

CAPÍTULO 6

RESULTADOS COMPUTACIONALES

En este capítulo se muestra el estudio realizado para llevar a cabo el análisis de sensibilidad del algoritmo variando los parámetros, y la comparación entre los resultados del algoritmo *VIRUS* y los obtenidos con *Lingo*.

Se comienza hablando de la generación de la batería de problemas con la cual se ha realizado el análisis. A continuación se expone el estudio sobre los parámetros del programa que han de ajustarse, y para terminar, se resuelven todos los problemas con la rutina *VIRUS* (con los parámetros ya ajustados), y con el optimizador *Lingo*, para así comparar los resultados de ambos métodos, si bien, sólo fue admisible la resolución exacta de problemas con hasta 100 trabajos por el optimizador.

6.1. BATERÍA DE PROBLEMAS

Para la obtención de los resultados se ha generado una batería de problemas con una amplia gama de casos posibles. Para esto se ha variado la amplitud máxima de la ventana de comienzos (5, 10 y 20), el grado de solapamiento de los trabajos (0.8, 1.4 y 2), el número de éstos (25, 50, 100, 200 y 400), y el número de máquinas (4, 8 y 16). El horizonte temporal en todos los casos es de 100 unidades de tiempo.

El grado de solapamiento es un parámetro que le sirve al programa generador para saber la duración máxima que pueden tener los trabajos. Esta duración sigue la siguiente fórmula:

$$t_j^{m\acute{a}x} = \frac{25 \cdot HT \cdot S}{n} \quad (6.1)$$

donde *HT* es el horizonte temporal, *S* es el grado de solapamiento y *n* el número de trabajos.

Para cada una de las combinaciones posibles de estos valores se generan diez problemas distintos (0 - 9).

La generación de los trabajos se realiza con una sencilla rutina de *Visual Basic* proporcionada por el tutor del proyecto. En la tabla 6.1 se exponen las posibles combinaciones generadas.

De todas estas combinaciones, sólo se van a emplear para el estudio de sensibilidad los problemas con número 0 y valor de la ventana de comienzo igual a 20; por lo que tendremos 3 problemas distintos con 25 trabajos, 6 con 50 trabajos, y 9 con 100, con 200 y con 400 trabajos. Esto se hace sólo para el siguiente apartado, cuyo objetivo es ajustar los parámetros del algoritmo para obtener la resolución adecuada.

Nº de trabajos	Ventana de comienzos máxima	Grado de solapamiento	Nº máquinas	Nº problema
25	5, 10, 20	0.8, 1.4, 2	4	0 - 9
50	5, 10, 20	0.8, 1.4, 2	4, 8	0 - 9
100	5, 10, 20	0.8, 1.4, 2	4, 8, 16	0 - 9
200	5, 10, 20	0.8, 1.4, 2	4, 8, 16	0 - 9
400	5, 10, 20	0.8, 1.4, 2	4, 8, 16	0 - 9

Tabla 6.1. Tipos de archivos de entrada

6.2. AJUSTE DE LOS PARÁMETROS

Para realizar el estudio con el método *VIRUS* se van a ir variando diferentes parámetros, y así obtener los valores óptimos de estos. Con estos valores se hará la posterior comparativa con los resultados obtenidos con *Lingo*.

En este estudio permanecerán fijos ciertos parámetros, cuyos valores se han obtenido en anteriores investigaciones ($LNR = 15$, $LIT = 10$, Probabilidad de inserción = 80%).

6.2.1. VARIANDO EL NÚMERO DE ITERACIONES

El primer análisis se realiza aumentando el valor del número de iteraciones del proceso. Se coge un número de iteraciones igual a 10^r , variando r de 1 hasta 5. Los demás valores se mantienen fijos.

A continuación algunos de los resultados obtenidos, donde la columna *Valor* muestra el valor de la solución encontrada, *Inicial* muestra el valor de la mejor solución inicial obtenida, *Iteraciones* muestra el número de éstas que se han procesado, *Tiempo* indica la duración del cálculo, *Población* el tamaño de ésta, *Pliti* y *Pliso* muestran respectivamente las probabilidades de que un elemento salga *lítico* o *lisogénico*, e *ItMejor* indica en qué iteración se ha encontrado la que se muestra como mejor solución.

