de las estrategias heurísticas VSP

```
Attribute VB_Name = "FSPNoclasesOperacional"
'ESTRATEGIA 2.1 VSP Ventana temporal fija +-1seg

Option Explicit
Option Base 1
Type DatosJobs
    si As Long
    fi As Integer
    ti As Integer
    wi As Integer
    di As Integer 'añadido para especificar la clase de trabajo
    ci As Integer 'añadido para almacenar la clase de máquina que ha resuelto este trabajo, útil para analizar los resultados en estrategia 2
    li As Integer 'vector de ordenacion
    mi As Integer 'añadido para asignar un nuevo indice de identificación en la estructura trabajosM que le quiero pasar a GenerarModelo
    xi As Integer 'Indice que me dira si el trabajo esta realizado o no 1 realizado 0 no realizado
    ei As Integer 'Indice que me dira si uno de los trabajos de la ventana de tiempo se ha realizado
    ai As Integer 'Indice que indica en que posición de la ventana temporal está este problema
    ni As Integer 'Indice que indica cuantos trabajos he puesto en la ventana temporal, si es fija será el mismo número en todos los trabajos
End Type
Dim Trabajos() As DatosJobs
Dim TrabajosM() As DatosJobs 'DataJobs que solo va a contener los trabajos que quiero que haga ese modelo (argumento de GenerarModelo)
Dim NumJobs As Integer 'cantidad total de trabajos
Dim NumJobsM As Integer 'cantidad de trabajos que va a haber en TrabajosM() cada vez
Dim NumMachines As Integer 'cantidad total de máquinas
Dim NumClasJobs As Integer 'añadido para la cantidad de clases de trabajos
Dim NumClasMachines As Integer 'añadido para la cantidad de clases de máquinas
Dim NumArchivoR As Integer

Dim iSolape, iCantidadTrabajos, iTipoL, iCantidadMaquinas, iDistribucionMaquinas, iContador As Integer
Dim Solape(), CantidadTrabajos(), CantidadMaquinas() As Integer

Dim L() As Integer
Dim LModMach() As Integer 'Para almacenar la matriz L() y poder modificarla cuando hayamos usado una maquina

Dim PrioMach As Integer 'Almacenamos el valor de la maquina prioritaria en cada iteracion

Dim Maquinas() As Integer

Dim TipoClaseMaquina As Integer

Dim FOTotal As Double
Dim NumeroTotalTrabajosRealizados As Double

Dim IDProblema As Integer

Dim VentanaTemporal As Integer 'Variable global donde almacenar la ventana temporal para el VSP
Dim NumJobsVentana As Double 'Numero de trabajos contando todos los creados al abrir una ventana temporal en cada uno
Dim TrabajosOriginales() As DatosJobs 'Donde almacenar los trabajos originales, que serán NumJobs o NumTrab, deben ser lo mismo

Dim NumArchivo2 As Integer
Public Declare Function GetSystemMenu% Lib "user32.dll" (ByVal hwnd%)

Sub main()

Dim iter As Integer
```

```

Dim i As Integer
Dim j As Integer

Dim PrioMachines() As Integer 'Creo un vector para almacenar el orden en el que doy prioridad a las maquinas para comparar si es el
optimo

Dim NumArchivoRes As Integer
Dim BanderaExcepcion As Integer

Dim T1 As Double
Dim T2 As Double
Dim T As Double

Dim MaquinaVSP As Integer

ReDim PrioMachines(3)

ReDim Solape(3)
ReDim CantidadTrabajos(5)
ReDim CantidadMaquinas(3)

Solape(1) = 1
Solape(2) = 2
Solape(3) = 4
CantidadTrabajos(1) = 50
CantidadTrabajos(2) = 100
CantidadTrabajos(3) = 200
CantidadTrabajos(4) = 500
CantidadTrabajos(5) = 1000
CantidadMaquinas(1) = 4
CantidadMaquinas(2) = 8
CantidadMaquinas(3) = 16

'Archivo donde ir haciendo un registro de comprobaciones
NumArchivo2 = FreeFile
Open App.Path & "\Comprobacion.dat" For Output As NumArchivo2
Print #NumArchivo2, "Inicio"

NumArchivoRes = FreeFile
Open App.Path & "\Resultados_Est_2" & ".txt" For Output As NumArchivoRes

For iSolape = 1 To 3
For iCantidadTrabajos = 1 To 5
For iTipoL = 1 To 3
For iCantidadMaquinas = 1 To 3
For iDistribucionMaquinas = 1 To 4
For iContador = 1 To 10

'Recojo aqui las excepciones ya que no hay problemas para algunas combinaciones que carecen de sentido
If CantidadTrabajos(iCantidadTrabajos) = 50 Then
  If CantidadMaquinas(iCantidadMaquinas) > 4 Then
    BanderaExcepcion = 1
  End If
End If
If CantidadTrabajos(iCantidadTrabajos) = 100 Then
  If CantidadMaquinas(iCantidadMaquinas) > 8 Then
    BanderaExcepcion = 1
  End If
End If

If BanderaExcepcion = 0 Then

T1 = Timer

```

```

IDProblema = IDProblema + 1
FOTotal = 0
NumeroTotalTrabajosRealizados = 0

LeerArchivoEntrada Solape(iSolape), CantidadTrabajos(iCantidadTrabajos), iTipoL, CantidadMaquinas(iCantidadMaquinas),
iDistribucionMaquinas, iContador

For iter = 1 To NumClasMachines

    AsignarPrioridadMaquinas

    PrioMachines(iter) = PrioMach
    For MaquinaVSP = 1 To Maquinas(PrioMach) 'Bucle por estar en VSP, hay que hacer tantas iteraciones como cantidad de maquinas
    inyectando siempre una máquina al flujocosteminimo

        For i = 1 To NumClasJobs 'Modifico la matriz L para hacer que no se vuelva a usar la misma maquina
            LModMach(i, PrioMach) = 0
        Next i

        GenerarTrabajosM PrioMach

        ImprimirTrabajos1 'Imprime en un nuevo archivo la matriz trabajos() antes de hacerle nada
        ImprimirTrabajosM 'Imprime en un nuevo archivo la matriz trabajosM() antes de hacerle nada

        GenerarModelo

        If NumJobsM > 0 Then 'Comprobación de que sigue habiendo trabajos disponibles para realizar
            ResolverFlujoCosteMinimo
            LeerArchivoSalida
            ImprimirTrabajos2 'Imprime en un nuevo archivo la matriz trabajos() despues de solucionar el flujo
        End If
    Next MaquinaVSP
Next iter
T2 = Timer
T = T2 - T1
ImprimirResultado
Print #NumArchivoRes, (CStr(IDProblema) & " " & CStr(FOTotal) & " " & CStr(PrioMachines(1)) & " " & CStr(PrioMachines(2)) & " " &
CStr(PrioMachines(3)) & " " & CStr(T))

End If 'if de no ejecutar si no existe el problema, para los de 50 y 100 que no tienen todas las combinaciones
BanderaExcepcion = 0

Next iContador
Next iDistribucionMaquinas
Next iCantidadMaquinas
Next iTipoL
Next iCantidadTrabajos
Next iSolape

End Sub
Sub LeerArchivoEntrada(ByVal Solap As Integer, ByVal NumTrab As Integer, ByVal TipL As Integer, ByVal NumMaquinas As Integer, ByVal
DisMaquinas As Integer, ByVal Con As Integer)

    Dim NumArchivo As Integer
    Dim Linea As String
    Dim Indice As Integer
    Dim i As Integer 'añadido para leer L
    Dim j As Integer 'añadido para leer L
    Dim Pos As Integer
    Dim Izq As String
    Dim Der As String
    Dim IndiceVentana As Double

    NumArchivo = FreeFile

```

```

Open App.Path & "\bateria2015\Fichero_" & CStr(Solap) & "_" & CStr(NumTrab) & "_" & CStr(TipL) & "_" & CStr(NumMaquinas) & "_"
& CStr(DisMaquinas) & "_" & CStr(Con) & ".txt" For Input As NumArchivo
Line Input #NumArchivo, Linea
Linea = Trim(Linea)
NumJobs = Clnt(Linea)

If NumJobs <> NumTrab Then 'seguridad de que NumJobs coincide con el argumento NumTrab
  Print #NumArchivo2, "Error! NumJobs distinto a NumTrab"
End If

ReDim TrabajosOriginales(NumJobs) 'Dimensiona el struct según el número de TrabajosOriginales
VentanaTemporal = 1 'Si el trabajo mas largo es de longitud 3 la mayor ventana temporal es de +-1 segundo
NumJobsVentana = 0 'Inicio a 0 para evitar problemas despues de iteraciones

For Indice = 1 To NumJobs
  Line Input #NumArchivo, Linea

  Linea = Trim(Linea)
  TrabajosOriginales(Indice).li = Indice
  Pos = InStr(Linea, " ")
  Izq = Left(Linea, Pos - 1) ' lo que hay a la izquierda
  TrabajosOriginales(Indice).si = Clnt(Izq) ' lee el instante de inicio
  Der = Right(Linea, Len(Linea) - Pos) ' lo que hay a la derecha

  Linea = Der
  Pos = InStr(Linea, " ")
  Izq = Left(Linea, Pos - 1)
  TrabajosOriginales(Indice).fi = Clnt(Izq) ' lee el instante de fin del trabajo
  Der = Right(Linea, Len(Linea) - Pos) ' lo que hay a la derecha

  Linea = Der
  Pos = InStr(Linea, " ")
  Izq = Left(Linea, Pos - 1)
  TrabajosOriginales(Indice).ti = Clnt(Izq) ' lee el tiempo del trabajo
  Der = Right(Linea, Len(Linea) - Pos) ' lo que hay a la derecha

  Linea = Der
  Pos = InStr(Linea, " ")
  Izq = Left(Linea, Pos - 1)
  TrabajosOriginales(Indice).wi = Clnt(Izq) ' lee el peso del trabajo
  Der = Right(Linea, Len(Linea) - Pos) ' lo que hay a la derecha

  TrabajosOriginales(Indice).di = Clnt(Der) 'lo que queda ya es el tipo de trabajo
  TrabajosOriginales(Indice).xi = 0 'inicio a cero todos los TrabajosOriginales como no realizados

  TrabajosOriginales(Indice).ei = 0 'inicio a cero el identificador de trabajo eliminado
  TrabajosOriginales(Indice).ai = 0 'inicio a cero el identificador de trabajos de la misma ventana

  For i = (TrabajosOriginales(Indice).si - VentanaTemporal) To (TrabajosOriginales(Indice).si + VentanaTemporal) 'Debo comprobar que
  cuando vaya a crear la ventana temporal no me salga un comienzo = 0 o menor
  If i > 0 Then
    TrabajosOriginales(Indice).ni = TrabajosOriginales(Indice).ni + 1 '(VentanaTemporal * 2) 'este sera el valor del total de trabajos en la
    misma ventana
    NumJobsVentana = NumJobsVentana + 1 'Cuento el número total de trabajos cuando hemos añadido los de la ventana, aquí si tengo
    que contar el original
  End If
  Next i
  TrabajosOriginales(Indice).ni = TrabajosOriginales(Indice).ni - 1 'En el valor ni solo quiero los que son de ventana temporal, sin contar
  el original

Next Indice

ReDim Trabajos(NumJobsVentana) 'Redimensionar la matriz de estructuras trabajo a la cantidad con ventana temporal

```

IndiceVentana = 0 'El IndiceVentana nos servira para desplazarnos por la matriz Trabajos con los trabajos de ventana temporal incorporados

For Indice = 1 To NumJobs

IndiceVentana = IndiceVentana + 1 'El IndiceVentana aumenta a ritmo distinto del Indice original
j = 0

Trabajos(IndiceVentana) = TrabajosOriginales(Indice) 'Este será el trabajo original que lleva .ai=0 y los tiempos originales
Trabajos(IndiceVentana).li = IndiceVentana

For i = 1 To VentanaTemporal 'Creo tantos trabajos como haya quepan en la ventana temporal

If (TrabajosOriginales(Indice).si - i) > 0 Then 'Tengo que comprobar que no entramos en negativo otra vez, en el de + i no importa alejarnos por la derecha

IndiceVentana = IndiceVentana + 1

j = j + 1

Trabajos(IndiceVentana) = TrabajosOriginales(Indice) 'Creo los trabajos nuevos con el indice que son de trabajo de la misma ventana temporal(ai) y el tiempo desplazados

Trabajos(IndiceVentana).ai = j

Trabajos(IndiceVentana).si = TrabajosOriginales(Indice).si - i

Trabajos(IndiceVentana).fi = TrabajosOriginales(Indice).fi - i

Trabajos(IndiceVentana).li = IndiceVentana

End If

IndiceVentana = IndiceVentana + 1

j = j + 1

Trabajos(IndiceVentana) = TrabajosOriginales(Indice)

Trabajos(IndiceVentana).ai = j

Trabajos(IndiceVentana).si = TrabajosOriginales(Indice).si + i

Trabajos(IndiceVentana).fi = TrabajosOriginales(Indice).fi + i

Trabajos(IndiceVentana).li = IndiceVentana

Next i

Next Indice

Line Input #NumArchivo, Linea 'hay una linea vacía

Line Input #NumArchivo, Linea 'leo la cantidad de tipos de trabajos

NumClasJobs = Clnt(Linea)

Line Input #NumArchivo, Linea 'leo la cantidad de tipos de maquinas

NumClasMachines = Clnt(Linea)

ReDim L(NumClasJobs, NumClasMachines) 'Creo la matriz L de dimension en función de las variables leidas

ReDim LModMach(NumClasJobs, NumClasMachines)

For i = 1 To NumClasJobs 'bucles para leer la matriz L en funcion del número de clases de maquinas y de trabajos

Line Input #NumArchivo, Linea

For j = 1 To NumClasMachines - 1

Pos = InStr(Linea, " ")

Izq = Left(Linea, Pos - 1)

L(i, j) = Clnt(Izq)

Der = Right(Linea, Len(Linea) - (Pos + 1)) ' Ojo: hay 2 espacios en la matriz L

Linea = Der

Next j

j = NumClasMachines

L(i, j) = Clnt(Der)

Next i

Line Input #NumArchivo, Linea 'leo el número total de maquinas

Linea = Trim(Linea)

NumMachines = Clnt(Linea)

ReDim Maquinas(NumClasMachines) 'Creo el vector de longitud acorde al número de clases de máquinas

For j = 1 To NumClasMachines 'leo cuantas máquinas de cada clase tengo

Line Input #NumArchivo, Linea

Maquinas(j) = Clnt(Linea)

```

Next j

LModMach() = L()
Close NumArchivo
End Sub

'Funcion para decidir la prioridad con la que vamos introduciendo las maquinas
Sub AsignarPrioridadMaquinas()

Dim AuxM() As Integer 'vector auxiliar donde almacenar la compatibilidad de cada maquina
Dim AuxJ() As Integer 'vector auxiliar donde almacenar la compatibilidad de cada trabajo

Dim i As Integer
Dim ii As Integer
Dim j As Integer
Dim jj As Integer

Dim Min As Integer
Dim MinimOs As Integer

Dim Max() As Integer
Dim Maxi As Integer
Dim Maximos As Integer

Dim SumatorioPesos() As Double
Dim MaxSumPesos As Double

ReDim AuxM(NumClasMachines)
ReDim AuxJ(NumClasJobs)

'1º Busco el grado de compatibilidad de cada maquina y el valor minimo
Min = 1000
For j = 1 To NumClasMachines
  For i = 1 To NumClasJobs
    If LModMach(i, j) = 1 Then
      AuxM(j) = AuxM(j) + 1
    End If
  Next i
  If AuxM(j) < Min Then
    If AuxM(j) > 0 Then
      Min = AuxM(j)
    End If
  End If
Next j
'Busco la/s maquina/s que tienen la compatibilidad minima

Minimos = 0
For j = 1 To NumClasMachines
  If AuxM(j) = Min Then
    MinimOs = MinimOs + 1
  End If
Next j

If MinimOs = 1 Then
  For j = 1 To NumClasMachines
    If AuxM(j) = Min Then
      PrioMach = j
    End If
  Next j
End If
'2º Si hay varias maquinas con el minimo grado de compatibilidad
If MinimOs > 1 Then

```

```

ReDim Max(NumClasMachines) 'Necesito almacenar "Minimos" número de Maximos (uno por cada maquina de compatibilidad
minima
Maxi = 0
'De las maquinas que tienen el minimo de compatibilidad busco la que hace el trabajo más compatible
For j = 1 To NumClasMachines
Max(j) = 0
If AuxM(j) = Min Then 'Las maquinas menos compatibles
For i = 1 To NumClasJobs
If LModMach(i, j) = 1 Then 'Para cada trabajo de los compatibles con la maquina
For jj = 1 To NumClasMachines
AuxJ(i) = AuxJ(i) + LModMach(i, jj) 'Almaceno la compatibilidad de los trabajos en AuxJ()
Next jj
If AuxJ(i) > Max(j) Then 'Almaceno el Maximo en Max(j) porque así tendre el número de compatibilidades del trabajo mas
compatible de cada maquina
Max(j) = AuxJ(i)
End If
If Max(j) > Maxi Then 'Busco el maximo de los maximos para identificarlo luego
Maxi = Max(j)
End If
End If
Next i
End If
Next j

Maximos = 0
For j = 1 To NumClasMachines
If Max(j) = Maxi Then 'Las maquinas menos compatibles
Maximos = Maximos + 1
End If
Next j

If Maximos = 1 Then 'Recorrer y acabo cuando encuentre dicho maximo asignando la prioridad
For j = 1 To NumClasMachines
If Max(j) = Maxi Then
PrioMach = j
End If
Next j
End If
If Maximos > 1 Then
'3º Grado, tienen misma compatibilidad las maquinas y misma compatibilidad el mas compatible de sus trabajos
'Busco mayor sumatorio de trabajos
ReDim SumatorioPesos(NumClasMachines)
For j = 1 To NumClasMachines 'Busco el sumatorio de pesos de cada maquina y el Maximo
SumatorioPesos(j) = 0
If AuxM(j) = Min Then
For ii = 1 To NumJobsVentana
If Trabajos(ii).xi = 0 Then
If Trabajos(ii).ei = 0 Then
If LModMach(Trabajos(ii).di, j) = 1 Then
SumatorioPesos(j) = SumatorioPesos(j) + Trabajos(ii).wi
End If
End If
End If
Next ii
If SumatorioPesos(j) > MaxSumPesos Then
MaxSumPesos = SumatorioPesos(j)
End If
End If
Next j
For j = 1 To NumClasMachines 'Busco cual de las máquinas esta en ese máximo y esa será la prioritaria
If SumatorioPesos(j) = MaxSumPesos Then
PrioMach = j
End If
Next j

