

Trabajo Fin de Grado
Grado en Ingeniería de las Tecnologías
Industriales

Análisis de la Programación de Operaciones en un
taller de flujo con operaciones perdidas

Autor: José María Mena Ramírez

Tutor: José Manuel Framiñan Torres

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

Sevilla, 2014



Trabajo Fin de Grado Ingeniería
de las Tecnologías Industriales

Análisis de la Programación de Operaciones en un taller de flujo con operaciones perdidas

Autor:

José María Mena Ramírez

Tutor:

Dr. José Manuel Framiñan Torres

Dep. de Organización Industrial y Gestión de Empresas I

Escuela Técnica Superior de Ingeniería

Universidad de Sevilla

Sevilla, 2014

Trabajo Fin de Grado: Análisis de la Programación de Operaciones en un taller de flujo con operaciones perdidas

Autor: José María Mena Ramírez

Tutor: Dr. José Manuel Framiñan Torres

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

Presidente:

Vocales:

Secretario:

Acuerdan otorgarle la calificación de:

Sevilla, 2014

El Secretario del Tribunal

Agradecimientos

Este trabajo es la culminación a años de esfuerzo y duro trabajo, llevando consigo un gran aprendizaje y desarrollo personal. Agradecer a mis padres y hermano por creer en mí y por la confianza que tienen depositada en mi persona. A todos los que me rodean y en especial a Alicia por el apoyo incondicional día tras día.

Por último, agradecer a José Manuel Framiñan Torres por su tiempo, dedicación y guía durante todo este trabajo.

José M^a Mena Ramírez

Sevilla, 2014

Índice

1.	Introducción.....	1
2.	Objeto.....	5
3.	Metodología.....	9
4.	Desarrollo de los algoritmos.....	11
4.1	Identificar algoritmos.....	11
4.1.1	Algoritmo Random.....	11
4.1.2	Algoritmo Tiempos medios.....	12
4.1.3	Algoritmo NEH.....	13
4.2	Pseudocódigos de los algoritmos.....	16
4.2.1	Pseudocódigo Random.....	16
4.2.2	Pseudocódigo Tiempos medios.....	16
4.2.3	Pseudocódigo NEH.....	17
5.	Experimentación.....	18
5.1	Validación.....	18
5.2	Comparación de los algoritmos.....	48
6.	Conclusiones.....	90
7.	Bibliografía.....	93
8.	Anexos.....	96
8.1	Anexo 1.....	96
8.1.1	Código Algoritmo Random.....	96
8.1.2	Código Algoritmo Tiempos Medios.....	97
8.1.3	Código Algoritmo NEH.....	99
8.2	Anexo 2.....	104
8.3	Anexo 3.....	112

Índice de tablas

Tabla 1: Matriz de tiempos de ejecución de la tarea J_i en la máquina M_j .

Tabla 2: Valores promedio para el experimento de 50 trabajos y 10 máquinas.

Tabla 3: Valores desviación para el experimento de 50 trabajos y 10 máquinas.

Tabla 4: Porcentaje de mejora respecto a NEH para el experimento de 50 trabajos y 10 máquinas.

Tabla 5: Valores promedio para el experimento de 50 trabajos y 20 máquinas.

Tabla 6: Valores desviación para el experimento de 50 trabajos y 20 máquinas.

Tabla 7: Porcentaje de mejora respecto a NEH para el experimento de 50 trabajos y 20 máquinas.

Tabla 8: Valores promedio para el experimento de 50 trabajos y 30 máquinas.

Tabla 9: Valores desviación para el experimento de 50 trabajos y 30 máquinas.

Tabla 10: Porcentaje de mejora respecto a NEH para el experimento de 50 trabajos y 30 máquinas.

Tabla 11: Valores promedio para el experimento de 50 trabajos y 40 máquinas.

Tabla 12: Valores desviación para el experimento de 50 trabajos y 40 máquinas.

Tabla 13: Porcentaje de mejora respecto a NEH para el experimento de 50 trabajos y 40 máquinas.

Tabla 14: Valores promedio para el experimento de 50 trabajos y 50 máquinas.

Tabla 15: Valores desviación para el experimento de 50 trabajos y 50 máquinas.

Tabla 16: Porcentaje de mejora respecto a NEH para el experimento de 50 trabajos y 50 máquinas.

Tabla 17: Valores promedio para el experimento de 150 trabajos y 10 máquinas.

Tabla 18: Valores desviación para el experimento de 150 trabajos y 10 máquinas.

Tabla 19: Porcentaje de mejora respecto a NEH para el experimento de 150 trabajos y 10 máquinas.

Tabla 20: Valores promedio para el experimento de 150 trabajos y 20 máquinas.

Tabla 21: Valores desviación para el experimento de 150 trabajos y 20 máquinas.

Tabla 22: Porcentaje de mejora respecto a NEH para el experimento de 150 trabajos y 20 máquinas.

Tabla 23: Valores promedio para el experimento de 150 trabajos y 30 máquinas.

Tabla 24: Valores desviación para el experimento de 150 trabajos y 30 máquinas.

Tabla 25: Porcentaje de mejora respecto a NEH para el experimento de 150 trabajos y 30 máquinas.

Tabla 26: Valores promedio para el experimento de 150 trabajos y 40 máquinas.

Tabla 27: Valores desviación para el experimento de 150 trabajos y 40 máquinas.

Tabla 28: Porcentaje de mejora respecto a NEH para el experimento de 150 trabajos y 40 máquinas.

Tabla 29: Valores promedio para el experimento de 150 trabajos y 50 máquinas.

Tabla 30: Valores desviación para el experimento de 150 trabajos y 50 máquinas.

Tabla 31: Porcentaje de mejora respecto a NEH para el experimento de 150 trabajos y 50 máquinas.

Tabla 32: Valores promedio para el experimento de 250 trabajos y 10 máquinas.

Tabla 33: Valores desviación para el experimento de 250 trabajos y 10 máquinas.

Tabla 34: Porcentaje de mejora respecto a NEH para el experimento de 250 trabajos y 10 máquinas.

Tabla 35: Valores promedio para el experimento de 250 trabajos y 20 máquinas.

Tabla 36: Valores desviación para el experimento de 250 trabajos y 20 máquinas.

Tabla 37: Porcentaje de mejora respecto a NEH para el experimento de 250 trabajos y 20 máquinas.