Nombre	Valor	Inicial	Iteraciones	Tiempo	Población	Pliti	Pliso	ItMejor
File_3_2_25_4_3_2_0.txt	3921	3921	10	0:00:00	50	70	30	0
File_3_2_25_4_3_2_0.txt	4347	3608	100	0:00:00	50	70	30	70
File_3_2_25_4_3_2_0.txt	4678	4065	1000	0:00:01	50	70	30	342
File_3_2_25_4_3_2_0.txt	4678	3688	10000	0:00:24	50	70	30	1579
File_3_2_25_4_3_2_0.txt	4678	4024	100000	0:05:12	50	70	30	1215

Tabla 6.2. Resultados con 25 trabajos, 4 máquinas y grado 0.8 de solapamiento

Nombre	Valor	Inicial	Iteraciones	Tiempo	Población	Pliti	Pliso	ItMejor
File_3_1,4_50_8_3_3_0.txt	10751	10751	10	0:00:00	50	70	30	0
File_3_1,4_50_8_3_3_0.txt	12368	11871	100	0:00:00	50	70	30	50
File_3_1,4_50_8_3_3_0.txt	15769	11925	1000	0:00:03	50	70	30	582
File_3_1,4_50_8_3_3_0.txt	17079	11836	10000	0:01:07	50	70	30	5718
File_3_1,4_50_8_3_3_0.txt	17403	13464	100000	0:15:34	50	70	30	85984

Tabla 6.3. Resultados con 50 trabajos, 8 máquinas y grado 1.4 de solapamiento

Nombre	Valor	Inicial	Iteraciones	Tiempo	Población	Pliti	Pliso	ItMejor
File_3_0,8_100_16_3_4_0.txt	30891	29919	10	0:00:00	50	70	30	5
File_3_0,8_100_16_3_4_0.txt	34753	31262	100	0:00:00	50	70	30	92
File_3_0,8_100_16_3_4_0.txt	40404	29876	1000	0:00:01	50	70	30	788
File_3_0,8_100_16_3_4_0.txt	48988	31033	10000	0:00:11	50	70	30	5738
File_3_0,8_100_16_3_4_0.txt	48988	31165	100000	0:01:58	50	70	30	3104

Tabla 6.4. Resultados con 100 trabajos, 16 máquinas y grado 0.8 de solapamiento

Vemos como al aumentar el número de iteraciones, el valor de la solución aumenta y que, como era de esperar, el tiempo de cálculo aumenta también.

Nombre	Valor	Inicial	Iteraciones	Tiempo	Población	Pliti	Pliso	ItMejor
File_3_0,8_200_8_3_3_0.txt	51652	51652	10	0:00:00	50	70	30	0
File_3_0,8_200_8_3_3_0.txt	53774	52803	100	0:00:00	50	70	30	45
File_3_0,8_200_8_3_3_0.txt	67094	52387	1000	0:00:03	50	70	30	930
File_3_0,8_200_8_3_3_0.txt	85901	55563	10000	0:00:36	50	70	30	6084
File_3_0,8_200_8_3_3_0.txt	87636	53084	100000	0:23:52	50	70	30	79864

Tabla 6.5. Resultados con 200 trabajos, 8 máquinas y grado 0.8 de solapamiento

Es fácil darse cuenta de que al aumentar el número de trabajos, también aumenta el tiempo de resolución, pues los bucles se hacen más largos, y además hacen falta más iteraciones para llegar a un valor lo suficientemente bueno. Vemos también que la diferencia en tiempo entre 10.000 y 100.000 iteraciones es demasiado alta, mientras que la diferencia en valor no lo es tanto, por ello se escogen 10.000 iteraciones para continuar con el estudio de los demás parámetros, pues en la mayoría de los casos, es con este número de iteraciones cuándo se consigue la mejor solución de compromiso.