```

```

End If

End If

End Sub

'**** Función creada por mí para hacer un TrabajosM con los trabajos para la clase de maquina que yo diga
'**** TrabajosM() quiero que contenga solo los trabajos que van a ser compatibles con la clase de maquina que recibe
'**** Además en TrabajosM() no van a entrar los trabajos que ya han sido realizados
Sub GenerarTrabajosM(ByVal ClaseMaquina As Integer)

Dim TrabajosMM() As DatosJobs 'Uso una variable local trabajosMM para que se reinicie cada vez que llame a esta función
Dim IndiceOriginal As Integer
Dim IndiceM As Integer
Dim d As Integer

IndiceM = 0
NumMachines = 0

' Qué clases de trabajos son compatibles con la máquina? -> Creo la matriz TrabajosM que sera argumento de GenerarModelo
For IndiceOriginal = 1 To NumJobsVentana
    'Recorrer toda la matriz Trabajos() original
    If Trabajos(IndiceOriginal).xi = 0 Then
        'Si el trabajo ya ha sido realizado no quiero que entre
        If Trabajos(IndiceOriginal).ei = 0 Then
            'En VSP tambien tengo que comprobar que no se ha hecho ninguno de la ventana
            temporal
            For d = 1 To NumClasJobs
                'Recorrer la matriz L para todas las clases de trabajos
                If L(d, ClaseMaquina) = 1 Then
                    'Si la clase de trabajo es compatible con la clase de maquina
                    If Trabajos(IndiceOriginal).di = d Then
                        'Si el trabajo es de esa clase de trabajo (compatible)
                        IndiceM = IndiceM + 1
                        'Incremento el Indice que me va a decir cuantos trabajos tengo en TrabajosM
                        Trabajos(IndiceOriginal).mi = IndiceM
                        'Almaceno cual va a ser su nuevo indice para poder hacer la conversion desde el
                        archivo Salida.dat
                        ReDim Preserve TrabajosMM(IndiceM)
                        'Hay que redimensionar cada vez la matriz de estructuras conservando los
                        elementos anteriores
                        TrabajosMM(IndiceM) = Trabajos(IndiceOriginal)
                        'Copio el trabajo a mi nueva matriz TrabajosMM
                    End If
                End If
            Next d
        End If
    End If
Next IndiceOriginal

TrabajosM() = TrabajosMM() 'Cuando termino almaceno en el TrabajosM que es el que voy a usar en los siguientes pasos; funciona si no
hay ningun trabajo compatible??
NumJobsM = IndiceM
'Almaceno una variable que me diga cuantos trabajos tengo en mi TrabajosM() (-1 porque si
'NumMachines = Maquinas(ClaseMaquina)
'Almaceno el número de máquinas de esta clase para usarlo en GenerarModelo
NumMachines = 1
End Sub

'**** GenerarModelo Modificada, usa las variables globales TrabajosM() NumJobsM y NumMachinesM y crea el archivo Entrada.dat
Sub GenerarModelo()
Dim NumArchivo As Integer
Dim Linea As String
Dim Cadena As String
Dim Indice As Long
Dim Pos As Integer
Dim i As Integer
Dim j As Integer
Dim K As Integer
Dim L As Integer
Dim Max As Integer
Dim M As Integer

```

```

Dim Ind As Long
Dim TrabajosEjecutados(2000) As Integer

'OrdenarPorsi Trabajos() 'Mejor que vengan ordenados, podría alterar la conversión de índices?

NumArchivo = FreeFile
Open App.Path & "\Entrada.dat" For Output As NumArchivo

' Numero de nodos y de arcos porque hay tantos nodos como segundos máximos (ojo, el segundo 0 no existe); el total de arcos es la
' la suma de los arcos de transporte (segundos Max-1)+los arcos de trabajos (NumJobsM)

Max = 0
For M = 1 To NumJobsM
  If Max < TrabajosM(M).fi Then
    Max = TrabajosM(M).fi
  End If
Next M

Cadena = CStr(Max) & " " & CStr((NumJobsM) + (Max - 1))
Print #NumArchivo, Cadena
Cadena = ""

' lineas de arcos correspondientes a los trabajos

For Indice = 1 To NumJobsM
  Cadena = CStr(TrabajosM(Indice).si) & " " & CStr((TrabajosM(Indice).si + TrabajosM(Indice).ti)) & " " & "1" & " " & "-" &
  CStr(TrabajosM(Indice).wi)
  Print #NumArchivo, Cadena
  Cadena = ""
Next Indice

' lineas de arcos para el desplazamiento temporal

For Indice = 1 To Max - 1
  Cadena = CStr(Indice) & " " & CStr(Indice + 1) & " " & 10000 & " " & 0
  Print #NumArchivo, Cadena
  Cadena = ""
Next Indice

' Unidades de flujo cedidas y absorbidas por cada nodo

Cadena = "-" & CStr(NumMachines)
Print #NumArchivo, Cadena
Cadena = ""
For Indice = 1 To Max - 2
  Cadena = 0
  Print #NumArchivo, Cadena
  Cadena = ""
Next Indice
Cadena = CStr(NumMachines)
Print #NumArchivo, Cadena
Cadena = ""
Close NumArchivo

' Damos una espera hasta que el archivo Entrada.dat este completamente cerrado

While Dir(App.Path & "\Entrada.dat") = " "
  Indice = Indice + 1
Wend

End Sub

**** Llamada a la funcion Shell para ejecutar Flujo.exe; este programa toma Entrada.dat y devuelve Salida.dat con la solucion del FCM

```

```

Sub ResolverFlujoCosteMinimo()

Dim Salida As Integer
Dim i As Integer
Dim fs As New Scripting.FileSystemObject
Dim F As String

Salida = Shell(App.Path & "\FLUJO.EXE entrada.dat salida.dat comproba.dat", vbHide)

F = App.Path & "\comproba.dat"
While fs.FileExists(F) = False

Wend
fs.DeleteFile F

For i = 1 To 10000
Next i

End Sub

'**** Función para interpretar la información de Salida.dat
Sub LeerArchivoSalida() 'Leemos la información del archivo salida (valor de la FO y arcos de trabajo usados solo, los arcos de transporte no me interesan)

'Dim FO As String      'Para almacenar la cadena del valor de la función objetivo
Dim IndiceOriginal As Integer
Dim NumArchivo As Integer
Dim Linea As String
Dim Pos As Integer
Dim Indice As Integer
Dim FO As Double

Dim InicioVentana As Integer 'Defino variables para mostrar como realizados todos los trabajos de la ventana temporal
Dim FinVentana As Integer
Dim iVentana As Integer

NumArchivo = FreeFile
Open App.Path & "\Salida.dat" For Input As NumArchivo
Line Input #NumArchivo, Linea      'Linea de la función objetivo
Pos = InStr(Linea, ".")
FO = Cdbl(Left(Linea, Pos - 1)) * -1      'Lo que hay a la izda del punto es el valor de la FO porque siempre es -xxx.00000
For Indice = 1 To NumJobsM      'Leo las primeras "NumJobsM" filas que van a ser las que me digan como estan los trabajos
  Line Input #NumArchivo, Linea
  If Cint(Linea) = 1 Then      'Si el trabajo se ha realizado aquí
    IndiceOriginal = TrabajosM(Indice).li      'La reconversión de indice. El indice original esta intacto en .li
    If Trabajos(IndiceOriginal).xi = 1 Then      'Por seguridad
      Print #NumArchivo2, "ERROR! El trabajo con este indice ya había sido realizado" & CStr(IndiceOriginal)
    End If
    Trabajos(IndiceOriginal).xi = 1      'El trabajo ha sido realizado
    Trabajos(IndiceOriginal).ci = PrioMach 'Almaceno aqui la clase de maquina que ha resuelto este trabajo
    NumeroTotalTrabajosRealizados = NumeroTotalTrabajosRealizados + 1 'Comprobación simplemente

  InicioVentana = IndiceOriginal - Trabajos(IndiceOriginal).ai 'El primer trabajo de la ventana es el índice menos el contador de ventana
  FinVentana = IndiceOriginal - Trabajos(IndiceOriginal).ai + Trabajos(IndiceOriginal).ni 'El ultimo trabajo de la ventana temporal es el
  inicial de la ventana más el número de trabajos en la ventana

  For iVentana = InicioVentana To FinVentana
    Trabajos(iVentana).ei = 1 'Pongo a 1 para indicar que estos trabajos no han sido realizados pero no se deben tener en cuenta por
    ser de la misma ventana temporal
  Next iVentana
End If
Next Indice
Close NumArchivo

```

```
FOTotal = FOTotal + FO 'Vamos acumulando el valor de la FO
End Sub
```

```
'**** Función complementaria para comprobar el valor de la matriz Trabajos() antes de resolver el FCM
```

```
Sub ImprimirTrabajos1()
Dim i As Integer
Dim NumArchivoX As Integer
Dim Cadena As String
NumArchivoX = FreeFile
Open App.Path & "\Trabajos1.dat" For Output As NumArchivoX
For i = 1 To NumJobsVentana
Cadena = ""
Cadena = CStr(Trabajos(i).si) & " " & CStr(Trabajos(i).fi) & " " & CStr(Trabajos(i).ti) & " " & CStr(Trabajos(i).wi) & " " & CStr(Trabajos(i).di)
& " " & CStr(Trabajos(i).li) & " " & CStr(Trabajos(i).mi) & " " & CStr(Trabajos(i).xi) & " " & CStr(Trabajos(i).ei) & " " & CStr(Trabajos(i).ai) &
" " & CStr(Trabajos(i).ni)
Print #NumArchivoX, Cadena
Next i
Close NumArchivoX
End Sub
```

```
'**** Función complementaria para comprobar el valor de TrabajosM() que van a entrar para generar el modelo que entrara a FCM
```

```
Sub ImprimirTrabajosM()
Dim i As Integer
Dim NumArchivoX As Integer
NumArchivoX = FreeFile
Open App.Path & "\TrabajosM.dat" For Output As NumArchivoX
For i = 1 To NumJobsM
Print #NumArchivoX, (CStr(TrabajosM(i).si) & " " & CStr(TrabajosM(i).fi) & " " & CStr(TrabajosM(i).ti) & " " & CStr(TrabajosM(i).wi) & " "
& CStr(TrabajosM(i).di) & " " & CStr(TrabajosM(i).li) & " " & CStr(TrabajosM(i).mi) & " " & CStr(TrabajosM(i).xi) & " " &
CStr(TrabajosM(i).ei) & " " & CStr(TrabajosM(i).ai) & " " & CStr(TrabajosM(i).ni))
Next i
Close NumArchivoX
End Sub
```

```
'**** Función complementaria para comprobar el valor de Trabajos() después de resolver la iteracion y habiendo interpretado salida.dat
```

```
Sub ImprimirTrabajos2()
Dim i As Integer
Dim NumArchivoX As Integer
NumArchivoX = FreeFile
Open App.Path & "\Trabajos2.dat" For Output As NumArchivoX
For i = 1 To NumJobsVentana
Print #NumArchivoX, (CStr(Trabajos(i).si) & " " & CStr(Trabajos(i).fi) & " " & CStr(Trabajos(i).ti) & " " & CStr(Trabajos(i).wi) & " " &
CStr(Trabajos(i).di) & " " & CStr(Trabajos(i).li) & " " & CStr(Trabajos(i).mi) & " " & CStr(Trabajos(i).xi) & " " & CStr(Trabajos(i).ei) & " " &
CStr(Trabajos(i).ai) & " " & CStr(Trabajos(i).ni))
Next i
Close NumArchivoX
End Sub
```

```
'**** Función complementaria para imprimir los resultados
```

```
Sub ImprimirResultado()
Dim i As Integer
Dim NumArchivoX As Integer

NumArchivoX = FreeFile
Open App.Path & "\Resultados" & "\Res_Est_2_" & CStr(Solape(iSolape)) & "_" & CStr(CantidadTrabajos(iCantidadTrabajos)) & "_" &
CStr(iTipoL) & "_" & CStr(CantidadMaquinas(iCantidadMaquinas)) & "_" & CStr(iDistribucionMaquinas) & "_" & CStr(iContador) & ".txt"
For Output As NumArchivoX
```

```
Print #NumArchivoX, CStr(FOTotal) & " " & CStr(NumeroTotalTrabajosRealizados)
```

```
For i = 1 To NumJobsVentana
```

```
Print #NumArchivoX, (CStr(Trabajos(i).si) & " " & CStr(Trabajos(i).fi) & " " & CStr(Trabajos(i).ti) & " " & CStr(Trabajos(i).wi) & " " &
CStr(Trabajos(i).mi) & " " & CStr(Trabajos(i).li) & " " & CStr(Trabajos(i).di) & " " & CStr(Trabajos(i).ci) & " " & CStr(Trabajos(i).xi) & " " &
CStr(Trabajos(i).ei) & " " & CStr(Trabajos(i).ai) & " " & CStr(Trabajos(i).ni))
```

```
Next i  
Close NumArchivoX  
End Sub
```

```
Sub AbrirR() ' Abrir archivo para la escritura de resultados  
NumArchivoR = FreeFile  
Open App.Path & "\Resultados.txt" For Output As NumArchivoR  
End Sub
```

```
Sub CerrarR()  
Close NumArchivoR  
End Sub
```

9.3 Código estrategia control 1 FSP

Se adjunta en este punto de anexo el código completo utilizado por la estrategia de control de las estrategias 1 en secuenciación FSP para permitir una mejor comprensión del significado que nos referimos cuando hablamos de estrategia de control.

```
Attribute VB_Name = "FSPNoClasesOperacional"
'ESTRATEGIA 1 CONTROL

Option Explicit
Option Base 1
Type DatosJobs
    si As Long
    fi As Integer
    ti As Integer
    wi As Integer 'peso inicial del trabajo
    wwi As Integer 'creado para almacenar el peso original mientras modificamos
    di As Integer 'añadido para especificar la clase de trabajo
    ci As Integer 'añadido para almacenar la clase de máquina que ha resuelto este trabajo, útil para analizar los resultados en estrategia 2
    li As Integer 'vector de ordenacion
    mi As Integer 'añadido para asignar un nuevo indice de identificación en la estructura trabajosM que le quiero pasar a GenerarModelo
    xi As Integer 'Indice que me dira si el trabajo esta realizado o no 1 realizado 0 no realizado
End Type
Dim Trabajos() As DatosJobs
Dim TrabajosM() As DatosJobs 'DataJobs que solo va a contener los trabajos que quiero que haga ese modelo (argumento de GenerarModelo)
Dim NumJobs As Integer 'cantidad total de trabajos
Dim NumJobsM As Integer 'cantidad de trabajos que va a haber en TrabajosM() cada vez
Dim NumMachines As Integer 'cantidad total de máquinas
Dim NumClasJobs As Integer 'añadido para la cantidad de clases de trabajos
Dim NumClasMachines As Integer 'añadido para la cantidad de clases de máquinas
Dim NumArchivoR As Integer

Dim iSolape, iCantidadTrabajos, iTipoL, iCantidadMaquinas, iDistribucionMaquinas, iContador As Integer
Dim Solape(), CantidadTrabajos(), CantidadMaquinas() As Integer

Dim L() As Integer 'Matriz de compatibilidad original
Dim LModMachine() As Integer 'Matriz de compatibilidad modificada para no usar dos veces la misma clase de maquina
Dim LModJob() As Integer 'Matriz de compatibilidad modificada para no priorizar dos veces el mismo trabajo

Dim Maquinas() As Integer 'Almacena el numero de maquinas de cada clase

Dim TipoClaseMaquina As Integer
Dim TipoClaseTrabajo As Integer

Dim PrioJob As Integer 'Variable para devolver el trabajo prioritario cada vez
Dim PrioMachine As Integer 'variable para devolver la maquina prioritaria para cada trabajo

Dim PrioMachines() As Integer

Dim E As Integer 'Peso a sumar para priorizar

Dim MatrizLM(), MatrizLJ1(), MatrizLJ2(), MatrizLJ3() As Integer

Dim iCombMach, iCombJob As Integer 'Para ir recorriendo las matrices MatrizLM, MatrizLJ123; global para imprimir en resultado
Dim incompatibilidad As Integer 'Porque al hacer todas las combinaciones hay muchas posibilidades de que se creen incompatibilidades

Dim FOTotal As Double
Dim NumeroTotalTrabajosRealizados As Double
```

```

Dim IDProblema As Integer

Dim NumArchivo2 As Integer
Public Declare Function GetSystemMenu% Lib "user32.dll" (ByVal hwnd%)

Sub main()

Dim iL As Integer
Dim iter As Integer
Dim NumCombJobs As Integer
Dim FOMax As Double 'Para almacenar el valor maximo de la funcion objetivo y ver en que combinacion ha ocurrido
Dim NumArchivoRes As Integer
Dim BanderaExcepcion As Integer

Dim T1 As Double
Dim T2 As Double
Dim T As Double

ReDim Solape(3)
ReDim CantidadTrabajos(5)
ReDim CantidadMaquinas(3)

Solape(1) = 1
Solape(2) = 2
Solape(3) = 4
CantidadTrabajos(1) = 50
CantidadTrabajos(2) = 100
CantidadTrabajos(3) = 200
CantidadTrabajos(4) = 500
CantidadTrabajos(5) = 1000
CantidadMaquinas(1) = 4
CantidadMaquinas(2) = 8
CantidadMaquinas(3) = 16

E = 1000
'Archivo donde ir haciendo un registro de comprobaciones
NumArchivo2 = FreeFile
Open App.Path & "\Comprobacion.dat" For Output As NumArchivo2
Print #NumArchivo2, "Inicio"

NumArchivoRes = FreeFile
Open App.Path & "\ResultadosMaximos" & ".txt" For Output As NumArchivoRes

LeerMatricesCombinaciones 'Funcion para rellenar las combinaciones de Trabajos y Maquinas

For iSolape = 1 To 3
For iCantidadTrabajos = 1 To 5
For iTipoL = 1 To 3
For iCantidadMaquinas = 1 To 3
For iDistribucionMaquinas = 1 To 4
For iContador = 1 To 10

'Recojo aqui las excepciones ya que no hay problemas para algunas combinaciones que carecen de sentido
If CantidadTrabajos(iCantidadTrabajos) = 50 Then
  If CantidadMaquinas(iCantidadMaquinas) > 4 Then
    BanderaExcepcion = 1
  End If
End If
If CantidadTrabajos(iCantidadTrabajos) = 100 Then
  If CantidadMaquinas(iCantidadMaquinas) > 8 Then
    BanderaExcepcion = 1
  End If