Tabla 38: Valores promedio para el experimento de 250 trabajos y 30 máquinas.

Tabla 39: Valores desviación para el experimento de 250 trabajos y 30 máquinas.

Tabla 40: Porcentaje de mejora respecto a NEH para el experimento de 250 trabajos y 30 máquinas.

Tabla 41: Valores promedio para el experimento de 250 trabajos y 40 máquinas.

Tabla 42: Valores desviación para el experimento de 250 trabajos y 40 máquinas.

Tabla 43: Porcentaje de mejora respecto a NEH para el experimento de 250 trabajos y 40 máquinas.

Tabla 44: Valores promedio para el experimento de 250 trabajos y 50 máquinas.

Tabla 45: Valores desviación para el experimento de 250 trabajos y 50 máquinas.

Tabla 46: Porcentaje de mejora respecto a NEH para el experimento de 250 trabajos y 50 máquinas.

Tabla 47: Valores promedio para el experimento de 400 trabajos y 10 máquinas.

Tabla 48: Valores desviación para el experimento de 400 trabajos y 10 máquinas.

Tabla 49: Porcentaje de mejora respecto a NEH para el experimento de 400 trabajos y 10 máquinas.

Tabla 50: Valores promedio para el experimento de 400 trabajos y 20 máquinas.

Tabla 51: Valores desviación para el experimento de 400 trabajos y 20 máquinas.

Tabla 52: Porcentaje de mejora respecto a NEH para el experimento de 400 trabajos y 20 máquinas.

Tabla 53: Valores promedio para el experimento de 400 trabajos y 30 máquinas.

Tabla 54: Valores desviación para el experimento de 400 trabajos y 30 máquinas.

Tabla 55: Porcentaje de mejora respecto a NEH para el experimento de 400 trabajos y 30 máquinas.

Tabla 56: Valores promedio para el experimento de 400 trabajos y 40 máquinas.

Tabla 57: Valores desviación para el experimento de 400 trabajos y 40 máquinas.

Tabla 58: Porcentaje de mejora respecto a NEH para el experimento de 400 trabajos y 40 máquinas.

Tabla 59: Valores promedio para el experimento de 400 trabajos y 50 máquinas.

Tabla 60: Valores desviación para el experimento de 400 trabajos y 50 máquinas.

Tabla 61: Porcentaje de mejora respecto a NEH para el experimento de 400 trabajos y 50 máquinas.

Tabla 62: Valores promedio para el experimento de 500 trabajos y 10 máquinas.

Tabla 63: Valores desviación para el experimento de 500 trabajos y 10 máquinas.

Tabla 64: Porcentaje de mejora respecto a NEH para el experimento de 500 trabajos y 10 máquinas.

Tabla 65: Valores promedio para el experimento de 500 trabajos y 20 máquinas.

Tabla 66: Valores desviación para el experimento de 500 trabajos y 20 máquinas.

Tabla 67: Porcentaje de mejora respecto a NEH para el experimento de 500 trabajos y 20 máquinas.

Tabla 68: Valores promedio para el experimento de 500 trabajos y 30 máquinas.

Tabla 69: Valores desviación para el experimento de 500 trabajos y 30 máquinas.

Tabla 70: Porcentaje de mejora respecto a NEH para el experimento de 500 trabajos y 30 máquinas.

Tabla 71: Valores promedio para el experimento de 500 trabajos y 40 máquinas.

Tabla 72: Valores desviación para el experimento de 500 trabajos y 40 máquinas.

Tabla 73: Porcentaje de mejora respecto a NEH para el experimento de 500 trabajos y 40 máquinas.

Tabla 74: Valores promedio para el experimento de 500 trabajos y 50 máquinas.

Tabla 75: Valores desviación para el experimento de 500 trabajos y 50 máquinas.

Tabla 76: Porcentaje de mejora respecto a NEH para el experimento de 500 trabajos y 50 máquinas.

Tabla 77: Valores medios makespan de las 30 instancias de cada experimento para cada algoritmo utilizado, en el tipo en el que el 10% de los tiempos de procesamientos son cero.

Tabla 78: Valores medios makespan de las 30 instancias de cada experimento para cada algoritmo utilizado, en el tipo en el que el 20% de los tiempos de procesamientos son cero.

Tabla 79: Valores medios makespan de las 30 instancias de cada experimento para cada algoritmo utilizado, en el tipo en el que el 30% de los tiempos de procesamientos son cero.

Tabla 80: Valores medios makespan de las 30 instancias de cada experimento para cada algoritmo utilizado, en el tipo en el que el 40% de los tiempos de procesamientos son cero.

Tabla 81: Valores medios de la desviación típica de las 30 instancias de cada experimento para cada algoritmo utilizado, en el tipo en el que el 10% de los tiempos de procesamientos son cero.

Tabla 82: Valores medios de la desviación típica de las 30 instancias de cada experimento para cada algoritmo utilizado, en el tipo en el que el 20% de los tiempos de procesamientos son cero.

Tabla 83: Valores medios de la desviación típica de las 30 instancias de cada experimento para cada algoritmo utilizado, en el tipo en el que el 30% de los tiempos de procesamientos son cero.

Tabla 84: Valores medios de la desviación típica de las 30 instancias de cada experimento para cada algoritmo utilizado, en el tipo en el que el 40% de los tiempos de procesamientos son cero.

Índice de figuras

Figura 1: Proceso de Planificación, programación y control de la producción.

Figura 2: Modelo Flow Shop.

Figura 3: Cálculo del makespan para la secuencia $J_2 - J_3$.

Figura 4: Cálculo del makespan para la secuencia $J_3 - J_2$.

Figura 5: Cálculo del makespan para la secuencia $J_1 - J_3 - J_2$.

Figura 6: Cálculo del makespan para la secuencia $J_3 - J_1 - J_2$.

Figura 7: Cálculo del makespan para la secuencia $J_3 - J_2 - J_1$.

Figura 8: Evolución del makespan en los experimentos para cada algoritmo, en el tipo en el que el 10% de los tiempos de procesamientos son cero.

Figura 9: Evolución del makespan en los experimentos para cada algoritmo, en el tipo en el que el 20% de los tiempos de procesamientos son cero.

Figura 10: Evolución del makespan en los experimentos para cada algoritmo, en el tipo en el que el 30% de los tiempos de procesamientos son cero.