Nombre	Valor	Inicial	Iteraciones	Tiempo	Población	Pliti	Pliso	ItMejor
File_3_2_400_16_3_4_0.txt	100759	100759	10	0:00:00	50	70	30	0
File_3_2_400_16_3_4_0.txt	101729	99735	100	0:00:01	50	70	30	97
File_3_2_400_16_3_4_0.txt	114529	99037	1000	0:00:13	50	70	30	991
File_3_2_400_16_3_4_0.txt	155934	101974	10000	0:02:09	50	70	30	9556
File_3_2_400_16_3_4_0.txt	161892	102147	100000	1:31:44	50	70	30	89023

Tabla 6.6. Resultados con 400 trabajos, 16 máquinas y grado 2 de solapamiento

6.2.2. VARIANDO LAS PROBABILIDADES LÍTICA Y LISOGÉNICA

En este segundo análisis se va a partir de una $p_{Liti} = 0.1$ y se va a ir aumentando en saltos de 0.2 hasta que valga 0.9. Por supuesto, como $p_{Liso} = 1 - p_{Liti}$, a la vez que p_{Liti} aumenta p_{Liso} disminuye.

Estos parámetros son muy importantes, pues como se va a ver en los resultados, el hecho de que la probabilidad de un elemento de ser *lítico* sea mayor o menor, influirá en cuántos elementos de la población hacen la replicación *lítica*, lo que es lo mismo que decir cuántas veces se entra en esta función.

En general, cuantas más veces en el proceso se entre en la función *lítica* mejor para el valor de la solución, pues el estudio de vecindades es más complejo, lo que hace que se tarde más en el cálculo. Veamos algunos ejemplos.

Nombre	Valor	Inicial	Iteraciones	Tiempo	Población	Pliti	Pliso	ItMejor
File_3_1,4_25_4_3_2_0.txt	8070	5879	10000	0:00:03	50	10	90	3804
File_3_1,4_25_4_3_2_0.txt	8095	5775	10000	0:00:08	50	30	70	6876
File_3_1,4_25_4_3_2_0.txt	8364	5769	10000	0:00:17	50	50	50	3590
File_3_1,4_25_4_3_2_0.txt	8359	6733	10000	0:00:26	50	70	30	4065
File_3_1,4_25_4_3_2_0.txt	8125	5872	10000	0:00:41	50	90	10	9716

Tabla 6.7. Resultados con 25 trabajos, 4 máquinas y grado 1.4 de solapamiento

Nombre	Valor	Inicial	Iteraciones	Tiempo	Población	Pliti	Pliso	ItMejor
File_3_0,8_50_8_3_3_0.txt	17676	12690	10000	0:00:05	50	10	90	9580
File_3_0,8_50_8_3_3_0.txt	19235	13510	10000	0:00:18	50	30	70	9950
File_3_0,8_50_8_3_3_0.txt	19368	12683	10000	0:00:33	50	50	50	9975
File_3_0,8_50_8_3_3_0.txt	19056	13537	10000	0:00:57	50	70	30	8973
File_3_0,8_50_8_3_3_0.txt	18966	13433	10000	0:01:15	50	90	10	3632

Tabla 6.8. Resultados con 50 trabajos, 8 máquinas y grado 0.8 de solapamiento

Nombre	Valor	Inicial	Iteraciones	Tiempo	Población	Pliti	Pliso	ItMejor
File_3_2_100_16_3_4_0.txt	30409	21402	10000	0:00:21	50	10	90	7573
File_3_2_100_16_3_4_0.txt	31986	21220	10000	0:01:12	50	30	70	9990
File_3_2_100_16_3_4_0.txt	32427	20863	10000	0:02:18	50	50	50	9772
File_3_2_100_16_3_4_0.txt	32242	20126	10000	0:03:47	50	70	30	6082
File_3_2_100_16_3_4_0.txt	32700	21349	10000	0:04:55	50	90	10	9647

Tabla 6.9. Resultados con 100 trabajos, 16 máquinas y grado 2 de solapamiento

Puede apreciarse que para problemas con tan pocos trabajos, con una $p_{Liti} = 0.5$ se tiene la mejor solución de compromiso, pero como se verá a continuación, para problemas mayores p_{Liti} es mejor que valga 0.7.