```

```

End If

If BanderaExcepcion = 0 Then

T1 = Timer
FOMax = 0
IDProblema = IDProblema + 1

If iTipoL = 1 Then 'Asigno el numero de ciclos que vamos a tener que hacer para este problema
NumCombJobs = 8
End If
If iTipoL = 2 Then
NumCombJobs = 27
End If
If iTipoL = 3 Then
NumCombJobs = 27
End If

For iCombMach = 1 To 6 'Para cada combinacion de ordenar las maquinas probamos todas las combinaciones de priorizar los trabajos
For iCombJob = 1 To NumCombJobs '(combinaciones de priorizar los trabajos)
LeerArchivoEntrada Solape(iSolape), CantidadTrabajos(iCantidadTrabajos), iTipoL, CantidadMaquinas(iCantidadMaquinas),
iDistribucionMaquinas, iContador
incompatibilidad = 0
FOTotal = 0
NumeroTotalTrabajosRealizados = 0
For iter = 1 To NumClasMachines 'Siempre son tantas iteraciones como máquinas haya
PrioMachine = MatrizLM(iCombMach, iter)
If iTipoL = 1 Then
PrioJob = MatrizLJ1(iCombJob, iter)
End If
If iTipoL = 2 Then
PrioJob = MatrizLJ2(iCombJob, iter)
End If
If iTipoL = 3 Then 'En esta ocasión MatrizLJ3() es igual que MatrizLJ2() pero si los tipos de matriz 3 tuvieran más clases de trabajos
habría que modificarlo
PrioJob = MatrizLJ3(iCombJob, iter)
End If

If L(PrioJob, PrioMachine) = 1 Then 'Solo si es compatible la combinacion continuamos
PriorizarTrabajo PrioJob
For iL = 1 To NumClasJobs 'Forma de dejar recogido que ya se ha usado la maquina
LModJob(iL, PrioMachine) = 0
Next iL

GenerarTrabajosM PrioMachine

ImprimirTrabajos1 'Imprime en un nuevo archivo la matriz trabajos() antes de hacerle nada
ImprimirTrabajosM 'Imprime en un nuevo archivo la matriz trabajosM() antes de hacerle nada

GenerarModelo

If NumJobsM > 0 Then 'Comprobación de que sigue habiendo trabajos disponibles para realizar
ResolverFlujoCosteMinimo
LeerArchivoSalida
ImprimirTrabajos2 'Imprime en un nuevo archivo la matriz trabajos() despues de solucionar el flujo
End If

NormalizarPrioridades
End If
If L(PrioJob, PrioMachine) = 0 Then
incompatibilidad = 1
End If
Next iter
If FOTotal >= FOMax Then 'Imprimimos el resultado solo si es maximo, habrá varios por problema pero el que vale es el ultimo

```

```

    If incompatibilidad = 0 Then

        FOMax = FOTotal
        ImprimirResultado
        Print #NumArchivoRes, (CStr(IDProblema) & " " & CStr(FOTotal) & " " & CStr(iCombMach) & " " & CStr(iCombJob))
    End If
End If
Next iCombJob
Next iCombMach

T2 = Timer
T = T2 - T1
Print #NumArchivo2, (CStr(IDProblema) & " " & CStr(T))

End If
BanderaExcepcion = 0

Next iContador
Next iDistribucionMaquinas
Next iCantidadMaquinas
Next iTipoL
Next iCantidadTrabajos
Next iSolape

End Sub

'Funcion para leer los archivos con las matrices de combinaciones, estos archivos estan cargados con todas las combinaciones
'posibles entre trabajos y maquinas y las guardan en cada matriz MatrizLM MatrizLJ1 MatrizLJ2 MatrizLJ3
'MatrizLM contiene las combinaciones de maquinas (3maquinas no repetibles)
'MatrizLJ1 contiene las combinaciones para priorizacion de trabajos del tipo de problema 1(2trabajos 3maquinas)
'MatrizLJ2 contiene las combinaciones para priorizacion de trabajos del tipo de problema 2(3trabajos 3maquinas)
'MatrizLJ3 contiene las combinaciones para priorizacion de trabajos del tipo de problema 3(4trabajos 3maquinas)
Sub LeerMatricesCombinaciones()
Dim NumArchivo As Integer
Dim i As Integer
Dim Linea As String
Dim Pos As Integer
Dim Izq, Der As String

NumArchivo = FreeFile
Open App.Path & "\MatrizLM" & ".txt" For Input As NumArchivo

ReDim MatrizLM(6, 3)
For i = 1 To 6
    Line Input #NumArchivo, Linea

    Pos = InStr(Linea, " ")
    Izq = Left(Linea, Pos - 1) ' lo que hay a la izquierda
    MatrizLM(i, 1) = CInt(Izq) ' lee el instante de inicio
    Der = Right(Linea, Len(Linea) - Pos) ' lo que hay a la derecha
    Linea = Der

    Pos = InStr(Linea, " ")
    Izq = Left(Linea, Pos - 1) ' lo que hay a la izquierda
    MatrizLM(i, 2) = CInt(Izq) ' lee el instante de inicio
    Der = Right(Linea, Len(Linea) - Pos) ' lo que hay a la derecha

    MatrizLM(i, 3) = Der
Next i

NumArchivo = FreeFile
Open App.Path & "\MatrizLJ1" & ".txt" For Input As NumArchivo

```

```
ReDim MatrizL1(8, 3)
For i = 1 To 8
  Line Input #NumArchivo, Linea

  Pos = InStr(Linea, " ")
  Izq = Left(Linea, Pos - 1) ' lo que hay a la izquierda
  MatrizL1(i, 1) = CInt(Izq) ' lee el instante de inicio
  Der = Right(Linea, Len(Linea) - Pos) ' lo que hay a la derecha
  Linea = Der

  Pos = InStr(Linea, " ")
  Izq = Left(Linea, Pos - 1) ' lo que hay a la izquierda
  MatrizL1(i, 2) = CInt(Izq) ' lee el instante de inicio
  Der = Right(Linea, Len(Linea) - Pos) ' lo que hay a la derecha

  MatrizL1(i, 3) = Der

Next i

NumArchivo = FreeFile
Open App.Path & "\MatrizL2" & ".txt" For Input As NumArchivo

ReDim MatrizL2(27, 3)
For i = 1 To 27
  Line Input #NumArchivo, Linea

  Pos = InStr(Linea, " ")
  Izq = Left(Linea, Pos - 1) ' lo que hay a la izquierda
  MatrizL2(i, 1) = CInt(Izq) ' lee el instante de inicio
  Der = Right(Linea, Len(Linea) - Pos) ' lo que hay a la derecha
  Linea = Der

  Pos = InStr(Linea, " ")
  Izq = Left(Linea, Pos - 1) ' lo que hay a la izquierda
  MatrizL2(i, 2) = CInt(Izq) ' lee el instante de inicio
  Der = Right(Linea, Len(Linea) - Pos) ' lo que hay a la derecha

  MatrizL2(i, 3) = Der

Next i

NumArchivo = FreeFile
Open App.Path & "\MatrizL3" & ".txt" For Input As NumArchivo

ReDim MatrizL3(27, 3)
For i = 1 To 27
  Line Input #NumArchivo, Linea

  Pos = InStr(Linea, " ")
  Izq = Left(Linea, Pos - 1) ' lo que hay a la izquierda
  MatrizL3(i, 1) = CInt(Izq) ' lee el instante de inicio
  Der = Right(Linea, Len(Linea) - Pos) ' lo que hay a la derecha
  Linea = Der

  Pos = InStr(Linea, " ")
  Izq = Left(Linea, Pos - 1) ' lo que hay a la izquierda
  MatrizL3(i, 2) = CInt(Izq) ' lee el instante de inicio
  Der = Right(Linea, Len(Linea) - Pos) ' lo que hay a la derecha

  MatrizL3(i, 3) = Der

Next i
```

End Sub

Sub LeerArchivoEntrada(ByVal Solap As Integer, ByVal NumTrab As Integer, ByVal TipL As Integer, ByVal NumMaquinas As Integer, ByVal DisMaquinas As Integer, ByVal Con As Integer)

Dim NumArchivo As Integer

Dim Linea As String

Dim Indice As Integer

Dim i As Integer 'añadido para leer L

Dim j As Integer 'añadido para leer L

Dim Pos As Integer

Dim Izq As String

Dim Der As String

NumArchivo = FreeFile

Open App.Path & "\bateria2015\Fichero_" & CStr(Solap) & "_" & CStr(NumTrab) & "_" & CStr(TipL) & "_" & CStr(NumMaquinas) & "_" & CStr(DisMaquinas) & "_" & CStr(Con) & ".txt" For Input As NumArchivo

Line Input #NumArchivo, Linea

Linea = Trim(Linea)

NumJobs = CInt(Linea)

If NumJobs <> NumTrab Then 'seguridad de que NumJobs coincide con el argumento NumTrab

Print #NumArchivo2, "Error! NumJobs distinto a NumTrab"

End If

ReDim Trabajos(NumJobs) 'Dimensiona el struct según el número de trabajos

For Indice = 1 To NumJobs

Line Input #NumArchivo, Linea

Linea = Trim(Linea)

Trabajos(Indice).li = Indice

Pos = InStr(Linea, " ")

Izq = Left(Linea, Pos - 1) ' lo que hay a la izquierda

Trabajos(Indice).si = CInt(Izq) ' lee el instante de inicio

Der = Right(Linea, Len(Linea) - Pos) ' lo que hay a la derecha

Linea = Der

Pos = InStr(Linea, " ")

Izq = Left(Linea, Pos - 1)

Trabajos(Indice).fi = CInt(Izq) ' lee el instante de fin del trabajo

Der = Right(Linea, Len(Linea) - Pos) ' lo que hay a la derecha

Linea = Der

Pos = InStr(Linea, " ")

Izq = Left(Linea, Pos - 1)

Trabajos(Indice).ti = CInt(Izq) ' lee el tiempo del trabajo

Der = Right(Linea, Len(Linea) - Pos) ' lo que hay a la derecha

Linea = Der

Pos = InStr(Linea, " ")

Izq = Left(Linea, Pos - 1)

Trabajos(Indice).wi = CInt(Izq) ' lee el peso del trabajo

Trabajos(Indice).wwi = Trabajos(Indice).wi 'para tener una copia no alterada del peso original

Der = Right(Linea, Len(Linea) - Pos) ' lo que hay a la derecha

Trabajos(Indice).di = CInt(Der) 'lo que queda ya es el tipo de trabajo

Trabajos(Indice).xi = 0 'inicio a cero todos los trabajos como no realizados

Next

Line Input #NumArchivo, Linea 'hay una linea vacía

Line Input #NumArchivo, Linea 'leo la cantidad de tipos de trabajos

```

NumClasJobs = Clnt(Linea)

Line Input #NumArchivo, Linea 'leo la cantidad de tipos de maquinas
NumClasMachines = Clnt(Linea)

ReDim L(NumClasJobs, NumClasMachines) 'Creo la matriz L de dimension en función de las variables leidas

For i = 1 To NumClasJobs 'bucles para leer la matriz L en funcion del número de clases de maquinas y de trabajos
  Line Input #NumArchivo, Linea
  For j = 1 To NumClasMachines - 1
    Pos = InStr(Linea, " ")
    Izq = Left(Linea, Pos - 1)
    L(i, j) = Clnt(Izq)
    Der = Right(Linea, Len(Linea) - (Pos + 1)) ' Ojo: hay 2 espacios en la matriz L
    Linea = Der
  Next j
  j = NumClasMachines
  L(i, j) = Clnt(Der)
Next i
LModMachine() = L()
LModJob() = L()

Line Input #NumArchivo, Linea 'leo el número total de maquinas
Linea = Trim(Linea)
NumMachines = Clnt(Linea)

ReDim Maquinas(NumClasMachines) 'Creo el vector de longitud acorde al número de clases de máquinas
For j = 1 To NumClasMachines 'leo cuantas máquinas de cada clase tengo
  Line Input #NumArchivo, Linea
  Maquinas(j) = Clnt(Linea)
Next j

Close NumArchivo
End Sub

'Funcion para decidir las prioridades de los trabajos
Sub DecisionPrioridadesTrabajos()
Dim Sum As Integer
Dim i As Integer
Dim ii As Integer
Dim iii As Integer
Dim j As Integer
Dim Aux() As Integer
Dim SumatorioPesos() As Double 'Aqui voy a almacenar el sumatorio de pesos para el criterio 1.2.1
Dim Min As Integer
Dim Max As Double
Dim Minimios As Integer

ReDim Aux(NumClasJobs)

'1ºGrado criterio, el trabajo más difícil realizable, es el que sume menos valor en la matriz L
Min = 10000
For i = 1 To NumClasJobs
  For j = 1 To NumClasMachines 'Averiguo cuantas clases de máquina son compatibles con mi trabajo
    Aux(i) = Aux(i) + LModJob(i, j)
  Next j
  If Aux(i) < Min Then 'Cuanto es el valor minimo de compatibilidades
    If Aux(i) > 0 Then
      Min = Aux(i)
    End If
  End If
Next i

Minimios = 0

```

```

For i = 1 To NumClasJobs 'Cuantas igualdades tenemos en ese valor minimo?
  If Aux(i) = Min Then
    Minimios = Minimios + 1
  End If
Next i

'2º Criterio, cuando tenemos igualdad/es en el criterio 1
If Minimios > 1 Then
  ReDim SumatorioPesos(NumClasJobs) 'Calculo el sumatoriospesos de cada trabajo que tenga compatibilidad minima
  For i = 1 To NumClasJobs 'Recorro todas las clases de trabajo pero en realidad solo las que coincidan con Min por la siguiente linea
    If Aux(i) = Min Then 'Los que coincidan con Min
      For j = 1 To NumJobs 'Recorro todos los trabajos
        If Trabajos(j).di = i Then 'Acepto los que sean de la clase que estoy mirando
          If Trabajos(j).xi = 0 Then 'Acepto los que no esten realizados
            SumatorioPesos(i) = SumatorioPesos(i) + Trabajos(j).wwi 'OJO: Cuidado de no coger el peso modificado
          End If
        End If
      Next j
    End If
  Next i
  Max = 0
  For ii = 1 To NumClasJobs 'Busco el Maximo de los sumatorios de pesos
    If SumatorioPesos(ii) > Max Then
      Max = SumatorioPesos(ii)
    End If
  Next ii

  For iii = 1 To NumClasJobs 'Busco que trabajo es el que tiene el maximo sumatorio de pesos
    If SumatorioPesos(iii) = Max Then
      PrioJob = iii
    End If
  Next iii

End If
'Si no hemos tenido que entrar en el segundo criterio (solo un minimo) ya sabemos que ese es el trabajo prioritario
If Minimios = 1 Then
  For i = 1 To NumClasJobs
    If Aux(i) = Min Then
      PrioJob = i
    End If
  Next i
End If

'Recordar poner a 0 la columna de LModJob de la máquina que usemos
End Sub

'Funcion para decidir cual de las maquinas es prioritaria para ese trabajo
Sub DecisionPrioridadesMaquinas()
  Dim j As Integer
  Dim i As Integer
  Dim Aux() As Integer
  Dim Min As Integer
  Dim MinMaq As Integer
  Dim Minimios As Integer

  ReDim Aux(NumClasMachines)

  '1º Busco la maquina que sea compatible con menor numero de trabajos
  Min = 1000
  For j = 1 To NumClasMachines 'Sumo en Aux() con cuantos trabajos es compatible cada maquina y el minimo
    If LModJob(PrioJob, j) = 1 Then 'Solo sumo si la maquina es compatible con el trabajo priorizado(lo hago para cada trabajo)

```

```

For i = 1 To NumClasJobs 'Recorro la columna para sumar las compatibilidades
  If LModJob(i, j) = 1 Then
    Aux(j) = Aux(j) + 1
  End If
Next i
If Aux(j) < 1000 Then
  If Aux(j) > 0 Then
    Min = Aux(j)
  End If
End If
End If
Next j

'Busco cuantas maquinas estan en ese valor minimo de compatibilidades
Minimos = 0
For j = 1 To NumClasMachines 'Sumo en Aux() con cuantos trabajos es compatible cada maquina y el minimo
  If Aux(j) = Min Then
    Minimos = Minimos + 1
  End If
Next j

'2º De las clases de maquinas que estan en los mínimos cual tiene más maquinas
MinMaq = 1000

If Minimos > 1 Then
  For j = 1 To NumClasMachines
    If LModJob(PrioJob, j) = 1 Then
      If Maquinas(j) < MinMaq Then
        MinMaq = Maquinas(j)
      End If
    End If
  Next j
  For j = 1 To NumClasMachines
    If LModJob(PrioJob, j) = 1 Then
      If Maquinas(j) = MinMaq Then
        PrioMachine = j
      End If
    End If
  Next j
End If
If Minimos = 1 Then
  For j = 1 To NumClasMachines
    If Aux(j) = Min Then
      PrioMachine = j
    End If
  Next j
End If

End Sub

'Funcion para priorizar los trabajos que queremos en cada iteracion
Sub PriorizarTrabajo(ByVal Trab As Integer)
Dim iTrab As Integer

For iTrab = 1 To NumJobs
  If Trabajos(iTrab).di = Trab Then
    Trabajos(iTrab).wi = Trabajos(iTrab).wi + E
  End If
Next iTrab

End Sub

'Funcion para normalizar las prioridades
Sub NormalizarPrioridades()