Figura 11: Evolución del makespan en los experimentos para cada algoritmo, en el tipo en el que el 40% de los tiempos de procesamientos son cero.

Figura 12: Valor medio del makespan de los algoritmos para el tipo de 10% de los tiempos ceros.

Figura 13: Valor medio del makespan de los algoritmos para el tipo de 20% de los tiempos ceros.

Figura 14: Valor medio del makespan de los algoritmos para el tipo de 30% de los tiempos ceros.

Figura 15: Valor medio del makespan de los algoritmos para el tipo de 40% de los tiempos ceros.

Figura 16: Evolución de la desviación típica en los experimentos para cada algoritmo, en el tipo en el que el 10% de los tiempos de procesamientos son cero.

Figura 17: Evolución de la desviación típica en los experimentos para cada algoritmo, en el tipo en el que el 20% de los tiempos de procesamientos son cero.

Figura 18: Evolución de la desviación típica en los experimentos para cada algoritmo, en el tipo en el que el 30% de los tiempos de procesamientos son cero.

Figura 19: Evolución de la desviación típica en los experimentos para cada algoritmo, en el tipo en el que el 40% de los tiempos de procesamientos son cero.

Figura 20: Valor medio de la desviación típica de los algoritmos para todos los experimentos y cada tipo de algoritmo, en el tipo en el que el 10% de los tiempos de procesamientos son cero.

Figura 21: Valor medio de la desviación típica de los algoritmos para todos los experimentos y cada tipo de algoritmo, en el tipo en el que el 20% de los tiempos de procesamientos son cero.

Figura 22: Valor medio de la desviación típica de los algoritmos para todos los experimentos y cada tipo de algoritmo, en el tipo en el que el 30% de los tiempos de procesamientos son cero.

Figura 23: Valor medio de la desviación típica de los algoritmos para todos los experimentos y cada tipo de algoritmo, en el tipo en el que el 40% de los tiempos de procesamientos son cero.

Figura 24: Valor promedio frente a porcentaje de cero para cada algoritmo del experimento de 50 trabajos y 10 máquinas.

Figura 25: Valor desviación típica frente a porcentaje de cero para cada algoritmo del experimento de 50 trabajos y 10 máquinas.

Figura 26: Porcentaje de mejora respecto a NEH de los algoritmos Random y T.Medio frente al porcentaje de ceros para el experimento de 50 trabajos y 10 máquinas.

Figura 27: Valor promedio frente a porcentaje de cero para cada algoritmo del experimento de 50 trabajos y 20 máquinas.

Figura 28: Valor desviación típica frente a porcentaje de cero para cada algoritmo del experimento de 50 trabajos y 20 máquinas.

Figura 29: Porcentaje de mejora respecto a NEH de los algoritmos Random y T.Medio frente al porcentaje de ceros para el experimento de 50 trabajos y 20 máquinas.

Figura 30: Valor promedio frente a porcentaje de cero para cada algoritmo del experimento de 50 trabajos y 30 máquinas.

Figura 31: Valor desviación típica frente a porcentaje de cero para cada algoritmo del experimento de 50 trabajos y 30 máquinas.

Figura 32: Porcentaje de mejora respecto a NEH de los algoritmos Random y T.Medio frente al porcentaje de ceros para el experimento de 50 trabajos y 30 máquinas.

Figura 33: Valor promedio frente a porcentaje de cero para cada algoritmo del experimento de 50 trabajos y 40 máquinas.

Figura 34: Valor desviación típica frente a porcentaje de cero para cada algoritmo del experimento de 50 trabajos y 40 máquinas.

Figura 35: Porcentaje de mejora respecto a NEH de los algoritmos Random y T.Medio frente al porcentaje de ceros para el experimento de 50 trabajos y 40 máquinas.

Figura 36: Valor promedio frente a porcentaje de cero para cada algoritmo del experimento de 50 trabajos y 50 máquinas.

Figura 37: Valor desviación típica frente a porcentaje de cero para cada algoritmo del experimento de 50 trabajos y 50 máquinas.

Figura 38: Porcentaje de mejora respecto a NEH de los algoritmos Random y T.Medio frente al porcentaje de ceros para el experimento de 50 trabajos y 50 máquinas.

Figura 39: Valor promedio frente a porcentaje de cero para cada algoritmo del experimento de 150 trabajos y 10 máquinas.

Figura 40: Valor desviación típica frente a porcentaje de cero para cada algoritmo del experimento de 150 trabajos y 10 máquinas.

Figura 41: Porcentaje de mejora respecto a NEH de los algoritmos Random y T.Medio frente al porcentaje de ceros para el experimento de 150 trabajos y 10 máquinas.

Figura 42: Valor promedio frente a porcentaje de cero para cada algoritmo del experimento de 150 trabajos y 20 máquinas.

Figura 43: Valor desviación típica frente a porcentaje de cero para cada algoritmo del experimento de 150 trabajos y 20 máquinas.

Figura 44: Porcentaje de mejora respecto a NEH de los algoritmos Random y T.Medio frente al porcentaje de ceros para el experimento de 150 trabajos y 20 máquinas.

Figura 45: Valor promedio frente a porcentaje de cero para cada algoritmo del experimento de 150 trabajos y 30 máquinas.

Figura 46: Valor desviación típica frente a porcentaje de cero para cada algoritmo del experimento de 150 trabajos y 30 máquinas.

Figura 47: Porcentaje de mejora respecto a NEH de los algoritmos Random y T.Medio frente al porcentaje de ceros para el experimento de 150 trabajos y 30 máquinas.

Figura 48: Valor promedio frente a porcentaje de cero para cada algoritmo del experimento de 150 trabajos y 40 máquinas.

Figura 49: Valor desviación típica frente a porcentaje de cero para cada algoritmo del experimento de 150 trabajos y 40 máquinas.

Figura 50: Porcentaje de mejora respecto a NEH de los algoritmos Random y T.Medio frente al porcentaje de ceros para el experimento de 150 trabajos y 40 máquinas.

Figura 51: Valor promedio frente a porcentaje de cero para cada algoritmo del experimento de 150 trabajos y 50 máquinas.

Figura 52: Valor desviación típica frente a porcentaje de cero para cada algoritmo del experimento de 150 trabajos y 50 máquinas.

Figura 53: Porcentaje de mejora respecto a NEH de los algoritmos Random y T.Medio frente al porcentaje de ceros para el experimento de 150 trabajos y 50 máquinas.