Nombre	Valor	Inicial	Iteraciones	Tiempo	Población	Pliti	Pliso	ItMejor
File_3_1,4_200_16_3_4_0.txt	81456	57041	10000	0:00:10	50	10	90	9555
File_3_1,4_200_16_3_4_0.txt	87548	58941	10000	0:00:21	50	30	70	9886
File_3_1,4_200_16_3_4_0.txt	90579	58055	10000	0:00:34	50	50	50	9471
File_3_1,4_200_16_3_4_0.txt	92428	60156	10000	0:00:42	50	70	30	8828
File_3_1,4_200_16_3_4_0.txt	91144	58259	10000	0:00:57	50	90	10	4652

Tabla 6.10. Resultados con 200 trabajos, 16 máquinas y grado 1.4 de solapamiento

Nombre	Valor	Inicial	Iteraciones	Tiempo	Población	Pliti	Pliso	ItMejor
File_3_0,8_400_8_3_3_0.txt	137612	102942	10000	0:00:14	50	10	90	9932
File_3_0,8_400_8_3_3_0.txt	147117	102944	10000	0:00:37	50	30	70	9671
File_3_0,8_400_8_3_3_0.txt	159631	104761	10000	0:00:54	50	50	50	9967
File_3_0,8_400_8_3_3_0.txt	167470	101722	10000	0:01:03	50	70	30	9448
File_3_0,8_400_8_3_3_0.txt	167367	111888	10000	0:01:12	50	90	10	9460

Tabla 6.11. Resultados con 400 trabajos, 8 máquinas y grado 0.8 de solapamiento

Por lo tanto se debe concluir que no hay una probabilidad p_{Liti} que sea adecuada en todos los casos, se cogerá en cada caso la que proceda, es decir, cuando tengamos pocos trabajos cogeremos p_{Liti} igual a 0.5 y cuando tengamos muchos p_{Liti} valdrá 0.7. De todos modos, para hacer el siguiente análisis se cogerá $p_{Liti} = 0.7$ en todos los casos, para que este parámetro no influya en la comparativa.

6.2.3. VARIANDO EL TAMAÑO DE LA POBLACIÓN

Para este último análisis se prueba con tamaños de población de 50, 75 y 100 elementos en cada caso, viendo así la influencia de este parámetro en la solución. Para ilustrarlo, se muestran algunos de los resultados obtenidos.

Nombre	Valor	Inicial	Iteraciones	Tiempo	Población	Pliti	Pliso	ItMejor
File_3_2_25_4_3_2_0.txt	4678	3433	10000	0:00:25	50	70	30	1793
File_3_2_25_4_3_2_0.txt	4678	3779	10000	0:00:22	75	70	30	1322
File_3_2_25_4_3_2_0.txt	4678	4065	10000	0:00:21	100	70	30	2038

Tabla 6.12. Resultados con 25 trabajos, 4 máquinas y grado 2 de solapamiento

Nombre	Valor	Inicial	Iteraciones	Tiempo	Población	Pliti	Pliso	ItMejor
File_3_2_50_8_3_3_0.txt	12403	8326	10000	0:01:48	50	70	30	3659
File_3_2_50_8_3_3_0.txt	12464	8285	10000	0:01:33	75	70	30	4817
File_3_2_50_8_3_3_0.txt	12452	9131	10000	0:01:10	100	70	30	2280

Tabla 6.13. Resultados con 50 trabajos, 8 máquinas y grado 2 de solapamiento

Nombre	Valor	Inicial	Iteraciones	Tiempo	Población	Pliti	Pliso	ItMejor
File_3_1,4_100_8_3_3_0.txt	29174	18673	10000	0:01:44	50	70	30	8168
File_3_1,4_100_8_3_3_0.txt	29269	19471	10000	0:01:25	75	70	30	9980
File_3_1,4_100_8_3_3_0.txt	28547	19961	10000	0:01:02	100	70	30	8492