```

```

Dim i As Integer

For i = 1 To NumJobs
Trabajos(i).wi = Trabajos(i).wwi
Next i
End Sub

'**** Función creada por mí para hacer un TrabajosM con los trabajos para la clase de maquina que yo diga
'**** TrabajosM() quiero que contenga solo los trabajos que van a ser compatibles con la clase de maquina que recibe
'**** Además en TrabajosM() no van a entrar los trabajos que ya han sido realizados
Sub GenerarTrabajosM(ByVal ClaseMaquina As Integer)

Dim TrabajosMM() As DatosJobs 'Uso una variable local trabajosMM para que se reinicie cada vez que llame a esta función
Dim IndiceOriginal As Integer
Dim IndiceM As Integer
Dim d As Integer

IndiceM = 0
NumMachines = 0

' Qué clases de trabajos son compatibles con la máquina? -> Creo la matriz TrabajosM que sera argumento de GenerarModelo
For IndiceOriginal = 1 To NumJobs
    'Recorrer toda la matriz Trabajos() original
    If Trabajos(IndiceOriginal).xi = 0 Then
        'Si el trabajo ya ha sido realizado no quiero que entre
        For d = 1 To NumClasJobs
            'Recorrer la matriz L para todas las clases de trabajos
            If L(d, ClaseMaquina) = 1 Then
                'Si la clase de trabajo es compatible con la clase de maquina
                If Trabajos(IndiceOriginal).di = d Then
                    'Si el trabajo es de esa clase de trabajo (compatible)
                    IndiceM = IndiceM + 1
                    'Incremento el Indice que me va a decir cuantos trabajos tengo en TrabajosM
                    Trabajos(IndiceOriginal).mi = IndiceM
                    'Almaceno cual va a ser su nuevo indice para poder hacer la conversion desde el
                    archivo Salida.dat
                    ReDim Preserve TrabajosMM(IndiceM)
                    'Hay que redimensionar cada vez la matriz de estructuras conservando los
                    elementos anteriores
                    TrabajosMM(IndiceM) = Trabajos(IndiceOriginal)
                    'Copio el trabajo a mi nueva matriz TrabajosMM
                End If
            End If
        Next d
    End If
Next IndiceOriginal

TrabajosM() = TrabajosMM() 'Cuando termino almaceno en el TrabajosM que es el que voy a usar en los siguientes pasos; funciona si no
hay ningun trabajo compatible??
NumJobsM = IndiceM
'Almaceno una variable que me diga cuantos trabajos tengo en mi TrabajosM() (-1 porque si
NumMachines = Maquinas(ClaseMaquina)
'Almaceno el número de máquinas de esta clase para usarlo en GenerarModelo

End Sub

'**** GenerarModelo Modificada, usa las variables globales TrabajosM() NumJobsM y NumMachinesM y crea el archivo Entrada.dat
Sub GenerarModelo()
Dim NumArchivo As Integer
Dim Linea As String
Dim Cadena As String
Dim Indice As Long
Dim Pos As Integer
Dim i As Integer
Dim j As Integer
Dim K As Integer
Dim L As Integer
Dim Max As Integer
Dim M As Integer
Dim Ind As Long
Dim TrabajosEjecutados(2000) As Integer

'OrdenarPorsi Trabajos() 'Mejor que vengan ordenados, podría alterar la conversión de índices?

```

```

NumArchivo = FreeFile
Open App.Path & "\Entrada.dat" For Output As NumArchivo

' Numero de nodos y de arcos porque hay tantos nodos como segundos máximos (ojo, el segundo 0 no existe); el total de arcos es la
' la suma de los arcos de transporte (segundos Max-1)+los arcos de trabajos (NumJobsM)

Max = 0
For M = 1 To NumJobsM
  If Max < TrabajosM(M).fi Then
    Max = TrabajosM(M).fi
  End If
Next M

Cadena = CStr(Max) & " " & CStr((NumJobsM) + (Max - 1))
Print #NumArchivo, Cadena
Cadena = ""

' lineas de arcos correspondientes a los trabajos

For Indice = 1 To NumJobsM
  Cadena = CStr(TrabajosM(Indice).si) & " " & CStr((TrabajosM(Indice).si + TrabajosM(Indice).ti)) & " " & "1" & " " & "-" &
  CStr(TrabajosM(Indice).wi)
  Print #NumArchivo, Cadena
  Cadena = ""
Next Indice

' lineas de arcos para el desplazamiento temporal

For Indice = 1 To Max - 1
  Cadena = CStr(Indice) & " " & CStr(Indice + 1) & " " & 10000 & " " & 0
  Print #NumArchivo, Cadena
  Cadena = ""
Next Indice

' Unidades de flujo cedidas y absorbidas por cada nodo

Cadena = "-" & CStr(NumMachines)
Print #NumArchivo, Cadena
Cadena = ""
For Indice = 1 To Max - 2
  Cadena = 0
  Print #NumArchivo, Cadena
  Cadena = ""
Next Indice
Cadena = CStr(NumMachines)
Print #NumArchivo, Cadena
Cadena = ""
Close NumArchivo

' Damos una espera hasta que el archivo Entrada.dat este completamente cerrado

While Dir(App.Path & "\Entrada.dat") = " "
  Indice = Indice + 1
Wend

End Sub

'**** Llamada a la funcion Shell para ejecutar Flujo.exe; este programa toma Entrada.dat y devuelve Salida.dat con la solucion del FCM
Sub ResolverFlujoCosteMinimo()

Dim Salida As Integer
Dim i As Integer

```

```

Dim fs As New Scripting.FileSystemObject
Dim F As String

Salida = Shell(App.Path & "\\FLUJO.EXE entrada.dat salida.dat comproba.dat", vbHide)

F = App.Path & "\\comproba.dat"
While fs.FileExists(F) = False

Wend
fs.DeleteFile F

For i = 1 To 10000
Next i

End Sub

'**** Función para interpretar la información de Salida.dat
Sub LeerArchivoSalida() 'Leemos la información del archivo salida (valor de la FO y arcos de trabajo usados solo, los arcos de transporte no me interesan)

'Dim FO As String 'Para almacenar la cadena del valor de la función objetivo
Dim IndiceOriginal As Integer
Dim NumArchivo As Integer
Dim Linea As String
Dim Pos As Integer
Dim Indice As Integer
Dim FO As Double 'Tiene que ser double para que no se produzca desbordamiento

NumArchivo = FreeFile
Open App.Path & "\\Salida.dat" For Input As NumArchivo
Line Input #NumArchivo, Linea 'Linea de la función objetivo
Pos = InStr(Linea, ".")
FO = Cdbl(Left(Linea, Pos - 1)) * -1 'Lo que hay a la izda del punto es el valor de la FO porque siempre es -xxx.00000
For Indice = 1 To NumJobsM 'Leo las primeras "NumJobsM" filas que van a ser las que me digan como estan los trabajos
Line Input #NumArchivo, Linea
If Cint(Linea) = 1 Then 'Si el trabajo se ha realizado aquí
IndiceOriginal = TrabajosM(Indice).li 'La reconversión de indice. El indice original esta intacto en .li
If Trabajos(IndiceOriginal).xi = 1 Then 'Por seguridad
Print #NumArchivo2, "ERROR! El trabajo con este indice ya había sido realizado" & CStr(IndiceOriginal)
End If
If Trabajos(IndiceOriginal).wi > 1000 Then 'Elimino el exceso de peso creado para la priorizacion
FO = FO - 1000
End If
Trabajos(IndiceOriginal).xi = 1 'El trabajo ha sido realizado
Trabajos(IndiceOriginal).ci = PrioMachine 'Almaceno aqui la clase de maquina que ha resuelto este trabajo
NumeroTotalTrabajosRealizados = NumeroTotalTrabajosRealizados + 1 'Comprobación simplemente
End If
Next Indice
Close NumArchivo
FOTotal = FOTotal + FO 'Vamos acumulando el valor de la FO
End Sub

'**** Función complementaria para comprobar el valor de la matriz Trabajos() antes de resolver el FCM
Sub ImprimirTrabajos1()
Dim i As Integer
Dim NumArchivoX As Integer
Dim Cadena As String
NumArchivoX = FreeFile
Open App.Path & "\\Trabajos1.dat" For Output As NumArchivoX
For i = 1 To NumJobs
Cadena = ""
Cadena = CStr(Trabajos(i).si) & " " & CStr(Trabajos(i).fi) & " " & CStr(Trabajos(i).ti) & " " & CStr(Trabajos(i).wi) & " " & CStr(Trabajos(i).di) & " " & CStr(Trabajos(i).li) & " " & CStr(Trabajos(i).mi) & " " & CStr(Trabajos(i).xi)

```

```
Print #NumArchivoX, Cadena
Next i
Close NumArchivoX
End Sub
```

```
**** Función complementaria para comprobar el valor de TrabajosM() que van a entrar para generar el modelo que entrara a FCM
Sub ImprimirTrabajosM()
Dim i As Integer
Dim NumArchivoX As Integer
NumArchivoX = FreeFile
Open App.Path & "\TrabajosM.dat" For Output As NumArchivoX
For i = 1 To NumJobsM
Print #NumArchivoX, (CStr(TrabajosM(i).si) & " " & CStr(TrabajosM(i).fi) & " " & CStr(TrabajosM(i).ti) & " " & CStr(TrabajosM(i).wi) & " " & CStr(TrabajosM(i).di) & " " & CStr(TrabajosM(i).li) & " " & CStr(TrabajosM(i).mi) & " " & CStr(TrabajosM(i).xi))
Next i
Close NumArchivoX
End Sub
```

```
**** Función complementaria para comprobar el valor de Trabajos() después de resolver la iteracion y habiendo interpretado salida.dat
Sub ImprimirTrabajos2()
Dim i As Integer
Dim NumArchivoX As Integer
NumArchivoX = FreeFile
Open App.Path & "\Trabajos2.dat" For Output As NumArchivoX
For i = 1 To NumJobs
Print #NumArchivoX, (CStr(Trabajos(i).si) & " " & CStr(Trabajos(i).fi) & " " & CStr(Trabajos(i).ti) & " " & CStr(Trabajos(i).wi) & " " & CStr(Trabajos(i).di) & " " & CStr(Trabajos(i).li) & " " & CStr(Trabajos(i).mi) & " " & CStr(Trabajos(i).xi))
Next i
Close NumArchivoX
End Sub
```

```
**** Función complementaria para imprimir los resultados
Sub ImprimirResultado()
Dim i As Integer
Dim NumArchivoX As Integer

NumArchivoX = FreeFile
Open App.Path & "\Resultados" & "\Res_Est_1_" & CStr(Solape(iSolape)) & "_" & CStr(CantidadTrabajos(iCantidadTrabajos)) & "_" & CStr(iTipoL) & "_" & CStr(CantidadMaquinas(iCantidadMaquinas)) & "_" & CStr(iDistribucionMaquinas) & "_" & CStr(iContador) & ".txt"
For Output As NumArchivoX
```

```
Print #NumArchivoX, CStr(FOTotal) & " " & CStr(NumeroTotalTrabajosRealizados)
```

```
For i = 1 To NumJobs
Print #NumArchivoX, (CStr(Trabajos(i).si) & " " & CStr(Trabajos(i).fi) & " " & CStr(Trabajos(i).ti) & " " & CStr(Trabajos(i).wi) & " " & CStr(Trabajos(i).mi) & " " & CStr(Trabajos(i).li) & " " & CStr(Trabajos(i).di) & " " & CStr(Trabajos(i).ci) & " " & CStr(Trabajos(i).xi))
Next i
Close NumArchivoX
End Sub
```

```
Sub AbrirR() ' Abrir archivo para la escritura de resultados
NumArchivoR = FreeFile
Open App.Path & "\Resultados.txt" For Output As NumArchivoR
End Sub
```

```
Sub CerrarR()
Close NumArchivoR
End Sub
```


9.4 Mapa de distribución error relativo estrategias FSP

Se adjunta a continuación como anexo el mapa de distribuciones de los errores relativos de cada uno de los problemas al ser resuelto por cada estrategia FSP para una mayor representatividad de los resultados, ya que en las tablas mostradas en el proyecto solo se puede ver el valor promedio y no la distribución de estos valores.

		Comparación Optimos y Est 1																										
		Rango	0%	1%	2%	3%	4%	5%	6%	7%	8%	9%	10%	11%	12%	13%	14%	15%	16%	17%	18%	19%	20%	21%	22%	23%	24%	25%
Total	n° Problemas	94	161	208	256	266	318	328	351	350	374	314	273	250	230	184	144	94	51	22	20	16	9	2	2	2	0	
	% Problemas	2%	4%	5%	6%	6%	7%	8%	8%	8%	9%	7%	6%	6%	5%	4%	3%	2%	1%	1%	0%	0%	0%	0%	0%	0%	0%	0%
Solape	1	n° Problemas	67	99	110	128	92	101	95	124	91	111	94	70	69	56	54	37	24	9	3	5	1	0	0	0	0	0
	% Problemas	5%	7%	8%	9%	6%	7%	7%	9%	6%	8%	7%	5%	5%	4%	4%	3%	2%	1%	0%	0%	0%	0%	0%	0%	0%	0%	0%
	2	n° Problemas	17	36	63	83	82	125	136	112	140	124	113	83	69	77	51	57	32	16	4	7	5	3	2	0	2	0
	% Problemas	1%	3%	4%	6%	6%	9%	9%	8%	10%	9%	8%	6%	5%	5%	4%	4%	2%	1%	0%	0%	0%	0%	0%	0%	0%	0%	0%
	4	n° Problemas	10	26	35	45	92	92	97	115	119	139	107	120	112	97	79	50	38	26	15	8	10	6	0	2	0	0
	% Problemas	1%	2%	2%	3%	6%	6%	7%	8%	8%	10%	7%	8%	8%	7%	5%	3%	3%	2%	1%	1%	1%	1%	0%	0%	0%	0%	0%
Numero Trabajos	50	n° Problemas	22	22	15	17	30	31	21	23	26	34	16	24	17	6	8	10	8	8	4	6	4	3	0	2	2	0
	% Problemas	6%	6%	4%	5%	8%	9%	6%	6%	7%	9%	4%	7%	5%	2%	2%	3%	2%	2%	1%	2%	1%	1%	0%	1%	1%	0%	
	100	n° Problemas	13	21	36	50	36	62	58	77	65	52	53	52	34	37	26	13	12	9	6	5	1	2	0	0	0	0
	% Problemas	2%	3%	5%	7%	5%	9%	8%	11%	9%	7%	7%	7%	5%	5%	4%	2%	2%	1%	1%	1%	0%	0%	0%	0%	0%	0%	0%
	200	n° Problemas	39	47	38	68	62	75	104	102	76	92	67	67	73	49	42	38	14	7	6	0	8	4	2	0	0	0
	% Problemas	4%	4%	4%	6%	6%	7%	10%	9%	7%	9%	6%	6%	7%	5%	4%	4%	1%	1%	1%	0%	1%	0%	0%	0%	0%	0%	0%
500	n° Problemas	12	30	56	65	74	77	71	66	87	89	104	69	78	70	39	39	29	17	1	5	2	0	0	0	0	0	
% Problemas	1%	3%	5%	6%	7%	7%	7%	6%	8%	8%	10%	6%	7%	6%	4%	4%	3%	2%	0%	0%	0%	0%	0%	0%	0%	0%	0%	
1000	n° Problemas	8	41	63	56	64	73	74	83	96	107	74	61	48	68	69	44	31	10	5	4	1	0	0	0	0	0	
% Problemas	1%	4%	6%	5%	6%	7%	7%	8%	9%	10%	7%	6%	4%	6%	6%	4%	3%	1%	0%	0%	0%	0%	0%	0%	0%	0%	0%	
Numero Maquir Tipol	1	n° Problemas	52	64	91	127	106	162	163	148	124	131	84	61	56	29	21	11	4	1	3	1	1	0	0	0	0	0
	% Problemas	4%	4%	6%	9%	7%	11%	11%	10%	9%	9%	6%	4%	4%	2%	1%	1%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
	2	n° Problemas	31	80	103	111	142	130	134	159	131	105	68	62	50	42	35	30	11	8	1	4	0	1	0	2	0	0
	% Problemas	2%	6%	7%	8%	10%	9%	9%	11%	9%	7%	5%	4%	3%	3%	2%	2%	1%	1%	0%	0%	0%	0%	0%	0%	0%	0%	0%
	3	n° Problemas	11	17	14	18	18	26	31	44	95	138	162	150	144	159	128	103	79	42	18	15	15	8	2	0	2	0
	% Problemas	1%	1%	1%	1%	1%	2%	2%	3%	7%	10%	11%	10%	10%	11%	9%	7%	5%	3%	1%	1%	1%	1%	0%	0%	0%	0%	0%
Numero Maquir	4	n° Problemas	29	48	65	108	110	142	130	154	161	165	126	118	113	89	73	55	35	28	13	12	14	5	2	2	2	0
	% Problemas	2%	3%	4%	6%	6%	8%	7%	9%	9%	9%	7%	7%	6%	5%	4%	3%	2%	2%	1%	1%	1%	0%	0%	0%	0%	0%	0%
	8	n° Problemas	10	32	57	67	80	113	121	113	107	126	120	105	89	90	69	65	41	15	7	7	2	4	0	0	0	0
	% Problemas	1%	2%	4%	5%	6%	8%	8%	8%	7%	9%	8%	7%	6%	6%	5%	5%	3%	1%	0%	0%	0%	0%	0%	0%	0%	0%	0%
	16	n° Problemas	55	81	86	81	76	63	77	84	82	83	68	50	48	51	42	24	18	8	2	1	0	0	0	0	0	0
	% Problemas	5%	8%	8%	8%	7%	6%	7%	8%	8%	8%	6%	5%	4%	5%	4%	2%	2%	1%	0%	0%	0%	0%	0%	0%	0%	0%	0%
Distr Maquinas	1	n° Problemas	16	21	15	23	30	38	50	60	71	103	90	98	116	101	96	71	46	15	8	7	2	1	0	0	2	0
	% Problemas	1%	2%	1%	2%	3%	4%	5%	6%	7%	10%	8%	9%	11%	9%	9%	7%	4%	1%	1%	1%	0%	0%	0%	0%	0%	0%	0%
	2	n° Problemas	21	27	39	36	34	66	103	127	131	132	100	84	44	46	34	21	11	8	3	6	4	2	0	0	0	0
	% Problemas	2%	3%	4%	3%	3%	6%	10%	12%	12%	12%	9%	8%	4%	4%	3%	2%	1%	1%	0%	1%	0%	0%	0%	0%	0%	0%	0%
	3	n° Problemas	20	40	75	97	92	108	102	93	85	85	83	51	46	38	18	14	9	11	4	1	5	1	1	1	0	0
	% Problemas	2%	4%	7%	9%	9%	10%	9%	9%	8%	8%	8%	5%	4%	4%	2%	1%	1%	1%	0%	0%	0%	0%	0%	0%	0%	0%	0%
4	n° Problemas	37	73	79	100	110	106	73	71	63	54	41	40	44	45	36	38	28	17	7	6	5	5	1	1	0	0	
% Problemas	3%	7%	7%	9%	10%	10%	7%	7%	6%	5%	4%	4%	4%	4%	4%	3%	4%	3%	2%	1%	1%	0%	0%	0%	0%	0%	0%	