Figura 54: Valor promedio frente a porcentaje de cero para cada algoritmo del experimento de 250 trabajos y 10 máquinas.

Figura 55: Valor desviación típica frente a porcentaje de cero para cada algoritmo del experimento de 250 trabajos y 10 máquinas.

Figura 56: Porcentaje de mejora respecto a NEH de los algoritmos Random y T.Medio frente al porcentaje de ceros para el experimento de 250 trabajos y 10 máquinas.

Figura 57: Valor promedio frente a porcentaje de cero para cada algoritmo del experimento de 250 trabajos y 20 máquinas.

Figura 58: Valor desviación típica frente a porcentaje de cero para cada algoritmo del experimento de 250 trabajos y 20 máquinas.

Figura 59: Porcentaje de mejora respecto a NEH de los algoritmos Random y T.Medio frente al porcentaje de ceros para el experimento de 250 trabajos y 20 máquinas.

Figura 60: Valor promedio frente a porcentaje de cero para cada algoritmo del experimento de 250 trabajos y 30 máquinas.

Figura 61: Valor desviación típica frente a porcentaje de cero para cada algoritmo del experimento de 250 trabajos y 30 máquinas.

Figura 62: Porcentaje de mejora respecto a NEH de los algoritmos Random y T.Medio frente al porcentaje de ceros para el experimento de 250 trabajos y 30 máquinas.

Figura 63: Valor promedio frente a porcentaje de cero para cada algoritmo del experimento de 250 trabajos y 40 máquinas.

Figura 64: Valor desviación típica frente a porcentaje de cero para cada algoritmo del experimento de 250 trabajos y 40 máquinas.

Figura 65: Porcentaje de mejora respecto a NEH de los algoritmos Random y T.Medio frente al porcentaje de ceros para el experimento de 250 trabajos y 40 máquinas.

Figura 66: Valor promedio frente a porcentaje de cero para cada algoritmo del experimento de 250 trabajos y 50 máquinas.

Figura 67: Valor desviación típica frente a porcentaje de cero para cada algoritmo del experimento de 250 trabajos y 50 máquinas.

Figura 68: Porcentaje de mejora respecto a NEH de los algoritmos Random y T.Medio frente al porcentaje de ceros para el experimento de 250 trabajos y 50 máquinas.

Figura 69: Valor promedio frente a porcentaje de cero para cada algoritmo del experimento de 400 trabajos y 10 máquinas.

Figura 70: Valor desviación típica frente a porcentaje de cero para cada algoritmo del experimento de 400 trabajos y 10 máquinas.

Figura 71: Porcentaje de mejora respecto a NEH de los algoritmos Random y T.Medio frente al porcentaje de ceros para el experimento de 400 trabajos y 10 máquinas.

Figura 72: Valor promedio frente a porcentaje de cero para cada algoritmo del experimento de 400 trabajos y 20 máquinas.

Figura 73: Valor desviación típica frente a porcentaje de cero para cada algoritmo del experimento de 400 trabajos y 20 máquinas.

Figura 74: Porcentaje de mejora respecto a NEH de los algoritmos Random y T.Medio frente al porcentaje de ceros para el experimento de 400 trabajos y 20 máquinas.

Figura 75: Valor promedio frente a porcentaje de cero para cada algoritmo del experimento de 400 trabajos y 30 máquinas.

Figura 76: Valor desviación típica frente a porcentaje de cero para cada algoritmo del experimento de 400 trabajos y 30 máquinas.

Figura 77: Porcentaje de mejora respecto a NEH de los algoritmos Random y T.Medio frente al porcentaje de ceros para el experimento de 400 trabajos y 30 máquinas.

Figura 78: Valor promedio frente a porcentaje de cero para cada algoritmo del experimento de 400 trabajos y 40 máquinas.

Figura 79: Valor desviación típica frente a porcentaje de cero para cada algoritmo del experimento de 400 trabajos y 40 máquinas.

Figura 80: Porcentaje de mejora respecto a NEH de los algoritmos Random y T.Medio frente al porcentaje de ceros para el experimento de 400 trabajos y 40 máquinas.

Figura 81: Valor promedio frente a porcentaje de cero para cada algoritmo del experimento de 400 trabajos y 50 máquinas.

Figura 82: Valor desviación típica frente a porcentaje de cero para cada algoritmo del experimento de 400 trabajos y 50 máquinas.

Figura 83: Porcentaje de mejora respecto a NEH de los algoritmos Random y T.Medio frente al porcentaje de ceros para el experimento de 400 trabajos y 50 máquinas.

Figura 84: Valor promedio frente a porcentaje de cero para cada algoritmo del experimento de 500 trabajos y 10 máquinas.

Figura 85: Valor desviación típica frente a porcentaje de cero para cada algoritmo del experimento de 500 trabajos y 10 máquinas.

Figura 86: Porcentaje de mejora respecto a NEH de los algoritmos Random y T.Medio frente al porcentaje de ceros para el experimento de 500 trabajos y 10 máquinas.

Figura 87: Valor promedio frente a porcentaje de cero para cada algoritmo del experimento de 500 trabajos y 20 máquinas.

Figura 88: Valor desviación típica frente a porcentaje de cero para cada algoritmo del experimento de 500 trabajos y 20 máquinas.

Figura 89: Porcentaje de mejora respecto a NEH de los algoritmos Random y T.Medio frente al porcentaje de ceros para el experimento de 500 trabajos y 20 máquinas.

Figura 90: Valor promedio frente a porcentaje de cero para cada algoritmo del experimento de 500 trabajos y 30 máquinas.

Figura 91: Valor desviación típica frente a porcentaje de cero para cada algoritmo del experimento de 500 trabajos y 30 máquinas.

Figura 92: Porcentaje de mejora respecto a NEH de los algoritmos Random y T.Medio frente al porcentaje de ceros para el experimento de 500 trabajos y 30 máquinas.

Figura 93: Valor promedio frente a porcentaje de cero para cada algoritmo del experimento de 500 trabajos y 40 máquinas.

Figura 94: Valor desviación típica frente a porcentaje de cero para cada algoritmo del experimento de 500 trabajos y 40 máquinas.

Figura 95: Porcentaje de mejora respecto a NEH de los algoritmos Random y T.Medio frente al porcentaje de ceros para el experimento de 500 trabajos y 40 máquinas.