Tabla 6.14. Resultados con 100 trabajos, 8 máquinas y grado 1.4 de solapamiento

Parece claro que cuantos más trabajos tengamos, menos elementos en la población se necesitan, esto es porque al haber pocos elementos, los que hay

evolucionan mucho. Cuando el tamaño de la población aumenta, la probabilidad de que en el algoritmo se escoja un elemento concreto de ésta disminuye, lo que hace que en general se tarde menos, pero las soluciones son menos complejas. Sin embargo, cuando se tiene un problema con pocos trabajos no es necesario que la solución alcance un grado de complejidad alto, pues hay menos combinaciones posibles. Es decir, al igual que en el análisis anterior, dependiendo del número de trabajos que tengamos, escogeremos un tamaño de población u otro.

Nombre	Valor	Inicial	Iteraciones	Tiempo	Población	Pliti	Pliso	ItMejor
File_3_2_200_4_3_2_0.txt	44681	24545	10000	0:01:12	50	70	30	6733
File_3_2_200_4_3_2_0.txt	45802	26069	10000	0:00:52	75	70	30	8846
File_3_2_200_4_3_2_0.txt	43939	25510	10000	0:00:45	100	70	30	4559

Tabla 6.15. Resultados con 200 trabajos, 4 máquinas y grado 2 de solapamiento

Nombre	Valor	Inicial	Iteraciones	Tiempo	Población	Pliti	Pliso	ItMejor
File_3_1,4_400_16_3_4_0.txt	169448	106068	10000	0:01:17	50	70	30	9747
File_3_1,4_400_16_3_4_0.txt	165831	105957	10000	0:01:20	75	70	30	9963
File_3_1,4_400_16_3_4_0.txt	158073	109374	10000	0:01:07	100	70	30	9800

Tabla 6.16. Resultados con 400 trabajos, 8 máquinas y grado 1.4 de solapamiento

Para concluir podemos decir que los valores adecuados de los parámetros que se han ido variando para cada grupo de problemas son los siguientes:

Nº de trabajos	Número de iteraciones	p_{Liti}	p_{Liso}	Tamaño de la población
25	10000	0.5	0.5	100
50	10000	0.5	0.5	75
100	25000	0.5	0.5	75
200	25000	0.7	0.3	75
400	25000	0.7	0.3	50

Tabla 6.17. Valores finales de los parámetros

Se ha decidido incrementar el número de iteraciones para 100, 200 y 400 trabajos, sin llegar hasta el siguiente orden de magnitud, para mejorar la solución, ya que en éstos se hace más lenta la obtención del óptimo, debido a la gran cantidad de combinaciones posibles.

6.3. RESOLUCIÓN COMPLETA Y COMPARATIVA

En este apartado se resuelve la batería entera de problemas, generada como se explica en el apartado 6.1, con los valores de los parámetros que se han hallado en el apartado anterior.

Para realizar la comparación se resolverán los problemas de hasta 100 trabajos con el optimizador *Lingo* para así hallar el error del método. Problemas superiores a estos en número de trabajos fueron intratables con el uso del optimizador. En el anexo I se recogen los tiempos, en media, que ha necesitado el optimizador para resolver los problemas expuestos.

En las gráficas que se mostrarán se pondrá la media del error de las diez instancias distintas para cada caso. En los gráficos la leyenda indica la ventana de comienzo máxima.