		Comparación Optimos y Est 12																										
		Rango	0%	1%	2%	3%	4%	5%	6%	7%	8%	9%	10%	11%	12%	13%	14%	15%	16%	17%	18%	19%	20%	21%	22%	23%	24%	25%
Total	nº Problemas	84	169	214	210	239	220	284	245	272	269	238	222	249	214	173	166	171	153	99	107	119	72	53	36	16	0	
	% Problemas	2%	4%	5%	5%	6%	5%	7%	6%	6%	6%	6%	5%	6%	5%	4%	4%	4%	4%	2%	2%	3%	2%	1%	1%	0%	0%	
Solape	1	nº Problemas	60	110	108	93	91	60	91	65	81	75	74	59	73	66	39	42	53	43	21	26	33	22	20	15	6	0
	% Problemas	4%	8%	8%	6%	6%	4%	6%	5%	6%	5%	5%	4%	5%	5%	3%	3%	4%	3%	1%	2%	2%	2%	1%	1%	0%	0%	
	2	nº Problemas	18	45	80	68	85	72	88	87	104	108	82	74	70	76	57	52	48	50	37	41	42	27	13	5	6	0
	% Problemas	1%	3%	6%	5%	6%	5%	6%	6%	7%	8%	6%	5%	5%	5%	4%	4%	3%	3%	3%	3%	3%	3%	2%	1%	0%	0%	0%
	4	nº Problemas	6	14	26	49	63	88	105	93	87	86	82	89	106	72	77	72	70	60	41	40	44	23	20	16	4	0
	% Problemas	0%	1%	2%	3%	4%	6%	7%	6%	6%	6%	6%	6%	7%	5%	5%	5%	5%	4%	3%	3%	3%	3%	2%	1%	1%	0%	0%
Numero Trabajos	50	nº Problemas	9	12	12	10	28	24	23	21	31	25	17	25	24	15	9	9	13	11	5	7	8	2	8	1	2	0
	% Problemas	3%	3%	3%	3%	8%	7%	6%	6%	9%	7%	5%	7%	7%	4%	3%	3%	4%	3%	1%	2%	2%	1%	2%	0%	1%	0%	
	100	nº Problemas	9	12	24	24	42	31	58	48	58	41	49	41	38	41	35	30	29	37	22	22	13	7	2	2	1	0
	% Problemas	1%	2%	3%	3%	6%	4%	8%	7%	8%	6%	7%	6%	5%	6%	5%	4%	4%	5%	3%	3%	2%	1%	0%	0%	0%	0%	
	200	nº Problemas	29	49	41	44	56	61	72	60	63	62	57	46	67	63	43	48	39	36	33	26	33	10	17	9	8	0
	% Problemas	3%	5%	4%	4%	5%	6%	7%	6%	6%	6%	6%	5%	4%	6%	6%	4%	4%	4%	3%	3%	2%	3%	1%	2%	1%	1%	0%
500	nº Problemas	14	38	48	70	63	50	72	53	58	73	60	61	67	43	37	36	47	39	21	30	41	20	12	18	4	0	
% Problemas	1%	4%	4%	6%	6%	5%	7%	5%	5%	7%	6%	6%	6%	4%	3%	3%	4%	4%	2%	3%	4%	2%	1%	2%	0%	0%		
1000	nº Problemas	23	58	89	62	50	54	59	63	62	68	55	49	53	52	49	43	43	30	18	22	24	33	14	6	1	0	
% Problemas	2%	5%	8%	6%	5%	5%	5%	6%	6%	6%	6%	5%	5%	5%	5%	5%	4%	4%	3%	2%	2%	2%	3%	1%	1%	0%	0%	
Numero Maquir Tipol	1	nº Problemas	66	95	114	114	128	112	126	106	107	115	82	75	66	51	33	24	15	4	4	0	1	1	0	0	1	0
	% Problemas	5%	7%	8%	8%	9%	8%	9%	7%	7%	8%	6%	5%	5%	4%	2%	2%	1%	0%	0%	0%	0%	0%	0%	0%	0%	0%	
	2	nº Problemas	18	73	100	96	104	99	142	129	125	102	95	74	80	48	39	34	33	19	12	12	1	1	0	0	0	0
	% Problemas	1%	5%	7%	7%	7%	7%	10%	9%	9%	7%	7%	5%	6%	3%	3%	2%	2%	1%	1%	1%	0%	0%	0%	0%	0%	0%	
	3	nº Problemas	0	1	0	0	7	9	16	10	40	52	61	73	103	115	101	108	123	130	83	95	117	70	53	36	15	0
	% Problemas	0%	0%	0%	0%	0%	1%	1%	1%	3%	4%	4%	5%	7%	8%	7%	8%	9%	9%	6%	7%	8%	5%	4%	3%	1%	0%	
Numero Maquir	4	nº Problemas	10	29	63	63	103	117	155	112	152	142	106	106	105	101	64	74	81	60	36	38	29	9	14	8	7	0
	% Problemas	1%	2%	4%	4%	6%	7%	9%	6%	8%	8%	6%	6%	6%	6%	4%	4%	5%	3%	2%	2%	2%	1%	1%	0%	0%	0%	
	8	nº Problemas	16	45	65	71	80	60	87	86	75	77	80	78	95	70	68	60	65	71	45	48	51	18	11	10	5	0
	% Problemas	1%	3%	5%	5%	6%	4%	6%	6%	5%	5%	6%	5%	7%	5%	5%	4%	5%	5%	3%	3%	4%	1%	1%	1%	0%	0%	
	16	nº Problemas	58	95	86	76	56	43	42	47	45	50	52	38	49	43	41	32	25	22	18	21	39	45	28	18	4	0
	% Problemas	5%	9%	8%	7%	5%	4%	4%	4%	4%	5%	5%	4%	5%	4%	4%	3%	2%	2%	2%	2%	4%	4%	3%	2%	0%	0%	
Distr Maquinas	1	nº Problemas	10	17	10	13	18	21	25	33	57	75	73	90	104	93	86	90	59	46	32	34	35	20	19	11	5	0
	% Problemas	1%	2%	1%	1%	2%	2%	2%	3%	5%	7%	7%	8%	10%	9%	8%	8%	5%	4%	3%	3%	3%	2%	2%	1%	0%	0%	
	2	nº Problemas	7	21	34	22	28	37	75	98	102	116	98	82	74	48	26	34	30	32	25	27	20	20	7	6	5	0
	% Problemas	1%	2%	3%	2%	3%	3%	7%	9%	9%	11%	9%	8%	7%	4%	2%	3%	3%	3%	2%	3%	2%	2%	1%	1%	0%	0%	
	3	nº Problemas	16	37	81	81	97	85	105	76	75	48	38	27	40	36	29	24	44	39	17	20	24	16	12	7	1	0
	% Problemas	1%	3%	8%	8%	9%	8%	10%	7%	7%	4%	4%	3%	4%	3%	3%	2%	4%	4%	2%	2%	2%	1%	1%	1%	0%	0%	
4	nº Problemas	51	94	89	94	96	77	79	38	38	30	29	23	31	37	32	18	38	36	25	26	40	16	15	12	5	0	
% Problemas	5%	9%	8%	9%	9%	7%	7%	4%	4%	3%	3%	2%	3%	3%	3%	2%	4%	3%	2%	2%	4%	1%	1%	1%	0%	0%		

		Comparación Optimos y Est 13																													
		Rango					Comparación Optimos y Est 13																								
		0%	1%	2%	3%	4%	5%	6%	7%	8%	9%	10%	11%	12%	13%	14%	15%	16%	17%	18%	19%	20%	21%	22%	23%	24%	25%				
Total	nº Problemas	107	168	219	260	276	321	345	391	389	403	330	276	266	214	157	86	47	32	10	11	2	2	1	6	0	0				
	% Problemas	2%	4%	5%	6%	6%	7%	8%	9%	9%	9%	8%	6%	6%	5%	4%	2%	1%	1%	0%	0%	0%	0%	0%	0%	0%	0%				
Solape	1	nº Problemas	78	106	120	136	97	97	93	133	101	111	97	61	82	48	36	17	13	9	1	3	1	0	0	0	0				
	% Problemas	5%	7%	8%	9%	7%	7%	6%	9%	7%	8%	7%	4%	6%	3%	3%	1%	1%	1%	0%	0%	0%	0%	0%	0%	0%	0%				
	2	nº Problemas	17	36	63	81	85	133	144	137	163	133	104	80	79	72	47	26	18	7	5	5	0	0	0	4	0				
	% Problemas	1%	3%	4%	6%	6%	9%	10%	10%	11%	9%	7%	6%	5%	5%	3%	2%	1%	0%	0%	0%	0%	0%	0%	0%	0%	0%				
	4	nº Problemas	12	26	36	43	94	91	108	121	125	159	129	135	105	94	74	43	16	16	4	3	1	2	1	2	0				
	% Problemas	1%	2%	3%	3%	7%	6%	8%	8%	9%	11%	9%	9%	7%	7%	5%	3%	1%	1%	0%	0%	0%	0%	0%	0%	0%	0%				
Numero Trabajos	50	nº Problemas	24	22	16	15	32	32	24	21	25	33	26	17	21	7	8	6	2	9	4	6	1	2	0	6	0				
	% Problemas	7%	6%	4%	4%	9%	9%	7%	6%	7%	9%	7%	5%	6%	2%	2%	2%	1%	3%	1%	2%	0%	1%	0%	2%	0%					
	100	nº Problemas	13	22	37	54	41	64	64	83	58	56	61	52	39	29	22	8	6	7	2	2	0	0	0	0	0				
	% Problemas	2%	3%	5%	8%	6%	9%	9%	12%	8%	8%	8%	7%	5%	4%	3%	1%	1%	1%	0%	0%	0%	0%	0%	0%	0%	0%				
	200	nº Problemas	50	53	38	64	64	80	110	110	93	106	79	60	60	40	30	19	9	8	3	3	0	0	1	0	0				
	% Problemas	5%	5%	4%	6%	6%	7%	10%	10%	9%	10%	7%	6%	6%	4%	3%	2%	1%	1%	0%	0%	0%	0%	0%	0%	0%	0%				
500	nº Problemas	12	30	65	71	75	72	73	73	97	100	110	77	82	63	39	25	10	5	0	0	1	0	0	0	0					
% Problemas	1%	3%	6%	7%	7%	7%	7%	7%	7%	9%	9%	10%	7%	8%	6%	4%	2%	1%	0%	0%	0%	0%	0%	0%	0%	0%					
1000	nº Problemas	8	41	63	56	64	73	74	104	116	108	54	70	64	75	58	28	20	3	1	0	0	0	0	0	0					
% Problemas	1%	4%	6%	5%	6%	7%	7%	10%	11%	10%	5%	6%	6%	7%	5%	3%	2%	0%	0%	0%	0%	0%	0%	0%	0%	0%					
Numero Maquir Tipol	1	nº Problemas	52	64	91	127	106	162	163	148	124	131	84	61	56	29	21	11	4	1	3	1	1	0	0	0					
	% Problemas	4%	4%	6%	9%	7%	11%	11%	10%	9%	9%	6%	4%	4%	2%	1%	1%	0%	0%	0%	0%	0%	0%	0%	0%	0%					
	2	nº Problemas	31	80	103	111	142	130	134	159	131	105	68	62	50	42	35	30	11	8	1	4	0	1	0	2					
	% Problemas	2%	6%	7%	8%	10%	9%	9%	11%	9%	7%	5%	4%	3%	3%	2%	2%	1%	1%	0%	0%	0%	0%	0%	0%	0%					
	3	nº Problemas	24	24	25	22	28	29	48	84	134	167	178	153	160	143	101	45	32	23	6	6	1	1	1	4					
	% Problemas	2%	2%	2%	2%	2%	2%	3%	6%	9%	12%	12%	11%	11%	10%	7%	3%	2%	2%	0%	0%	0%	0%	0%	0%	0%					
Numero Maquir	4	nº Problemas	31	48	66	107	112	142	134	167	175	177	160	115	118	81	60	37	19	23	6	10	2	2	1						
	% Problemas	2%	3%	4%	6%	6%	8%	7%	9%	10%	10%	9%	6%	7%	5%	3%	2%	1%	1%	0%	1%	0%	0%	0%	0%						
	8	nº Problemas	10	33	58	69	86	116	127	116	126	144	117	120	95	85	64	38	22	9	4	1	0	0	0	0					
	% Problemas	1%	2%	4%	5%	6%	8%	9%	8%	9%	10%	8%	8%	7%	6%	4%	3%	2%	1%	0%	0%	0%	0%	0%	0%	0%					
	16	nº Problemas	66	87	95	84	78	63	84	108	88	82	53	41	53	48	33	11	6	0	0	0	0	0	0	0					
	% Problemas	6%	8%	9%	8%	7%	6%	8%	10%	8%	8%	5%	4%	5%	4%	3%	1%	1%	0%	0%	0%	0%	0%	0%	0%	0%					
Distr Maquinas	1	nº Problemas	19	23	15	25	33	42	52	57	78	105	100	112	124	99	89	52	30	12	7	2	2	2	0						
	% Problemas	2%	2%	1%	2%	3%	4%	5%	5%	7%	10%	9%	10%	11%	9%	8%	5%	3%	1%	1%	0%	0%	0%	0%	0%						
	2	nº Problemas	26	28	42	36	33	66	100	146	146	139	99	77	48	37	22	14	9	3	3	4	0	0	1						
	% Problemas	2%	3%	4%	3%	3%	6%	9%	14%	14%	13%	9%	7%	4%	3%	2%	1%	1%	0%	0%	0%	0%	0%	0%	0%						
	3	nº Problemas	22	42	79	96	96	104	109	108	99	100	76	48	35	29	14	8	2	8	0	2	0	0	0	3					
	% Problemas	2%	4%	7%	9%	9%	10%	10%	10%	9%	9%	7%	4%	3%	3%	1%	1%	0%	1%	0%	0%	0%	0%	0%	0%						
4	nº Problemas	40	75	83	103	114	109	84	80	66	59	55	39	59	49	32	12	6	9	0	3	0	0	0	3						
% Problemas	4%	7%	8%	10%	11%	10%	8%	7%	6%	5%	5%	4%	5%	5%	3%	1%	1%	1%	0%	0%	0%	0%	0%	0%	0%						