Figura 96: Valor promedio frente a porcentaje de cero para cada algoritmo del experimento de 500 trabajos y 50 máquinas.

Figura 97: Valor desviación típica frente a porcentaje de cero para cada algoritmo del experimento de 500 trabajos y 50 máquinas.

Figura 98: Porcentaje de mejora respecto a NEH de los algoritmos Random y T.Medio frente al porcentaje de ceros para el experimento de 500 trabajos y 50 máquinas.

Glosario

N: Número de trabajos.

M: Número de máquinas.

C_{MAX}: Objetivo con el fin de minimizar el tiempo de finalización de todos los trabajos.

Makespan: Longitud de la programación de la secuencia de trabajos.

Schedule: Programa de trabajo o programa de producción, planificación.

Random: Palabra inglesa referida al azar, en este trabajo se utiliza para describir el algoritmo Random con el que los trabajos son ordenados de forma aleatoria.

Algoritmo: Es un conjunto de instrucciones o reglas bien definidas, ordenadas y finitas que permite realizar una actividad mediante pasos sucesivos.

Algoritmo de Johnson: Es un algoritmo que va construyendo la secuencia de tareas a ser procesadas, paso a paso, siendo el orden de menor mayor tiempo de proceso.

Flow-Shop: Problema de secuenciación de tareas en sistemas de producción lineal que consiste en programar la secuencia de procesamiento de N tareas en M máquinas de tal forma que se optimice alguna medida de efectividad como por ejemplo minimizar el tiempo total requerido para terminar todas las tareas (makespan). El caso más general de este problema considera que todas las tareas deben ser procesadas en el mismo orden en cada máquina.

Función Objetivo: La función objetivo define la medida de efectividad del sistema como una función matemática de las variables de decisión. La solución óptima será aquella que produzca el mejor valor de la función objetivo.

Heurísticas: Son algoritmos que encuentran soluciones de buena calidad para problemas combinatorias complejos con esfuerzos computacionales relativamente

pequeños, pero que desde el punto de vista teórico renuncian a encontrar la solución global del problema.

Optimización: O programación matemática es una rama de las matemáticas que intenta dar respuesta a un tipo general de problemas donde se desea establecer la mejor solución para un función objetivo en un espacio de soluciones.

Programación lineal: Es una técnica de programación matemática la cual se caracteriza porque tanto el espacio de soluciones como la función objetivo tienen un comportamiento lineal.

Programación de Tareas: Es la asignación secuencial de trabajos a un conjunto de máquinas. Las tareas pueden seguir un orden o no dentro del grupo de máquinas.

Tiempo de Flujo: Es el tiempo del trabajo i , que transcurre desde el inicio del primer trabajo en la primera máquina hasta que el trabajo i sale del sistema. En este tiempo se incluye el tiempo de espera.

1. Introducción

El presente estudio intenta analizar las soluciones de la programación de las operaciones en un taller de flujo regular en el que existen operaciones perdidas, es decir, algunos trabajos se pueden saltar algunas etapas de la producción. Por ello, el título de este trabajo fin de grado es Análisis de la programación de operaciones en un taller de flujo con operaciones perdidas.

En concreto, se han analizado el comportamiento de varios métodos aproximados. En primer lugar, el algoritmo Random, presentado como la alternativa a no realizar ninguna acción referente a la planificación de los trabajos. En segundo lugar, la utilización de un algoritmo basado en la clasificación de los trabajos por sus tiempos medios subrayando su simplicidad de cara a mejorar los resultados. Y por último, la heurística NEH, mostrada como la opción más eficiente para este problema y por lo tanto será la que se considere en la experimentación.

El presente trabajo fin de grado pertenece al departamento de Organización Industrial y Gestión de empresas I. Los conocimientos necesarios para abordar dicho estudio comprenden a todas las asignaturas de la rama de organización y producción de la titulación de grado en ingeniería de las tecnologías industriales. Dichas asignaturas dotan de la preparación necesaria en los ámbitos de la gestión, organización de la producción y programación de operaciones.

El proceso de planificación y control de la producción consta de una planificación estratégica a largo plazo, una planificación agregada o a medio plazo, un plan maestro de la producción, una programación de las operaciones y una ejecución y control de la fabricación (Domínguez-Machuca, 1995). Estas fases deben estar relacionadas entre sí y presentar un enfoque estratégico empresarial determinado para alcanzar los objetivos tácticos y operativos que se marque la empresa, ver figura 1. Cualquier empresa dedicada a la producción, independientemente de su actividad y dimensión, debe tener presente el modelo de gestión y planificación de

la fabricación aunque la forma como éstas se desarrollen dependerá de las características propias de cada sistema productivo.

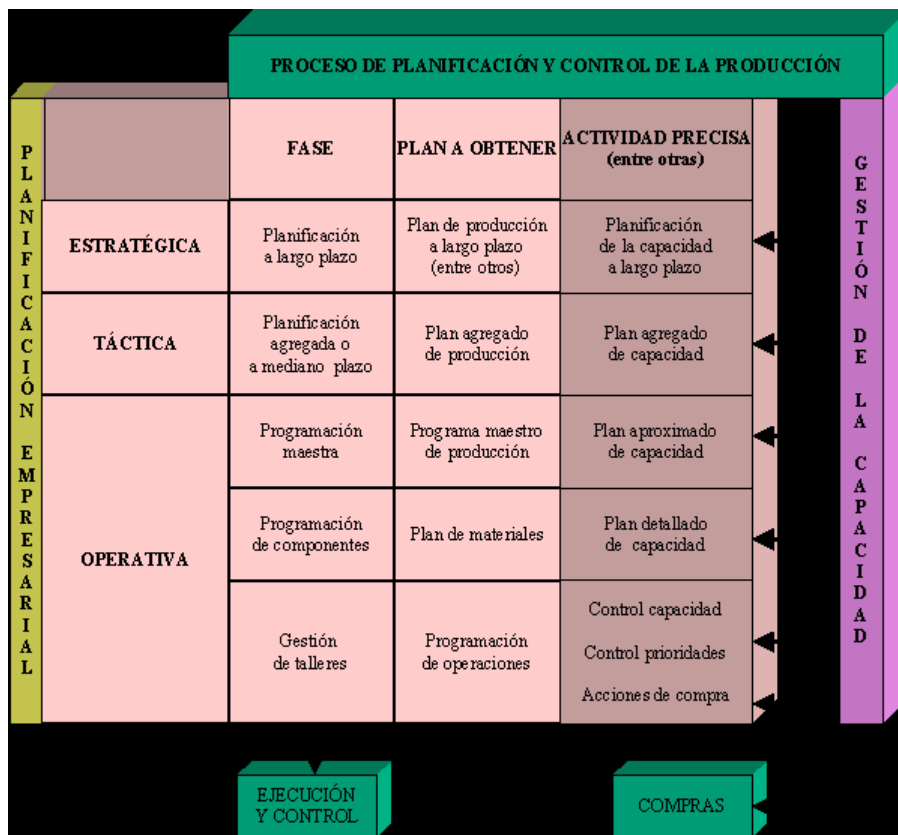


Figura 1: Proceso de Planificación, programación y control de la producción. Fuente: Domínguez-Machuca, 1995.