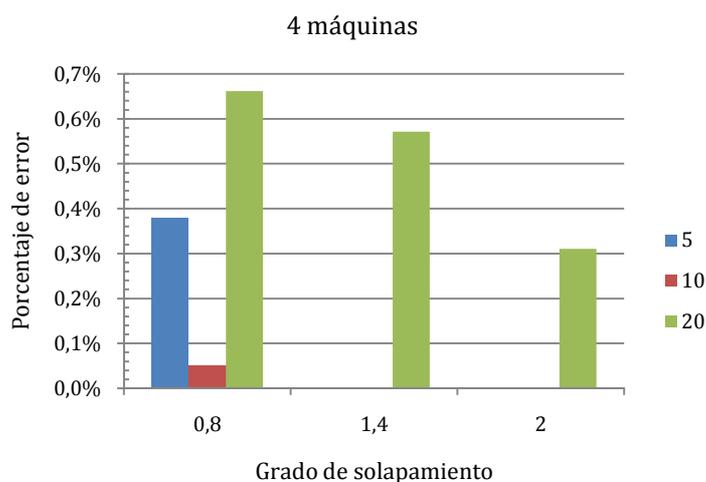


Figura 6.1. Error en problemas de 25 trabajos

En la figura 6.1 vemos cómo el error cometido por el método *VIRUS* es inferior al 1% en todos los casos con 25 trabajos, además, a medida que aumenta el grado de solapamiento, el error va disminuyendo. Esto es debido a que al aumentar este parámetro, disminuye la cantidad de combinaciones de distribución de trabajos posible. Aumentar el grado de solapamiento también hace que el cálculo de vecindades sea más complejo.

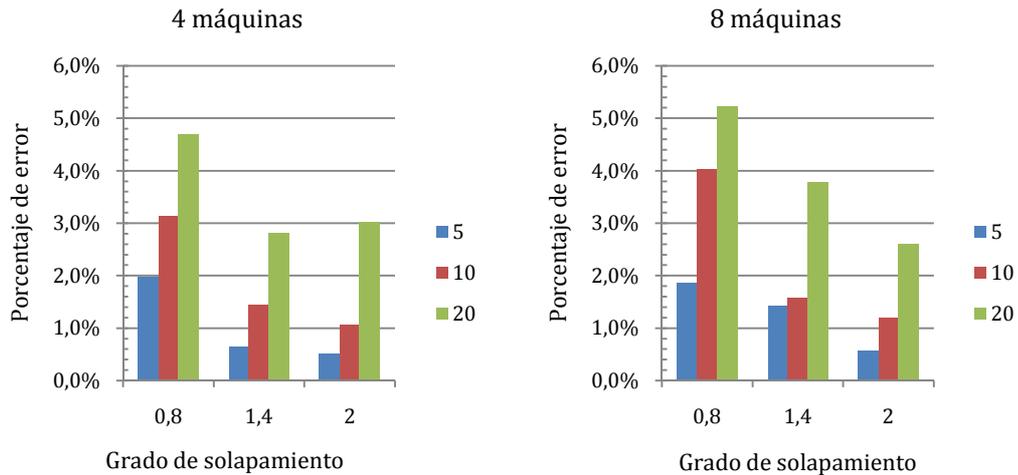


Figura 6.2. Error en problemas de 50 trabajos

Con 50 trabajos (figura 6.2) el error aumenta, y al igual que en el caso anterior, va disminuyendo conforme se incrementa el grado de solapamiento.

El error también se ve incrementado al aumentar el tamaño máximo de la ventana de comienzos. Esto es así porque los trabajos pueden realizarse en muchos más instantes de tiempo, lo que incrementa la cantidad de combinaciones posibles, y con ello, la complejidad. Como vimos antes, al aumentar la complejidad del problema se precisan más iteraciones para una mejor resolución, pero ello influye negativamente en el tiempo de cálculo.

Como se observa en la figura 6.3, no se ha hecho la comparativa con los problemas de 100 trabajos y ventana máxima de comienzos de 20. Esto es porque no se han podido obtener todos los valores necesarios con el optimizador, ya que la resolución de alguno de estos problemas superaba las 16 horas.

Siendo esto así no merece la pena obtener los resultados exactos de esos problemas, pues viendo el crecimiento del error, podemos asegurar casi completamente que el error en media no iba a estar por encima del 6%. Una ventana máxima de comienzos de 20 es la barrera de complejidad para los problemas de 100 trabajos a partir de la cual hay que usar métodos metaheurísticos, como el propuesto aquí.