		Comparación Optimos y Est 14																										
		Rango	0%	1%	2%	3%	4%	5%	6%	7%	8%	9%	10%	11%	12%	13%	14%	15%	16%	17%	18%	19%	20%	21%	22%	23%	24%	25%
Total	nº Problemas	100	184	239	228	252	231	298	298	322	329	304	278	280	245	199	195	119	94	49	36	15	12	6	2	1	0	
	% Problemas	2%	4%	6%	5%	6%	5%	7%	7%	7%	8%	7%	6%	6%	6%	5%	5%	3%	2%	1%	1%	0%	0%	0%	0%	0%	0%	0%
Solape	1	nº Problemas	74	125	128	106	99	70	84	86	105	100	89	60	70	73	41	49	22	28	9	14	6	1	0	0	1	0
	% Problemas	5%	9%	9%	7%	7%	5%	6%	6%	7%	7%	6%	4%	5%	5%	3%	3%	2%	2%	1%	1%	0%	0%	0%	0%	0%	0%	0%
	2	nº Problemas	18	45	82	69	88	74	101	109	126	122	100	98	93	70	61	64	44	30	23	9	3	5	4	2	0	0
	% Problemas	1%	3%	6%	5%	6%	5%	7%	8%	9%	8%	7%	7%	6%	5%	4%	4%	3%	2%	2%	1%	0%	0%	0%	0%	0%	0%	0%
	4	nº Problemas	8	14	29	53	65	87	113	103	91	107	115	120	117	102	97	82	53	36	17	13	6	6	2	0	0	0
	% Problemas	1%	1%	2%	4%	5%	6%	8%	7%	6%	7%	8%	8%	8%	7%	7%	6%	4%	3%	1%	1%	0%	0%	0%	0%	0%	0%	0%
Numero Trabajos	50	nº Problemas	11	12	17	16	31	25	22	18	30	24	19	26	20	7	8	11	16	13	9	9	3	5	2	2	1	0
	% Problemas	3%	3%	5%	4%	9%	7%	6%	5%	8%	7%	5%	7%	6%	2%	2%	3%	4%	4%	3%	3%	1%	1%	1%	1%	0%	0%	0%
	100	nº Problemas	9	13	28	26	46	40	67	61	67	62	55	49	41	50	26	21	12	21	9	8	3	2	3	0	0	0
	% Problemas	1%	2%	4%	4%	6%	6%	9%	8%	9%	9%	8%	7%	6%	7%	4%	3%	2%	3%	1%	1%	0%	0%	0%	0%	0%	0%	0%
	200	nº Problemas	43	58	46	46	59	65	83	82	81	85	85	64	68	65	35	40	22	21	13	11	4	3	1	0	0	0
	% Problemas	4%	5%	4%	4%	5%	6%	8%	8%	8%	8%	8%	6%	6%	6%	3%	4%	2%	2%	1%	1%	0%	0%	0%	0%	0%	0%	0%
500	nº Problemas	14	43	59	78	66	47	69	56	69	91	81	71	78	60	62	62	34	14	14	5	5	2	0	0	0	0	
% Problemas	1%	4%	5%	7%	6%	4%	6%	5%	6%	8%	8%	7%	7%	6%	6%	6%	3%	1%	1%	0%	0%	0%	0%	0%	0%	0%	0%	
1000	nº Problemas	23	58	89	62	50	54	57	81	75	67	64	68	73	63	68	61	35	25	4	3	0	0	0	0	0	0	
% Problemas	2%	5%	8%	6%	5%	5%	5%	8%	7%	6%	6%	6%	7%	6%	6%	6%	3%	2%	0%	0%	0%	0%	0%	0%	0%	0%	0%	
Numero Maquir	1	nº Problemas	66	95	114	114	128	112	126	106	107	115	82	75	66	51	33	24	15	4	4	0	1	1	0	0	1	0
	% Problemas	5%	7%	8%	8%	9%	8%	9%	7%	7%	8%	6%	5%	5%	4%	2%	2%	1%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
	2	nº Problemas	18	73	100	96	104	99	142	129	125	102	95	74	80	48	39	34	33	19	12	12	1	1	0	0	0	0
	% Problemas	1%	5%	7%	7%	7%	7%	10%	9%	9%	7%	7%	5%	6%	3%	3%	2%	2%	1%	1%	1%	0%	0%	0%	0%	0%	0%	0%
	3	nº Problemas	16	16	25	18	20	20	30	63	90	112	127	129	134	146	127	137	71	71	33	24	13	10	6	2	0	0
	% Problemas	1%	1%	2%	1%	1%	1%	2%	4%	6%	8%	9%	9%	9%	10%	9%	10%	5%	5%	2%	2%	1%	1%	0%	0%	0%	0%	0%
Numero Maquir	4	nº Problemas	12	29	68	69	106	124	158	129	166	163	132	129	103	94	57	71	53	49	30	26	12	9	4	2	1	0
	% Problemas	1%	2%	4%	4%	6%	7%	9%	7%	9%	9%	7%	7%	6%	5%	3%	4%	3%	3%	2%	1%	1%	1%	0%	0%	0%	0%	0%
	8	nº Problemas	16	46	69	73	84	63	92	100	91	102	115	105	113	99	83	85	43	29	15	9	3	3	2	0	0	0
	% Problemas	1%	3%	5%	5%	6%	4%	6%	7%	6%	7%	8%	7%	8%	7%	6%	6%	3%	2%	1%	1%	0%	0%	0%	0%	0%	0%	0%
	16	nº Problemas	72	109	102	86	62	44	48	69	65	64	57	44	64	52	59	39	23	16	4	1	0	0	0	0	0	0
	% Problemas	7%	10%	9%	8%	6%	4%	4%	6%	6%	6%	5%	4%	6%	5%	5%	4%	2%	1%	0%	0%	0%	0%	0%	0%	0%	0%	0%
Distr Maquinas	1	nº Problemas	17	21	21	19	24	24	28	50	74	80	89	101	125	110	92	81	53	23	22	14	4	4	1	0	1	0
	% Problemas	2%	2%	2%	2%	2%	2%	3%	5%	7%	7%	8%	9%	12%	10%	9%	8%	5%	2%	2%	1%	0%	0%	0%	0%	0%	0%	0%
	2	nº Problemas	9	20	37	25	28	40	72	109	114	127	113	88	69	51	50	50	23	25	13	5	6	3	3	0	0	0
	% Problemas	1%	2%	3%	2%	3%	4%	7%	10%	11%	12%	10%	8%	6%	5%	5%	5%	2%	2%	1%	0%	1%	0%	0%	0%	0%	0%	0%
	3	nº Problemas	18	40	84	88	100	86	109	89	86	73	55	43	31	39	27	32	26	28	8	9	2	3	2	1	0	0
	% Problemas	2%	4%	8%	8%	9%	8%	10%	8%	8%	7%	5%	4%	3%	4%	3%	3%	2%	3%	1%	1%	0%	0%	0%	0%	0%	0%	0%
4	nº Problemas	56	103	97	96	100	81	89	50	48	49	47	46	55	45	30	32	17	18	6	8	3	2	0	1	0	0	
% Problemas	5%	10%	9%	9%	9%	8%	8%	5%	4%	5%	4%	4%	5%	4%	3%	3%	2%	2%	1%	1%	0%	0%	0%	0%	0%	0%	0%	

		Comparación Optimos y Est 2																										
		Rango	0%	1%	2%	3%	4%	5%	6%	7%	8%	9%	10%	11%	12%	13%	14%	15%	16%	17%	18%	19%	20%	21%	22%	23%	24%	25%
Total	nº Problemas	1566	1137	649	404	227	150	86	41	30	17	10	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	% Problemas	36%	26%	15%	9%	5%	3%	2%	1%	1%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
Solape	1	nº Problemas	687	302	155	102	67	54	35	14	7	8	7	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	% Problemas	48%	21%	11%	7%	5%	4%	2%	1%	0%	1%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
	2	nº Problemas	542	372	203	127	80	50	30	18	8	6	3	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	% Problemas	38%	26%	14%	9%	6%	3%	2%	1%	1%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
	4	nº Problemas	337	463	291	175	80	46	21	9	15	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	% Problemas	23%	32%	20%	12%	6%	3%	1%	1%	1%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
Numero Trabajos	50	nº Problemas	96	58	34	47	38	29	21	20	10	5	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	% Problemas	27%	16%	9%	13%	11%	8%	6%	6%	3%	1%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
	100	nº Problemas	121	159	143	107	74	56	29	12	12	4	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	% Problemas	17%	22%	20%	15%	10%	8%	4%	2%	2%	1%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
	200	nº Problemas	261	302	199	152	79	45	20	6	7	5	3	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	% Problemas	24%	28%	18%	14%	7%	4%	2%	1%	1%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
Numero Maquir	500	nº Problemas	387	357	206	76	24	12	8	2	1	3	3	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	% Problemas	36%	33%	19%	7%	2%	1%	1%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
	1000	nº Problemas	701	261	67	22	12	8	8	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	% Problemas	65%	24%	6%	2%	1%	1%	1%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
	1	nº Problemas	686	389	177	111	40	21	6	8	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	% Problemas	48%	27%	12%	8%	3%	1%	0%	1%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
Distr Maquinas	2	nº Problemas	693	405	154	81	35	40	17	3	8	3	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	% Problemas	48%	28%	11%	6%	2%	3%	1%	0%	1%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
	3	nº Problemas	187	343	318	212	152	89	63	30	20	14	9	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	% Problemas	13%	24%	22%	15%	11%	6%	4%	2%	1%	1%	1%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
	4	nº Problemas	549	453	284	200	122	87	55	25	17	6	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	% Problemas	31%	25%	16%	11%	7%	5%	3%	1%	1%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
Tipol	8	nº Problemas	472	420	260	148	65	41	11	9	7	4	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	% Problemas	33%	29%	18%	10%	5%	3%	1%	1%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
	16	nº Problemas	545	264	105	56	40	22	20	7	6	7	6	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	% Problemas	50%	24%	10%	5%	4%	2%	2%	1%	1%	1%	1%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
	1	nº Problemas	378	270	146	79	54	50	41	23	17	12	8	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	% Problemas	35%	25%	14%	7%	5%	5%	4%	2%	2%	1%	1%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
Distr Maquinas	2	nº Problemas	299	282	178	134	92	51	21	8	7	5	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	% Problemas	28%	26%	16%	12%	9%	5%	2%	1%	1%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
	3	nº Problemas	355	337	194	104	43	27	12	5	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	% Problemas	33%	31%	18%	10%	4%	3%	1%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
4	nº Problemas	534	248	131	87	38	22	12	5	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
% Problemas	49%	23%	12%	8%	4%	2%	1%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	

		Comparación Optimos y Est 22																									
		Rango					Comparación Optimos y Est 22																				
		0%	1%	2%	3%	4%	5%	6%	7%	8%	9%	10%	11%	12%	13%	14%	15%	16%	17%	18%	19%	20%	21%	22%	23%	24%	25%
Total	nº Problemas	385	828	859	601	433	313	246	166	96	99	78	33	43	43	31	29	15	18	1	2	1	0	0	0	0	0
	% Problemas	9%	19%	20%	14%	10%	7%	6%	4%	2%	2%	2%	1%	1%	1%	1%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
Solape	1	nº Problemas	143	254	242	184	141	102	95	62	34	40	26	19	24	25	13	14	8	10	1	2	1	0	0	0	0
		% Problemas	10%	18%	17%	13%	10%	7%	7%	4%	2%	3%	2%	1%	2%	2%	1%	1%	1%	1%	0%	0%	0%	0%	0%	0%	0%
	2	nº Problemas	126	290	295	184	134	114	77	62	29	28	29	9	12	11	15	12	5	8	0	0	0	0	0	0	0
		% Problemas	9%	20%	21%	13%	9%	8%	5%	4%	2%	2%	2%	1%	1%	1%	1%	0%	1%	0%	0%	0%	0%	0%	0%	0%	0%
	4	nº Problemas	116	284	322	233	158	97	74	42	33	31	23	5	7	7	3	3	2	0	0	0	0	0	0	0	0
		% Problemas	8%	20%	22%	16%	11%	7%	5%	3%	2%	2%	2%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
Numero Trabajos	50	nº Problemas	71	20	41	40	48	35	30	23	12	12	11	5	6	2	0	0	2	2	0	0	0	0	0	0	0
		% Problemas	20%	6%	11%	11%	13%	10%	8%	6%	3%	3%	3%	1%	2%	1%	0%	0%	1%	1%	0%	0%	0%	0%	0%	0%	0%
	100	nº Problemas	54	89	117	92	85	76	71	39	31	17	21	6	7	7	5	3	0	0	0	0	0	0	0	0	0
		% Problemas	8%	12%	16%	13%	12%	11%	10%	5%	4%	2%	3%	1%	1%	1%	1%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
	200	nº Problemas	64	136	183	185	142	100	70	51	26	26	26	11	9	12	15	13	5	5	0	1	0	0	0	0	0
		% Problemas	6%	13%	17%	17%	13%	9%	6%	5%	2%	2%	2%	1%	1%	1%	1%	1%	0%	0%	0%	0%	0%	0%	0%	0%	0%
	500	nº Problemas	44	190	298	190	96	63	44	32	15	31	17	8	9	10	3	10	6	11	1	1	1	0	0	0	0
		% Problemas	4%	18%	28%	18%	9%	6%	4%	3%	1%	3%	2%	1%	1%	1%	0%	1%	1%	1%	0%	0%	0%	0%	0%	0%	0%
	1000	nº Problemas	152	393	220	94	62	39	31	21	12	13	3	3	12	12	8	3	2	0	0	0	0	0	0	0	0
		% Problemas	14%	36%	20%	9%	6%	4%	3%	2%	1%	1%	0%	0%	1%	1%	1%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
Numero Maquir	1	nº Problemas	164	371	377	231	137	83	46	15	9	2	5	0	0	0	0	0	0	0	0	0	0	0	0	0	
		% Problemas	11%	26%	26%	16%	10%	6%	3%	1%	1%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	
	2	nº Problemas	34	114	164	158	144	141	137	121	67	83	64	30	43	43	31	29	15	18	1	2	1	0	0	0	
		% Problemas	2%	8%	11%	11%	10%	10%	10%	8%	5%	6%	4%	2%	3%	3%	2%	2%	1%	1%	0%	0%	0%	0%	0%	0%	0%
	3	nº Problemas	187	343	318	212	152	89	63	30	20	14	9	3	0	0	0	0	0	0	0	0	0	0	0	0	
		% Problemas	13%	24%	22%	15%	11%	6%	4%	2%	1%	1%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	
Numero Maquir	4	nº Problemas	211	323	338	260	211	150	124	67	38	27	27	8	6	4	2	0	2	2	0	0	0	0	0	0	
		% Problemas	12%	18%	19%	14%	12%	8%	7%	4%	2%	2%	2%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	
	8	nº Problemas	75	311	321	209	145	100	75	59	35	27	28	13	15	13	7	6	0	0	0	1	0	0	0	0	
		% Problemas	5%	22%	22%	15%	10%	7%	5%	4%	2%	2%	2%	1%	1%	1%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	
	16	nº Problemas	99	194	200	132	77	63	47	40	23	45	23	12	22	26	22	23	13	16	1	1	1	0	0	0	
		% Problemas	9%	18%	19%	12%	7%	6%	4%	4%	2%	4%	2%	1%	2%	2%	2%	2%	1%	1%	0%	0%	0%	0%	0%	0%	
Distr Maquinas	1	nº Problemas	132	313	221	145	83	66	52	28	15	12	8	3	1	1	0	0	0	0	0	0	0	0	0	0	
		% Problemas	12%	29%	20%	13%	8%	6%	5%	3%	1%	1%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	
	2	nº Problemas	65	175	225	166	135	105	64	58	35	28	15	7	1	1	0	0	0	0	0	0	0	0	0	0	
		% Problemas	6%	16%	21%	15%	13%	10%	6%	5%	3%	3%	1%	1%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	
	3	nº Problemas	70	179	227	159	101	64	57	38	23	25	33	11	15	23	16	18	7	12	1	0	1	0	0	0	
		% Problemas	6%	17%	21%	15%	9%	6%	5%	4%	2%	2%	3%	1%	1%	2%	1%	2%	1%	1%	0%	0%	0%	0%	0%	0%	
4	nº Problemas	118	161	186	131	114	78	73	42	23	34	22	12	26	18	15	11	8	6	0	2	0	0	0	0		
	% Problemas	11%	15%	17%	12%	11%	7%	7%	4%	2%	3%	2%	1%	2%	2%	1%	1%	1%	1%	0%	0%	0%	0%	0%	0%		

		Comparación Optimos y Max Est 2																										
		Rango					Comparación Optimos y Max Est 2																					
		0%	1%	2%	3%	4%	5%	6%	7%	8%	9%	10%	11%	12%	13%	14%	15%	16%	17%	18%	19%	20%	21%	22%	23%	24%	25%	
Total	nº Problemas	2302	1310	526	139	35	5	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	% Problemas	53%	30%	12%	3%	1%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	
Solape	1	nº Problemas	922	334	137	36	9	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	% Problemas	64%	23%	10%	3%	1%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	
	2	nº Problemas	765	443	158	58	14	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	% Problemas	53%	31%	11%	4%	1%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	
	4	nº Problemas	615	533	231	45	12	2	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	% Problemas	43%	37%	16%	3%	1%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	
Numero Trabajos	50	nº Problemas	237	57	38	19	7	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	% Problemas	66%	16%	11%	5%	2%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	
	100	nº Problemas	269	213	153	58	21	4	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	% Problemas	37%	30%	21%	8%	3%	1%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	
	200	nº Problemas	402	406	209	56	7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	% Problemas	37%	38%	19%	5%	1%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	
Numero Maquir	500	nº Problemas	510	440	124	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	% Problemas	47%	41%	11%	1%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	
	1000	nº Problemas	884	194	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	% Problemas	82%	18%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	
	Tipol	1	nº Problemas	824	388	170	48	9	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		% Problemas	57%	27%	12%	3%	1%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
2		nº Problemas	847	395	137	43	11	4	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
% Problemas		59%	27%	10%	3%	1%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	
3		nº Problemas	631	527	219	48	15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
% Problemas		44%	37%	15%	3%	1%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	
Distr Maquinas	4	nº Problemas	933	505	241	83	31	4	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	% Problemas	52%	28%	13%	5%	2%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	
	8	nº Problemas	669	493	218	55	4	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	% Problemas	46%	34%	15%	4%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	
	16	nº Problemas	700	312	67	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	% Problemas	65%	29%	6%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	
Distr Maquinas	1	nº Problemas	662	300	86	21	10	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	% Problemas	61%	28%	8%	2%	1%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	
	2	nº Problemas	423	402	190	53	8	3	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	% Problemas	39%	37%	18%	5%	1%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	
	3	nº Problemas	481	370	174	43	10	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	% Problemas	45%	34%	16%	4%	1%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	
4	nº Problemas	736	238	76	22	7	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
% Problemas	68%	22%	7%	2%	1%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%		

9.5 Mapa de distribución error relativo estrategias VSP

Se adjunta a continuación como anexo el mapa de distribuciones de los errores relativos de cada uno de los problemas al ser resuelto por cada estrategia VSP para una mayor representatividad de los resultados, ya que en las tablas mostradas en el proyecto solo se puede ver el valor promedio y no la distribución de estos valores