La planificación de la producción define el momento de fabricación de los productos y establece las actividades a realizar para conseguir un volumen de producción que permita satisfacer la demanda estimada, constituyendo un equilibrio entre la capacidad de los diferentes niveles y la producción (Martínez-Sánchez, 2003). Por otro lado, busca contribuir a alcanzar los objetivos marcados de coste, calidad, plazo de entrega y servicio al cliente, a través de una planificación eficaz y eficiente, en términos de minimizar el coste de los recursos necesarios para satisfacer la demanda durante un determinado período, optimizando los procesos y contribuyendo al aprovechamiento de los recursos mejorando o igualando la calidad del producto, es decir, alta productividad reduciendo las tasas de fracaso y error (Velasco-Sánchez, 2013).

Se debe tener presente que un sistema de planificación y control de la producción efectivo puede proporcionar una ventaja competitiva sustancial para una empresa en su mercado. Del mismo modo una acertada elección de la estrategia comercial podrá ser el elemento diferenciador de cara a la competencia y ser los elegidos por los clientes.

Otro enfoque de la planificación es el propósito de reducir la incertidumbre mediante una programación de actividades teniendo en cuenta los escenarios más probables donde se desarrollarán. Un continuo y exhaustivo estudio del mercado y de la propia situación de la empresa es imprescindible para planificar con cierto éxito (Fernández, Avella y Fernández, 2003). Incluso si se fracasa en el logro de los objetivos estratégicos determinados en un principio, en el proceso se adquiere un mayor conocimiento de la empresa, de sus posibilidades, de su entorno y de sus medios porque esa investigación que se ha desarrollado genera posibles ideas para futuros resultados de la empresa.

Considerando el alcance que tiene la planificación y control de la producción y viendo la organización como un todo integrado, es lógico que sus diferentes áreas estén asociadas y juntas para operar de acuerdo a las metas de la empresa. La correcta coordinación entre las diferentes áreas dependerá de la fluidez de información que circula entre ellas (Wollmann, Berry y Whybark, 2005).

En el plan de producción se decide qué productos hacer, cuántos, cómo y dónde se han de fabricar, y el control de la producción vigila que el proceso sea viable y llegue a lograrse sin problemas. La producción consiste en una secuencia de operaciones que transforma los materiales o materias primas, haciendo que pasen de una forma dada a otra que se desea obtener (Domínguez-Machuca, 1995). Manteniendo los niveles óptimos de stocks que faciliten el equilibrio entre las necesidades y los costes de mantenimiento, se alcanza una optimización de la gestión de inventarios y la capacidad productiva. Esto está estrechamente ligado al ahorro en las compras, puesto que manteniendo los niveles de inventarios actualizados se permite no almacenar más que lo necesario y es posible tener un mayor control en los suministros.

Por último, destacar que el conocimiento de la producción y las operaciones, la adaptación y la capacidad de evolucionar hacia nuevas tecnologías, son hoy en día herramientas esenciales en la gestión de la producción, dados los altos niveles de competencia, el encarecimiento de los recursos (en algunas ocasiones) y la internacionalización de las operaciones. Esto último, actualmente, se reconoce como esencial y necesario para las empresas puesto que es una vía de expansión y crecimiento. Aumenta la independencia de los ciclos económicos del mercado local, se accede a un mercado más amplio con más oportunidades de obtener mayor rentabilidad, se diversifican los riesgos comerciales y de aprovisionamiento, y aumenta la capacidad productiva, todo ello posterior a una definida y acertada estrategia que resulte exitosa (Cruelles-Ruiz, 2013).

2. Objeto

La programación de operaciones consiste en identificar, organizar y ordenar en secuencia lógica, todas las actividades derivadas del plan de producción (García-Sánchez, 2014). En este caso, nos centraremos en los trabajos que deben realizar las máquinas para llevar a cabo la fabricación de los productos. Uno de los aspectos más importantes a considerar es la asignación de dichos trabajos, teniendo en cuenta el tiempo en que se realizan, que debe estar dentro del tiempo total disponible para la entrega de los mismos. Lo anterior implica, también, preservar las relaciones de interdependencia de los trabajos, así como asignar tiempos y responsables de ejecución y comprobación de los recursos para cada uno de dichos trabajos.

Por ello el presente trabajo, trata de la programación de las operaciones en un taller de flujo con operaciones perdidas, una parte fundamental en la planificación de la producción. Dicha programación y posterior análisis se realizará mediante algoritmos conocidos.

Una correcta programación de las operaciones tiene un impacto muy favorable en la productividad y por lo tanto en los beneficios económicos que conseguirá la empresa cuando presente sus resultados. Del mismo modo, aumenta el servicio al cliente al conocer las fechas de entrega de los productos, y es posible ajustar la programación para que la emisión de éstos se produzca en fecha y no con antelación, que repercute en gastos de mantenimiento, y por supuesto, no con retrasos.

La secuenciación de trabajos en máquinas, universalmente conocido como scheduling, se puede definir como la asignación en el tiempo de los recursos disponibles con objeto de optimizar una determinada medida de comportamiento (Framiñan y Leisten, 2004). Existen diversas formas y características que pueden asociarse a los trabajos y al modo de procesamiento en el sistema. Un problema se determina según la arquitectura del taller, las características de los trabajos y el criterio de optimización.