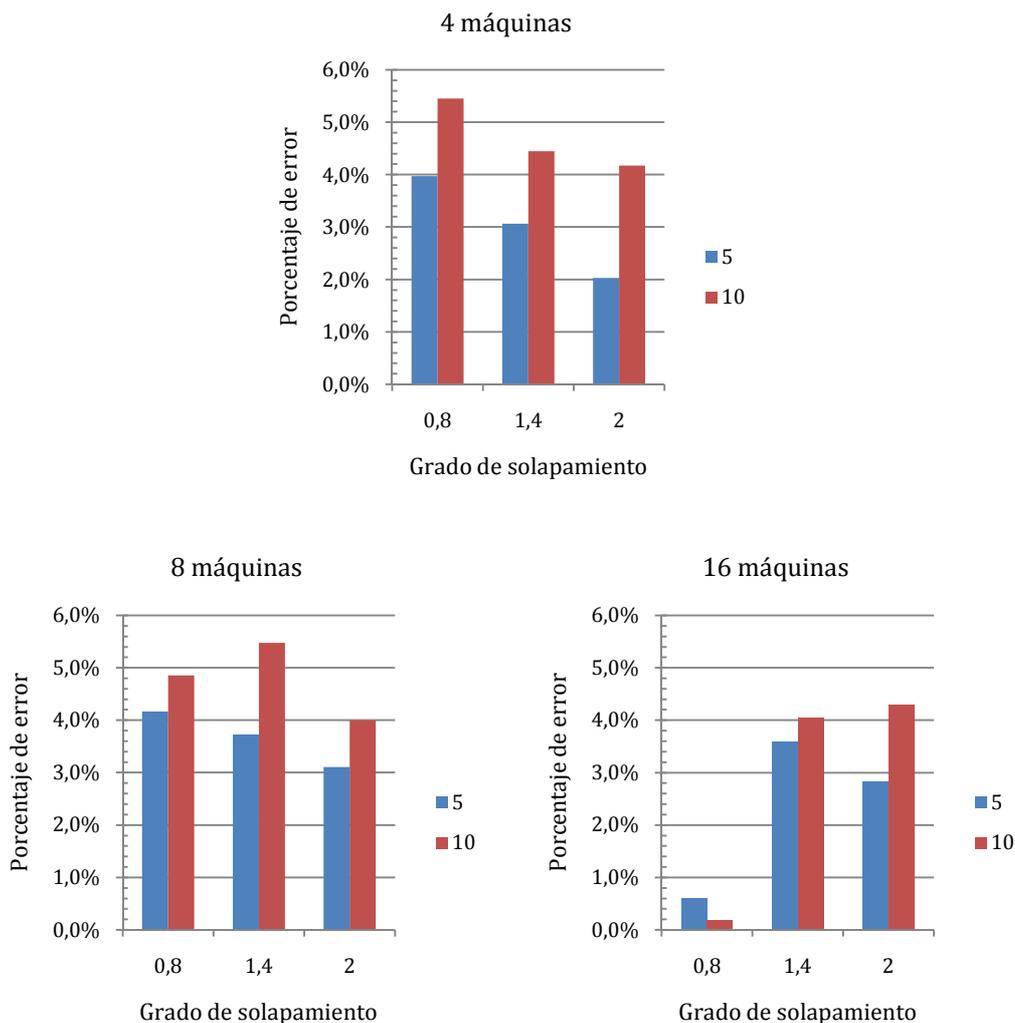


Figura 6.3. Error en problemas de 100 trabajos

Para ilustrar la ventaja de los métodos metaheurísticos, en la figura 6.4 se muestran los tiempos de procesamiento en media de los problemas con 100 trabajos de los que no se ha sacado el error. Además, en el anexo II se incluyen los tiempos de resolución de todos los problemas mediante el método *VIRUS*.