		Comparación Optimos y Est 1																										
		Rango	0%	1%	2%	3%	4%	5%	6%	7%	8%	9%	10%	11%	12%	13%	14%	15%	16%	17%	18%	19%	20%	21%	22%	23%	24%	25%
Total	nº Problemas	25	17	30	59	70	86	93	87	100	85	83	72	80	48	42	22	24	15	8	10	6	8	4	2	2	0	
	% Problemas	2%	2%	3%	5%	6%	8%	9%	8%	9%	8%	8%	7%	7%	4%	4%	2%	2%	1%	1%	1%	1%	1%	0%	0%	0%	0%	
Numero Trab Solape	1	nº Problemas	2	3	9	18	25	45	37	28	43	32	37	20	21	13	11	5	3	2	1	1	2	0	2	0	0	
		% Problemas	1%	1%	3%	5%	7%	13%	10%	8%	12%	9%	10%	6%	6%	4%	3%	1%	1%	1%	0%	0%	1%	0%	1%	0%	0%	
	2	nº Problemas	10	6	6	16	20	22	27	36	32	29	22	30	32	21	15	11	8	8	2	2	0	1	1	0	2	
		% Problemas	3%	2%	2%	4%	6%	6%	8%	10%	9%	8%	6%	8%	9%	6%	4%	3%	2%	2%	1%	1%	0%	0%	0%	0%	1%	
	4	nº Problemas	13	8	15	25	25	19	29	23	25	24	24	22	27	14	16	6	13	5	5	7	4	7	1	2		
		% Problemas	4%	2%	4%	7%	7%	5%	8%	6%	7%	7%	7%	6%	8%	4%	4%	2%	4%	1%	1%	2%	1%	2%	0%	1%		
Numero Trab	50	nº Problemas	22	12	20	27	36	42	27	17	25	23	19	17	21	11	11	1	9	4	4	3	1	2	3			
	% Problemas	6%	3%	6%	8%	10%	12%	8%	5%	7%	6%	5%	5%	6%	3%	3%	0%	3%	1%	1%	1%	0%	1%	1%	0%			
100	nº Problemas	3	5	10	32	34	44	66	70	75	62	64	55	59	37	31	21	15	11	4	7	5	6	1				
	% Problemas	0%	1%	1%	4%	5%	6%	9%	10%	10%	9%	9%	8%	8%	5%	4%	3%	2%	2%	1%	1%	1%	1%	0%	0%			
Numero Maq Tipol	1	nº Problemas	4	7	11	27	32	37	43	33	43	30	28	23	15	12	9	1	3	1	0	0	0	1	0			
		% Problemas	1%	2%	3%	8%	9%	10%	12%	9%	12%	8%	8%	6%	4%	3%	3%	0%	1%	0%	0%	0%	0%	0%	0%	0%		
	2	nº Problemas	16	8	15	25	24	40	37	43	34	36	19	11	20	13	6	4	5	1	1	1	0	1	0			
		% Problemas	4%	2%	4%	7%	7%	11%	10%	12%	9%	10%	5%	3%	6%	4%	2%	1%	1%	0%	0%	0%	0%	0%	0%	0%		
	3	nº Problemas	5	2	4	7	14	9	13	11	23	19	36	38	45	23	27	17	16	13	7	9	6	6	4			
		% Problemas	1%	1%	1%	2%	4%	3%	4%	3%	6%	5%	10%	11%	13%	6%	8%	5%	4%	4%	2%	3%	2%	2%	1%	1%		
Numero Maq	4	nº Problemas	24	14	26	44	47	60	56	47	67	57	54	46	50	28	24	17	17	9	7	4	6	7				
	% Problemas	3%	2%	4%	6%	7%	8%	8%	7%	9%	8%	8%	6%	7%	4%	3%	2%	2%	1%	1%	1%	1%	1%	0%				
8	nº Problemas	1	3	4	15	23	26	37	40	33	28	29	26	30	20	18	5	7	6	1	6	0	1	0				
	% Problemas	0%	1%	1%	4%	6%	7%	10%	11%	9%	8%	8%	7%	8%	6%	5%	1%	2%	2%	0%	2%	0%	0%	0%				
Distr Maquinas	1	nº Problemas	3	5	6	10	7	19	21	23	28	21	23	23	28	14	11	6	5	4	3	3	3	3				
		% Problemas	1%	2%	2%	4%	3%	7%	8%	9%	10%	8%	9%	9%	10%	5%	4%	2%	2%	1%	1%	1%	1%	0%				
	2	nº Problemas	1	5	6	6	19	23	21	22	27	20	22	17	25	17	10	5	6	4	3	5	1	3				
		% Problemas	0%	2%	2%	2%	7%	9%	8%	8%	10%	7%	8%	6%	9%	6%	4%	2%	2%	1%	1%	2%	0%	1%				
	3	nº Problemas	11	2	9	18	21	22	25	22	25	24	19	19	11	10	10	5	8	2	1	0	1	1				
		% Problemas	4%	1%	3%	7%	8%	8%	9%	8%	9%	9%	7%	7%	4%	4%	4%	2%	3%	1%	0%	0%	0%	0%				
	4	nº Problemas	10	5	9	25	23	22	26	20	20	20	19	13	16	7	11	6	5	5	1	2	1	1				
		% Problemas	4%	2%	3%	9%	9%	8%	10%	7%	7%	7%	7%	5%	6%	3%	4%	2%	2%	2%	0%	1%	0%	0%				

		Comparación Optimos y Est 12																										
		Rango	0%	1%	2%	3%	4%	5%	6%	7%	8%	9%	10%	11%	12%	13%	14%	15%	16%	17%	18%	19%	20%	21%	22%	23%	24%	25%
Total	nº Problemas	10	13	24	27	46	51	72	96	66	87	65	82	64	80	48	47	46	38	34	22	17	16	8	5	4	0	
	% Problemas	1%	1%	2%	3%	4%	5%	7%	9%	6%	8%	6%	8%	6%	7%	4%	4%	4%	4%	3%	2%	2%	1%	1%	0%	0%	0%	
Solape	1	nº Problemas	1	6	10	6	18	16	24	34	31	33	21	33	14	25	14	15	11	11	11	10	5	7	2	0	1	0
		% Problemas	0%	2%	3%	2%	5%	4%	7%	9%	9%	9%	6%	9%	4%	7%	4%	4%	3%	3%	3%	3%	1%	2%	1%	0%	0%	0%
	2	nº Problemas	6	2	8	7	12	17	24	33	12	33	21	22	24	33	17	20	18	12	12	7	6	2	2	2	0	0
		% Problemas	2%	1%	2%	2%	3%	5%	7%	9%	3%	9%	6%	6%	7%	9%	5%	6%	5%	3%	3%	2%	2%	1%	1%	1%	0%	0%
	4	nº Problemas	3	5	6	14	16	18	24	29	23	21	23	27	26	22	17	12	17	15	11	5	6	7	4	3	3	0
		% Problemas	1%	1%	2%	4%	4%	5%	7%	8%	6%	6%	6%	8%	7%	6%	5%	3%	5%	4%	3%	1%	2%	2%	1%	1%	1%	0%
Numero Trab	50	nº Problemas	9	7	13	15	16	20	26	28	27	37	22	20	22	20	7	15	16	9	5	2	2	8	4	0	2	0
		% Problemas	3%	2%	4%	4%	4%	6%	7%	8%	8%	10%	6%	6%	6%	6%	2%	4%	4%	3%	1%	1%	1%	2%	1%	0%	1%	0%
	100	nº Problemas	1	6	11	12	30	31	46	68	39	50	43	62	42	60	41	32	30	29	29	20	15	8	4	5	2	0
		% Problemas	0%	1%	2%	2%	4%	4%	6%	9%	5%	7%	6%	9%	6%	8%	6%	4%	4%	4%	4%	3%	2%	1%	1%	1%	0%	0%
Numero Mac	1	nº Problemas	5	9	15	12	27	27	36	55	24	30	29	27	15	22	8	8	3	6	2	0	0	0	0	0	0	0
		% Problemas	1%	3%	4%	3%	8%	8%	10%	15%	7%	8%	8%	8%	4%	6%	2%	2%	1%	2%	1%	0%	0%	0%	0%	0%	0%	0%
	2	nº Problemas	5	4	7	11	12	19	31	34	35	36	28	31	26	26	13	11	16	5	4	2	2	0	1	0	0	0
		% Problemas	1%	1%	2%	3%	3%	5%	9%	9%	10%	10%	8%	9%	7%	7%	4%	3%	4%	1%	1%	1%	1%	0%	0%	0%	0%	0%
	3	nº Problemas	0	0	2	4	7	5	5	7	7	21	8	24	23	32	27	28	27	27	28	20	15	16	7	5	4	0
		% Problemas	0%	0%	1%	1%	2%	1%	1%	2%	2%	6%	2%	7%	6%	9%	8%	8%	8%	8%	8%	6%	4%	4%	2%	1%	1%	0%
Numero Mac	4	nº Problemas	9	8	15	19	26	36	45	68	45	67	44	52	44	64	32	26	30	20	17	14	8	11	6	2	3	0
		% Problemas	1%	1%	2%	3%	4%	5%	6%	9%	6%	9%	6%	7%	6%	9%	4%	4%	4%	3%	2%	2%	1%	2%	1%	0%	0%	0%
	8	nº Problemas	1	5	9	8	20	15	27	28	21	20	21	30	20	16	16	21	16	18	17	8	9	5	2	3	1	0
		% Problemas	0%	1%	3%	2%	6%	4%	8%	8%	6%	6%	6%	8%	6%	4%	4%	6%	4%	5%	5%	2%	3%	1%	1%	1%	0%	0%
Distr Maquinas	1	nº Problemas	1	2	3	1	3	7	9	10	19	29	20	21	22	28	10	17	13	17	12	9	7	4	2	0	1	0
		% Problemas	0%	1%	1%	0%	1%	3%	3%	4%	7%	11%	7%	8%	8%	10%	4%	6%	5%	6%	4%	3%	3%	1%	1%	0%	0%	0%
	2	nº Problemas	0	2	4	7	8	7	14	28	18	24	22	17	18	22	12	10	13	8	10	5	6	2	5	1	2	0
		% Problemas	0%	1%	1%	3%	3%	3%	5%	10%	7%	9%	8%	6%	7%	8%	4%	4%	5%	3%	4%	2%	2%	1%	2%	0%	1%	0%
	3	nº Problemas	4	3	6	8	16	18	24	29	15	19	15	25	13	17	14	8	11	6	4	4	3	4	0	1	1	0
		% Problemas	1%	1%	2%	3%	6%	7%	9%	11%	6%	7%	6%	9%	5%	6%	5%	3%	4%	2%	1%	1%	1%	1%	0%	0%	0%	0%
4	nº Problemas	5	6	11	11	19	19	25	29	14	15	8	19	11	13	12	12	9	7	8	4	1	6	1	3	0	0	
	% Problemas	2%	2%	4%	4%	7%	7%	9%	11%	5%	6%	3%	7%	4%	5%	4%	4%	3%	3%	3%	1%	0%	2%	0%	1%	0%	0%	

		Comparación Optimos y Est 13																											
		Rango	0%	1%	2%	3%	4%	5%	6%	7%	8%	9%	10%	11%	12%	13%	14%	15%	16%	17%	18%	19%	20%	21%	22%	23%	24%	25%	
Total	nº Problemas	22	16	32	61	73	92	101	88	107	101	79	75	77	51	33	16	17	14	4	9	7	2	2	0	0	0		
	% Problemas	2%	1%	3%	6%	7%	9%	9%	8%	10%	9%	7%	7%	7%	5%	3%	1%	2%	1%	0%	1%	1%	0%	0%	0%	0%	0%		
Numero Trab Solape	1	nº Problemas	2	3	9	19	27	53	46	30	37	31	30	26	22	9	8	2	3	2	1	0	0	0	0	0	0		
	% Problemas	1%	1%	3%	5%	8%	15%	13%	8%	10%	9%	8%	7%	6%	3%	2%	1%	1%	1%	0%	0%	0%	0%	0%	0%	0%	0%		
	2	nº Problemas	10	6	6	17	18	21	28	35	36	37	21	31	30	27	9	10	3	4	0	5	3	0	2	0	0	0	
	% Problemas	3%	2%	2%	5%	5%	6%	8%	10%	10%	10%	6%	9%	8%	8%	3%	3%	1%	1%	0%	1%	1%	0%	1%	0%	0%	0%	0%	
	4	nº Problemas	10	7	17	25	28	18	27	23	34	33	28	18	25	15	16	4	11	8	3	4	4	2	0	0	0	0	
	% Problemas	3%	2%	5%	7%	8%	5%	8%	6%	9%	9%	8%	5%	7%	4%	4%	1%	3%	2%	1%	1%	1%	1%	1%	0%	0%	0%	0%	
Numero Trab	50	nº Problemas	19	11	22	26	35	42	28	17	32	28	18	17	17	13	11	1	6	5	1	2	4	2	2	0	0	0	
	% Problemas	5%	3%	6%	7%	10%	12%	8%	5%	9%	8%	5%	5%	5%	4%	3%	0%	2%	1%	0%	1%	1%	1%	1%	1%	0%	0%	0%	
100	nº Problemas	3	5	10	35	38	50	73	71	75	73	61	58	60	38	22	15	11	9	3	7	3	0	0	0	0	0		
% Problemas	0%	1%	1%	5%	5%	7%	10%	10%	10%	10%	8%	8%	8%	5%	3%	2%	2%	1%	0%	1%	0%	0%	0%	0%	0%	0%	0%		
Numero Maq Tipol	1	nº Problemas	4	7	11	27	32	37	43	33	43	30	28	23	15	12	9	1	3	1	0	0	0	1	0	0	0	0	
	% Problemas	1%	2%	3%	8%	9%	10%	12%	9%	12%	8%	8%	6%	4%	3%	3%	0%	1%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	
	2	nº Problemas	16	8	15	25	24	40	37	43	34	36	19	11	20	13	6	4	5	1	1	1	0	1	0	0	0	0	
	% Problemas	4%	2%	4%	7%	7%	11%	10%	12%	9%	10%	5%	3%	6%	4%	2%	1%	1%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	
	3	nº Problemas	2	1	6	9	17	15	21	12	30	35	32	41	42	26	18	11	9	12	3	8	7	0	2	0	0	0	0
	% Problemas	1%	0%	2%	3%	5%	4%	6%	3%	8%	10%	9%	11%	12%	7%	5%	3%	3%	3%	1%	2%	2%	0%	1%	0%	0%	0%	0%	
Numero Maq	4	nº Problemas	21	13	28	45	46	60	57	46	75	68	48	48	49	32	24	13	14	9	4	8	7	2	2	0	0	0	
	% Problemas	3%	2%	4%	6%	6%	8%	8%	6%	10%	9%	7%	7%	7%	4%	3%	2%	2%	1%	1%	1%	1%	0%	0%	0%	0%	0%	0%	
8	nº Problemas	1	3	4	16	27	32	44	42	32	33	31	27	28	19	9	3	3	5	0	1	0	0	0	0	0	0	0	
% Problemas	0%	1%	1%	4%	8%	9%	12%	12%	9%	9%	9%	9%	8%	8%	5%	3%	1%	1%	1%	0%	0%	0%	0%	0%	0%	0%	0%	0%	
Distr Maquinas	1	nº Problemas	3	5	6	11	7	23	23	22	28	25	24	26	28	13	9	3	4	2	2	0	4	1	0	0	0	0	
	% Problemas	1%	2%	2%	4%	3%	9%	9%	8%	10%	9%	9%	10%	10%	5%	3%	1%	1%	1%	1%	0%	1%	0%	0%	0%	0%	0%	0%	
	2	nº Problemas	2	4	6	7	23	24	21	22	28	23	23	18	18	17	8	6	5	6	2	5	1	1	0	0	0	0	0
	% Problemas	1%	1%	2%	3%	9%	9%	8%	8%	10%	9%	9%	7%	7%	6%	3%	2%	2%	2%	1%	2%	0%	0%	0%	0%	0%	0%	0%	0%
	3	nº Problemas	9	2	10	18	21	23	28	22	29	28	16	17	13	11	9	4	4	2	0	2	1	0	1	0	0	0	0
	% Problemas	3%	1%	4%	7%	8%	9%	10%	8%	11%	10%	6%	6%	5%	4%	3%	1%	1%	1%	0%	1%	0%	0%	0%	0%	0%	0%	0%	0%
	4	nº Problemas	8	5	10	25	22	22	29	22	22	25	16	14	18	10	7	3	4	4	0	2	1	0	1	0	0	0	0
	% Problemas	3%	2%	4%	9%	8%	8%	11%	8%	8%	9%	6%	5%	7%	4%	3%	1%	1%	1%	0%	1%	0%	0%	0%	0%	0%	0%	0%	0%