La disposición de la maquinaria define el tipo de configuración productiva necesaria para realizar el proceso de fabricación. Las necesidades de los clientes se traducen en órdenes de trabajo que se liberan y "transforman" en trabajos con fecha de entrega determinada. Como ya se citó anteriormente, la programación de la producción se preocupa de la asignación de recursos limitados a tareas productivas, y debe realizarse de manera detallada para mantener la eficiencia y el control de las operaciones dentro del sistema productivo y así constituir una ventaja competitiva.

Los distintos productos que se fabrican requieren en su elaboración de distintas operaciones, las que se realizan en un orden y configuración productiva determinada, dependiendo del tipo de producto, su volumen de producción, la variedad de productos que se produce en la misma línea, etc. Este trabajo atiende al diseño de máquinas en serie cuyo modelo de paso para los diferentes trabajos por las máquinas es el mismo. Esto se denomina sistemas de flujo uniforme, Flow shop, en la que todos los trabajos pasan por cada una de las máquinas del sistema usando el mismo orden de paso por las mismas, denominado flujo unidireccional. Es decir, el problema de Flow-Shop o problema de secuenciación de tareas de producción lineal, se caracteriza porque todos los trabajos que van a ser programados siguen el mismo flujo de producción y los tiempos de ejecución varían en cada una de las etapas.

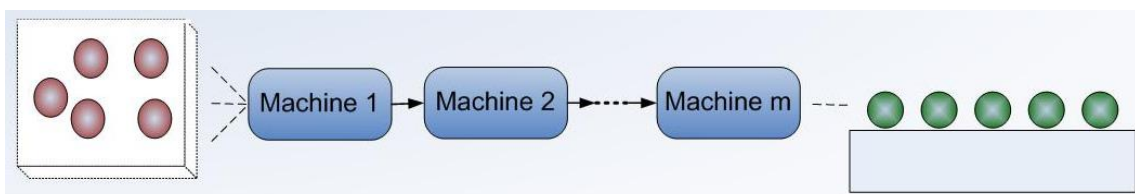


Figura 2: Modelo Flow Shop. Fuente: Wikipedia.

La idea básica del diseño de esta línea de producción consiste en la división del conjunto de operaciones a realizar para la obtención del producto y su asignación de manera coherente a una serie de máquinas, de tal manera que el producto puede ser fabricado o montado de una manera más económica y rápida. Para ello, el producto es transportado secuencialmente a lo largo de las máquinas, de tal manera que en cada una de ellos recibe sucesivamente las operaciones hasta conseguir el producto completo.

A partir de un determinado criterio de optimización, se trata de establecer la secuencia para el procesamiento de una serie de trabajos sobre un conjunto de máquinas. Para este trabajo se buscará minimizar el tiempo de finalización de todos los trabajos, C_{MAX} , también denominado longitud de la programación, makespan, teniendo como objetivo alcanzar una solución adecuada y satisfactoria para nuestro tipo de problema que se encuadra a continuación.

El problema de Flow-Shop, al igual que muchos otros, en el campo de secuenciación de tareas es de difícil solución. Esta característica hace que el problema sea de permutación, es decir, cada una de las combinatorias que pueden formarse deben contener las n tareas a procesar y una permutación se diferencia de otra únicamente en el orden de colocación de los elementos, obteniéndose de esta manera que el número de permutaciones posibles es $n!$. En la literatura especializada los problemas combinatorios están catalogados como NP-difícil, dado que no se cuentan con algoritmos de orden polinomial a partir de los cuales se pueda resolver el problema de forma óptima y en el menor tiempo posible. Se dice que un problema es NP-difícil cuando se demuestra que cualquier algoritmo de solución tiene un tiempo de ejecución que aumenta, en el peor de los casos, exponencialmente con el tamaño del problema. Esto invita a desarrollar estrategias de solución con las que se puedan aprovechar las características combinatorias del problema y adaptarlas de forma ágil a alguna técnica de solución para resolverlo en un tiempo computacional adecuado y de forma óptima. Por lo que elegimos el algoritmo NEH.

El problema se encuadra de la siguiente forma:

- Todos los trabajos son independientes y disponibles para ser procesados.
- Las máquinas están continuamente disponibles.
- Cada máquina solo puede procesar una tarea a la vez.
- Cada trabajo puede ser procesado solo en una máquina a la vez.
- Una vez un el procesamiento de un trabajo dado se ha iniciado en una máquina determinada, no puede ser interrumpido y el proceso continúa hasta ser completado.

- Los tiempos de preparación (set up) son independientes de la secuencia y son incluidos en el tiempo de proceso, o ignorados.

Para resolver este problema existen diferentes métodos exactos y heurísticos, constructivos o de mejora. Para el caso de dos máquinas, el clásico algoritmo de Johnson obtiene la solución óptima del problema con minimización de makespan y denotado por C_{MAX} (Johnson, 1954). Cuando se tienen tres o más máquinas, como es nuestro caso, se han desarrollado métodos heurísticos constructivos que entregan soluciones factibles, entre las que se destacan las conocidas heurísticas de Palmer, Gupta, MPS, CDS, RA y NEH.

Elegimos la heurística NEH porque para el problema que se plantea en este estudio ha mostrado ser la heurística más eficiente para encontrar la mejor solución (Fernandez-Viagas y Framiñan-Torres, 2014), y por lo tanto será la que se considere en la experimentación.

Los algoritmos están diseñados con el objetivo de minimizar el makespan en el problema sin tiempos de set up. Por lo que en el caso, en el que en la línea fabrique varios productos a la vez o se fabriquen varios modelos de productos muy diferentes entre sí, y existan ciertas diferencias entre las operaciones a realizar, se hace necesario invertir un cierto tiempo de cambio de partida en desmontar el utillaje usado en el anterior tipo de producto y montar el nuevo, ajustar maquinaria, lanzar lotes de prueba, etc. pero para nuestro estudio supondremos que estos tiempos de set up no son destacables.

3. Metodología

Este documento discute el modelo de programación de los trabajos tratando el proceso de planificación de la producción. Para ello se utiliza el programa informático Code::Blocks, formado por un conjunto de herramientas de programación para el desarrollo de programas en lenguaje C y C++.

Code::Blocks tiene como soporte de apoyo a la programación, espacios de trabajo (workspaces) para combinar múltiples proyectos adaptables para cambiar según la tarea que se esté realizando o como se configure, navegador de proyectos, vista de archivos, editor tabulado, lista de tareas y más funciones entre otras. También presenta un sistema de construcción (build) rápido y soportes para la compilación. Del mismo modo contiene para la depuración breakpoints, funciones locales o argumentos y pilas de llamadas para ver valores de variables definidas por el usuario (<http://www.codeblocks.org/>). Éstas son algunas de las características para la creación del programa deseado.