Vemos que los tiempos son bastante moderados, el que más tarda está por debajo de los seis minutos, y como se ha dicho antes, el error no llega a ser mayor al 6% en media. Aquí podemos observar la ventaja de la metaheurística sobre los algoritmos exactos, pues para resolver problemas de complejidad similar con el optimizador se necesitan muchísimas horas, mientras que con el método, en pocos minutos se obtiene una solución con un error bajo.

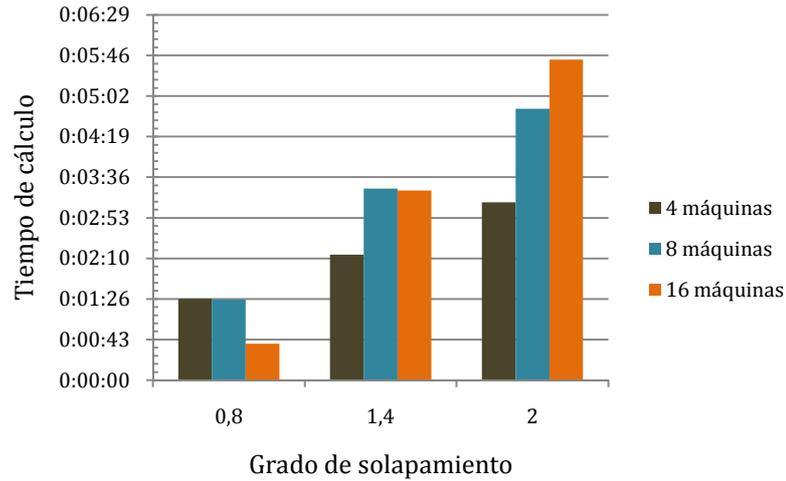


Figura 6.4. Tiempo de cálculo con VIRUS de problemas con 100 trabajos y máxima ventana 20

Otro modo de ver la bondad del método desarrollado aquí es observar, en cada serie de diez instancias, cuántas veces se alcanza el óptimo. Como se ha dicho antes, esta comparativa sólo puede hacerse para problemas de hasta 100 trabajos con ventana de comienzos de 10 iteraciones. Los resultados se muestran en las tablas 6.18 y 6.19.

Nº de trabajos	Nº de máquinas	Máxima ventana de comienzos	Grado de solapamiento	Nº de óptimos alcanzados
25	4	5	0.8	9
			1.4	10
			2	10
		10	0.8	8
			1.4	10
			2	10
		20	0.8	6
			1.4	7
			2	9
50	4	5	0.8	0
			1.4	5
			2	5
		10	0.8	0
			1.4	2
			2	5
		20	0.8	0
			1.4	0
			2	1

Tabla 6.18. Número de óptimos I

Nº de trabajos	Nº de máquinas	Máxima ventana de comienzos	Grado de solapamiento	Nº de óptimos alcanzados
50	8	5	0.8	0
			1.4	2
			2	5
		10	0.8	0
			1.4	1
			2	1
	20	0.8	0	
		1.4	0	
		2	1	
100	4	5	0.8	0
			1.4	0
			2	0
		10	0.8	0
			1.4	0
			2	0
	8	5	0.8	0
			1.4	0
			2	0
		10	0.8	0
			1.4	0
			2	0
	16	5	0.8	0
			1.4	0
			2	0
		10	0.8	3
			1.4	0
			2	0

Tabla 6.19. Número de óptimos II

Es fácil comprobar lo expuesto anteriormente, pues vemos cómo, en general, a medida que aumenta el grado de solapamiento en todos los casos, el método se comporta mejor (tiene más aciertos). También, como era de esperar, con menos trabajos alcanza más veces el óptimo, pues para problemas grandes se necesitan más iteraciones si se pretende alcanzar el óptimo.

Es obvio que el número de óptimos alcanzados varía también con la ventana máxima de comienzos y con el número de máquinas, pues al aumentar cualquiera de estos dos parámetros, aumenta la complejidad, y el método necesita más iteraciones, más tiempo.