		Comparación Optimos y Est 14																											
		Rango	0%	1%	2%	3%	4%	5%	6%	7%	8%	9%	10%	11%	12%	13%	14%	15%	16%	17%	18%	19%	20%	21%	22%	23%	24%	25%	
Total	nº Problemas	12	16	25	30	55	58	78	104	79	96	81	98	67	84	47	37	33	22	17	11	5	7	5	2	1	0		
	% Problemas	1%	1%	2%	3%	5%	5%	7%	10%	7%	9%	8%	9%	6%	8%	4%	3%	3%	2%	2%	1%	0%	1%	0%	0%	0%	0%		
Solape	1	nº Problemas	1	6	12	8	25	22	28	39	35	41	26	42	13	29	7	6	9	4	4	1	0	1	0	0	0	0	
		% Problemas	0%	2%	3%	2%	7%	6%	8%	11%	10%	11%	7%	12%	4%	8%	2%	2%	3%	1%	1%	0%	0%	0%	0%	0%	0%	0%	
	2	nº Problemas	6	2	8	8	12	18	25	37	21	38	24	22	25	27	23	20	8	10	7	7	1	2	3	0	0	0	
		% Problemas	2%	1%	2%	2%	3%	5%	7%	10%	6%	11%	7%	6%	7%	8%	6%	6%	2%	3%	2%	2%	0%	1%	1%	0%	0%	0%	
	4	nº Problemas	5	8	5	14	18	18	25	28	23	17	31	34	29	28	17	11	16	8	6	3	4	4	2	2	1	0	
		% Problemas	1%	2%	1%	4%	5%	5%	7%	8%	6%	5%	9%	9%	8%	8%	5%	3%	4%	2%	2%	1%	1%	1%	1%	1%	0%	0%	
Numero Trab	50	nº Problemas	11	8	14	16	17	20	27	29	29	39	27	24	18	13	10	19	13	4	5	1	1	6	3	2	0	0	
		% Problemas	3%	2%	4%	4%	5%	6%	8%	8%	8%	11%	8%	7%	5%	4%	3%	5%	4%	1%	1%	0%	0%	2%	1%	1%	0%	0%	
	100	nº Problemas	1	8	11	14	38	38	51	75	50	57	54	74	49	71	37	18	20	18	12	10	4	1	2	0	1	0	
		% Problemas	0%	1%	2%	2%	5%	5%	7%	10%	7%	8%	8%	10%	7%	10%	5%	3%	3%	3%	2%	1%	1%	0%	0%	0%	0%	0%	
Numero Mac	1	nº Problemas	5	9	15	12	27	27	36	55	24	30	29	27	15	22	8	8	3	6	2	0	0	0	0	0	0	0	
		% Problemas	1%	3%	4%	3%	8%	8%	10%	15%	7%	8%	8%	8%	4%	6%	2%	2%	1%	2%	1%	0%	0%	0%	0%	0%	0%	0%	
	2	nº Problemas	5	4	7	11	12	19	31	34	35	36	28	31	26	26	13	11	16	5	4	2	2	0	1	0	0	0	
		% Problemas	1%	1%	2%	3%	3%	5%	9%	9%	10%	10%	8%	9%	7%	7%	4%	3%	4%	1%	1%	1%	1%	0%	0%	0%	0%	0%	
	3	nº Problemas	2	3	3	7	16	12	11	15	20	30	24	40	26	36	26	18	14	11	11	9	3	7	4	2	1	0	
		% Problemas	1%	1%	1%	2%	4%	3%	4%	6%	8%	7%	11%	7%	10%	7%	5%	4%	3%	3%	3%	1%	2%	1%	1%	0%	0%	0%	
Numero Mac	4	nº Problemas	11	11	16	21	27	38	49	69	50	71	55	61	42	55	31	27	23	19	10	10	1	7	4	2	1	0	
		% Problemas	2%	2%	2%	3%	4%	5%	7%	10%	7%	10%	8%	8%	6%	8%	4%	4%	3%	3%	1%	1%	0%	1%	1%	0%	0%	0%	
	8	nº Problemas	1	5	9	9	28	20	29	35	29	25	26	37	25	29	16	10	10	3	7	1	4	0	1	0	0	0	
		% Problemas	0%	1%	3%	3%	8%	6%	8%	10%	8%	7%	7%	10%	7%	8%	4%	3%	3%	1%	2%	0%	1%	0%	0%	0%	0%	0%	
Distr Maquinas	1	nº Problemas	2	3	4	4	6	12	9	13	21	30	23	24	26	35	16	10	8	10	6	1	1	2	1	0	1	0	
		% Problemas	1%	1%	1%	1%	2%	4%	3%	5%	8%	11%	9%	9%	10%	13%	6%	4%	3%	4%	2%	0%	0%	1%	0%	0%	0%	0%	
	2	nº Problemas	1	2	4	8	8	9	14	28	24	30	22	27	14	24	16	9	15	4	6	1	1	1	1	1	0	0	0
		% Problemas	0%	1%	1%	3%	3%	3%	5%	10%	9%	11%	8%	10%	5%	9%	6%	3%	6%	1%	2%	0%	0%	0%	0%	0%	0%	0%	0%
	3	nº Problemas	4	4	6	7	18	18	28	29	17	19	22	23	17	13	8	8	6	4	3	5	2	2	2	1	0	0	0
		% Problemas	1%	1%	2%	3%	7%	7%	10%	11%	6%	7%	8%	9%	6%	5%	3%	3%	2%	1%	1%	2%	1%	1%	1%	0%	0%	0%	0%
4	nº Problemas	5	7	11	11	23	19	27	34	17	17	14	24	10	12	7	10	4	4	2	4	1	2	1	1	0	0	0	
	% Problemas	2%	3%	4%	4%	9%	7%	10%	13%	6%	6%	5%	9%	4%	4%	3%	4%	1%	1%	1%	1%	0%	1%	0%	0%	0%	0%	0%	

		Comparación Optimos y Max Est 1																													
		Rango	0%	1%	2%	3%	4%	5%	6%	7%	8%	9%	10%	11%	12%	13%	14%	15%	16%	17%	18%	19%	20%	21%	22%	23%	24%	25%			
Total	nº Problemas	83	54	89	122	144	124	119	130	102	53	37	15	5	2	1	0	0	0	0	0	0	0	0	0	0	0	0			
	% Problemas	8%	5%	8%	11%	13%	12%	11%	12%	9%	5%	3%	1%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%			
Numero Trab Solape	1	nº Problemas	16	17	24	39	64	50	55	42	27	11	8	4	2	1	0	0	0	0	0	0	0	0	0	0	0	0			
	% Problemas	4%	5%	7%	11%	18%	14%	15%	12%	8%	3%	2%	1%	1%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%			
	2	nº Problemas	18	15	36	35	40	42	35	48	42	25	15	7	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
	% Problemas	5%	4%	10%	10%	11%	12%	10%	13%	12%	7%	4%	2%	1%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%		
	4	nº Problemas	49	22	29	48	40	32	29	40	33	17	14	4	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0		
	% Problemas	14%	6%	8%	13%	11%	9%	8%	11%	9%	5%	4%	1%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%		
Numero Trab	50	nº Problemas	64	32	47	57	48	42	27	15	14	4	7	2	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0		
	% Problemas	18%	9%	13%	16%	13%	12%	8%	4%	4%	1%	2%	1%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%		
100	nº Problemas	19	22	42	65	96	82	92	115	88	49	30	13	5	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0		
% Problemas	3%	3%	6%	9%	13%	11%	13%	16%	12%	7%	4%	2%	1%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%		
Numero Maq Tipol	1	nº Problemas	20	17	33	39	54	53	42	41	31	9	13	5	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0		
	% Problemas	6%	5%	9%	11%	15%	15%	12%	11%	9%	3%	4%	1%	1%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%		
	2	nº Problemas	43	23	31	43	54	38	40	29	26	17	10	5	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	% Problemas	12%	6%	9%	12%	15%	11%	11%	8%	7%	5%	3%	1%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	
	3	nº Problemas	20	14	25	40	36	33	37	60	45	27	14	5	2	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	
	% Problemas	6%	4%	7%	11%	10%	9%	10%	17%	13%	8%	4%	1%	1%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	
Numero Maq	4	nº Problemas	72	43	66	96	93	84	76	74	57	27	17	10	3	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	
	% Problemas	10%	6%	9%	13%	13%	12%	11%	10%	8%	4%	2%	1%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	
8	nº Problemas	11	11	23	26	51	40	43	56	45	26	20	5	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
% Problemas	3%	3%	6%	7%	14%	11%	12%	16%	13%	7%	6%	1%	1%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	
Distr Maquinas	1	nº Problemas	14	6	17	22	26	41	28	35	39	18	12	7	4	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	% Problemas	5%	2%	6%	8%	10%	15%	10%	13%	14%	7%	4%	3%	1%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	
	2	nº Problemas	14	11	15	26	40	30	30	39	30	18	11	4	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	% Problemas	5%	4%	6%	10%	15%	11%	11%	14%	11%	7%	4%	1%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
	3	nº Problemas	23	14	30	35	39	27	34	30	19	10	7	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	% Problemas	9%	5%	11%	13%	14%	10%	13%	11%	7%	4%	3%	1%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
4	nº Problemas	32	23	27	39	39	26	27	26	14	7	7	2	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
% Problemas	12%	9%	10%	14%	14%	10%	10%	10%	5%	3%	3%	1%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	

		Comparación Optimos y Est 2																										
		Rango	0%	1%	2%	3%	4%	5%	6%	7%	8%	9%	10%	11%	12%	13%	14%	15%	16%	17%	18%	19%	20%	21%	22%	23%	24%	25%
Total	nº Problemas	117	105	177	213	164	117	71	58	29	12	6	6	2	1	2	0	0	0	0	0	0	0	0	0	0	0	0
	% Problemas	11%	10%	16%	20%	15%	11%	7%	5%	3%	1%	1%	1%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
Solape	1	nº Problemas	38	38	71	68	49	34	22	21	9	2	3	1	2	0	2	0	0	0	0	0	0	0	0	0	0	0
		% Problemas	11%	11%	20%	19%	14%	9%	6%	6%	3%	1%	1%	0%	1%	0%	1%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
	2	nº Problemas	36	34	47	77	63	41	20	19	11	5	2	4	0	1	0	0	0	0	0	0	0	0	0	0	0	0
		% Problemas	10%	9%	13%	21%	18%	11%	6%	5%	3%	1%	1%	1%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
	4	nº Problemas	43	33	59	68	52	42	29	18	9	5	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		% Problemas	12%	9%	16%	19%	14%	12%	8%	5%	3%	1%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
Numero Trab	50	nº Problemas	89	35	42	58	36	33	14	24	12	6	3	4	1	1	2	0	0	0	0	0	0	0	0	0	0	0
		% Problemas	25%	10%	12%	16%	10%	9%	4%	7%	3%	2%	1%	1%	0%	0%	1%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
	100	nº Problemas	28	70	135	155	128	84	57	34	17	6	3	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0
		% Problemas	4%	10%	19%	22%	18%	12%	8%	5%	2%	1%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
Numero Mac	1	nº Problemas	34	44	94	77	42	34	18	14	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		% Problemas	9%	12%	26%	21%	12%	9%	5%	4%	1%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
	2	nº Problemas	60	40	58	75	52	21	18	17	6	3	2	4	1	1	2	0	0	0	0	0	0	0	0	0	0	0
		% Problemas	17%	11%	16%	21%	14%	6%	5%	5%	2%	1%	1%	1%	0%	0%	1%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
	3	nº Problemas	23	21	25	61	70	62	35	27	21	8	4	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0
		% Problemas	6%	6%	7%	17%	20%	17%	10%	8%	6%	2%	1%	1%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
Numero Mac	4	nº Problemas	102	66	108	131	98	79	51	42	21	9	4	5	1	1	2	0	0	0	0	0	0	0	0	0	0	0
		% Problemas	14%	9%	15%	18%	14%	11%	7%	6%	3%	1%	1%	1%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
	8	nº Problemas	15	39	69	82	66	38	20	16	8	3	2	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
		% Problemas	4%	11%	19%	23%	18%	11%	6%	4%	2%	1%	1%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
Distr Maquinas	1	nº Problemas	20	21	41	45	53	29	20	18	8	6	3	4	1	1	0	0	0	0	0	0	0	0	0	0	0	0
		% Problemas	7%	8%	15%	17%	20%	11%	7%	7%	3%	2%	1%	1%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
	2	nº Problemas	24	21	40	55	42	33	25	13	11	2	1	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0
		% Problemas	9%	8%	15%	20%	16%	12%	9%	5%	4%	1%	0%	1%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
	3	nº Problemas	31	23	45	63	40	31	14	14	5	2	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
		% Problemas	12%	9%	17%	23%	15%	12%	5%	5%	2%	1%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
	4	nº Problemas	42	40	51	50	29	24	12	13	5	2	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
		% Problemas	16%	15%	19%	19%	11%	9%	4%	5%	2%	1%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%

		Comparación Optimos y Est 22																									
Rango		0%	1%	2%	3%	4%	5%	6%	7%	8%	9%	10%	11%	12%	13%	14%	15%	16%	17%	18%	19%	20%	21%	22%	23%	24%	25%
Total	nº Problemas	61	61	81	133	193	149	106	87	57	43	27	35	12	10	6	6	5	1	4	1	2	0	0	0	0	0
	% Problemas	6%	6%	8%	12%	18%	14%	10%	8%	5%	4%	3%	3%	1%	1%	1%	1%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
Numero Trab Solape	1	nº Problemas	9	14	22	40	68	50	37	25	18	14	17	16	8	6	5	3	2	1	2	1	2	0	0	0	0
	% Problemas	3%	4%	6%	11%	19%	14%	10%	7%	5%	4%	5%	4%	2%	2%	1%	1%	1%	0%	1%	0%	1%	0%	0%	0%	0%	0%
	2	nº Problemas	16	30	30	42	55	54	28	37	20	18	5	12	3	2	1	2	3	0	2	0	0	0	0	0	0
	% Problemas	4%	8%	8%	12%	15%	15%	8%	10%	6%	5%	1%	3%	1%	1%	0%	1%	1%	0%	1%	0%	0%	0%	0%	0%	0%	0%
	4	nº Problemas	36	17	29	51	70	45	41	25	19	11	5	7	1	2	0	1	0	0	0	0	0	0	0	0	0
	% Problemas	10%	5%	8%	14%	19%	13%	11%	7%	5%	3%	1%	2%	0%	1%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
Numero Trab	50	nº Problemas	49	33	34	46	43	36	29	26	15	14	9	11	3	3	1	1	2	1	2	0	2	0	0	0	0
	% Problemas	14%	9%	9%	13%	12%	10%	8%	7%	4%	4%	3%	3%	1%	1%	0%	0%	1%	0%	1%	0%	1%	0%	0%	0%	0%	0%
100	nº Problemas	12	28	47	87	150	113	77	61	42	29	18	24	9	7	5	5	3	0	2	1	0	0	0	0	0	
% Problemas	2%	4%	7%	12%	21%	16%	11%	8%	6%	4%	3%	3%	1%	1%	1%	1%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	
Numero Maq Tipol	1	nº Problemas	21	31	41	53	89	49	37	18	10	7	1	2	1	0	0	0	0	0	0	0	0	0	0	0	0
	% Problemas	6%	9%	11%	15%	25%	14%	10%	5%	3%	2%	0%	1%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
	2	nº Problemas	17	9	15	19	34	38	34	42	26	28	22	31	10	10	6	6	5	1	4	1	2	0	0	0	0
	% Problemas	5%	3%	4%	5%	9%	11%	9%	12%	7%	8%	6%	9%	3%	3%	2%	2%	1%	0%	1%	0%	1%	0%	0%	0%	0%	0%
	3	nº Problemas	23	21	25	61	70	62	35	27	21	8	4	2	1	0	0	0	0	0	0	0	0	0	0	0	0
	% Problemas	6%	6%	7%	17%	20%	17%	10%	8%	6%	2%	1%	1%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
Numero Maq	4	nº Problemas	60	51	56	93	102	100	62	63	41	29	15	25	6	7	2	1	2	1	2	0	2	0	0	0	0
	% Problemas	8%	7%	8%	13%	14%	14%	9%	9%	6%	4%	2%	3%	1%	1%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
8	nº Problemas	1	10	25	40	91	49	44	24	16	14	12	10	6	3	4	5	3	0	2	1	0	0	0	0	0	
% Problemas	0%	3%	7%	11%	25%	14%	12%	7%	4%	4%	3%	3%	2%	1%	1%	1%	1%	0%	1%	0%	0%	0%	0%	0%	0%	0%	
Distr Maquinas	1	nº Problemas	19	11	24	37	46	38	22	25	20	11	9	4	2	0	1	1	0	0	0	0	0	0	0	0	0
	% Problemas	7%	4%	9%	14%	17%	14%	8%	9%	7%	4%	3%	1%	1%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
	2	nº Problemas	17	12	18	37	54	37	31	21	17	10	6	5	2	1	1	0	0	1	0	0	0	0	0	0	0
	% Problemas	6%	4%	7%	14%	20%	14%	11%	8%	6%	4%	2%	2%	1%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
	3	nº Problemas	12	16	18	27	53	37	26	23	10	9	6	14	2	6	3	3	2	0	1	1	1	0	0	0	0
	% Problemas	4%	6%	7%	10%	20%	14%	10%	9%	4%	3%	2%	5%	1%	2%	1%	1%	1%	0%	0%	0%	0%	0%	0%	0%	0%	0%
	4	nº Problemas	13	22	21	32	40	37	27	18	10	13	6	12	6	3	1	2	3	0	3	0	1	0	0	0	0
	% Problemas	5%	8%	8%	12%	15%	14%	10%	7%	4%	5%	2%	4%	2%	1%	0%	1%	1%	0%	1%	0%	0%	0%	0%	0%	0%	0%

		Comparación Optimos y Max Est 2																											
		Rango	0%	1%	2%	3%	4%	5%	6%	7%	8%	9%	10%	11%	12%	13%	14%	15%	16%	17%	18%	19%	20%	21%	22%	23%	24%	25%	
Total	nº Problemas	255	206	237	212	117	33	12	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	% Problemas	24%	19%	22%	20%	11%	3%	1%	1%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	
Solape	1	nº Problemas	68	65	92	78	41	9	5	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
		% Problemas	19%	18%	26%	22%	11%	3%	1%	1%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
	2	nº Problemas	71	71	73	81	38	17	3	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		% Problemas	20%	20%	20%	23%	11%	5%	1%	2%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
	4	nº Problemas	116	70	72	53	38	7	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		% Problemas	32%	19%	20%	15%	11%	2%	1%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
Numero Trab	50	nº Problemas	182	46	42	46	21	14	4	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
		% Problemas	51%	13%	12%	13%	6%	4%	1%	1%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
	100	nº Problemas	73	160	195	166	96	19	8	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		% Problemas	10%	22%	27%	23%	13%	3%	1%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
Numero Mac	1	nº Problemas	78	67	96	67	31	11	5	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		% Problemas	22%	19%	27%	19%	9%	3%	1%	1%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
	2	nº Problemas	103	68	58	70	46	8	5	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		% Problemas	29%	19%	16%	19%	13%	2%	1%	1%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
	3	nº Problemas	74	71	83	75	40	14	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		% Problemas	21%	20%	23%	21%	11%	4%	1%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
4	nº Problemas	229	141	135	114	64	21	9	7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	% Problemas	32%	20%	19%	16%	9%	3%	1%	1%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	
8	nº Problemas	26	65	102	98	53	12	3	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	% Problemas	7%	18%	28%	27%	15%	3%	1%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	
Distr Maquinas	1	nº Problemas	50	48	69	55	35	10	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		% Problemas	19%	18%	26%	20%	13%	4%	1%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
	2	nº Problemas	57	41	62	63	28	15	3	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		% Problemas	21%	15%	23%	23%	10%	6%	1%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
	3	nº Problemas	65	49	54	54	35	5	4	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		% Problemas	24%	18%	20%	20%	13%	2%	1%	1%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
	4	nº Problemas	83	68	52	40	19	3	2	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		% Problemas	31%	25%	19%	15%	7%	1%	1%	1%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%