Code::Blocks nos permite la edición de los códigos de los algoritmos seleccionados para resolver el problema que se nos plantea. En la bibliografía de este documento se recoge el manual de usuario de dicho programa. Una vez creados, a través del símbolo del sistema de Windows, se procede a operar con los archivos ejecutables mostrando la secuencia de trabajos correcta para el criterio de minimizar el tiempo de finalización de todos los trabajos, makespan.

Para la implementación de los códigos y a su vez de los algoritmos nos hemos basado y apoyado en la librería Schedule facilitada por el tutor de este trabajo. Dicha librería presenta una colección de clases y funciones escritas en el núcleo del lenguaje. Tiene declaradas numerosas funciones y definidos tipos usados por varias operaciones estándar. Las funciones que se encuentran en dicha librería y que se han utilizado en este estudio, las recojo en el Anexo 2, aunque a continuación se realiza una breve presentación de su contenido.

Destacar funciones para la manipulación de matrices y vectores como por ejemplo copiar una fila de una matriz de enteros en un vector, ordenar un vector o introducir un valor en las posiciones de una matriz. Funciones cuyos objetivos están relacionados con las operaciones propias de Scheduling, como por ejemplo para nuestro estudio, facilitándole la matriz de tiempos, el número de máquinas y la secuencia de trabajos te devuelve el makespan (C_{MAX}) así como también se podría calcular la tardanza máxima, el número de trabajos no entregados a tiempo, la máxima precocidad o la suma de los retrasos. Para cada función es necesaria una serie de valores y parámetros específicos. Del mismo modo, subrayar funciones pertenecientes a la escritura y salida, desde la carga de los tiempos de procesos desde un archivo, a la impresión de un vector por pantalla. Y por otro lado también se puede conseguir el valor máximo o mínimo de dos números mediante funciones que te lo calculan una vez le proporcionas dichas cifras.

Por último, para el análisis de los resultados me he apoyado en el software Excel debido a que permite crear tablas, y calcular y analizar datos. Este programa es una hoja donde se puede calcular cualquier tipo de operación matemática, por eso se conoce como hoja de cálculo, donde tiene diversas utilidades, la más básica es crear tablas de números y operar con ellos a través de fórmulas, de una manera rápida y sencilla puesto que calcula de forma automática los totales de los valores numéricos que se especifiquen, imprime tablas con diseños cuidados y crear gráficos simples.

4. Desarrollo de los algoritmos

4.1 Identificar algoritmos

Los programas de producción se elaboran con una misión concreta, alcanzar un objetivo. Para el estudio de este trabajo, el objetivo es minimizar el tiempo de finalización de todos los trabajos, makespan. Buscaremos satisfacer dicho objetivo mediante la experimentación de tres algoritmos: algoritmo Random, algoritmo de cálculo de los tiempos medios y algoritmo NEH.

A continuación identificaremos cada uno de ellos.

4.1.1 Algoritmo Random

El algoritmo Random presenta una posible solución a nuestro problema cuya técnica es la búsqueda aleatoria en la región factible, en la cual genera una solución aleatoria de la secuencia de trabajos a procesar. Nuestra región factible es el número de trabajos que hay que efectuar para llevar a cabo la producción.

Esta técnica no garantiza la obtención de la solución más adecuada ni en los mejores casos, y es tratada como la alternativa a no realizar ninguna acción referente a la planificación de los trabajos. Permite obtener unas soluciones aceptables en intervalos cortos, es decir, para un número muy pequeño de trabajos a procesar.

Este método puede formar parte de un algoritmo más complejo, desarrollando una secuencia de trabajos inicialmente aleatoria para ir evolucionando hasta encontrar la solución más favorable dentro de las limitaciones. Por ello es utilizada para problemas de gran complejidad y para los cuales no existen procedimientos que permitan resolverlos de manera satisfactoria. La búsqueda aleatoria pura se presenta como una alternativa ya que permite abordar casi cualquier tipo de problema de optimización, con la ventaja adicional que no es necesario hacer prácticamente ningún supuesto, uno de los principales problemas de los modelos de optimización tradicionales.

4.1.2 Algoritmo Tiempos medios

Este algoritmo radica en ordenar los trabajos por su tiempo medio de proceso. Teniendo en cuenta el tiempo en cada una de las máquinas, se efectúa la media de cada uno de los trabajos. Con esos valores se discute y se forma la secuencia de trabajos buscada por el algoritmo.

Es un algoritmo sencillo y fácil de implantar, no presenta ninguna idea compleja ni se apoya en reglas empíricas para llegar a la solución.

Con esta técnica se busca minimizar el tiempo de finalización de todos los trabajos calculando C_{MAX} . A pesar del conocimiento certero de los tiempos que tarda cada máquina en procesar cada trabajo, la secuencia solución del algoritmo es muy mejorable para un problema de m máquinas y n trabajos a procesar, pero localiza una secuencia solución aceptable y considerablemente correcta para el objetivo marcado, en ciertas dimensiones del problema.

4.1.3 Algoritmo NEH

La heurística NEH construye una secuencia agregando trabajos de una lista de trabajos (ordenados de mayor a menor tiempo total de proceso) de forma sucesiva y evaluando múltiples inserciones de éstos en las secuencias parcialmente construidas.

La heurística NEH fue creada por Nawaz, Enscore y Ham (1983) y también es utilizada en otros algoritmos, como los evolutivos, con el fin de generar al inicio un individuo muy bueno. Los pasos de la heurística son los siguientes:

1. Se ordenan los trabajos de mayor a menor teniendo en cuenta la suma de los tiempos de procesamiento en las máquinas.
2. Se toman los dos primeros trabajos de la lista y se organizan de forma tal que el makespan sea el mínimo como si solo existieran esas dos tareas.
3. De aquí en adelante y hasta la cantidad de tareas, se va insertando el trabajo que sigue en la lista en el lugar que minimice el makespan.
4. Al final de las iteraciones se obtendrá el orden de los trabajos, con las condiciones iniciales, el cual tendrá el makespan mínimo.

Con el objetivo de reducir el número de combinaciones a estudiar y obtener un algoritmo más eficiente y menos complejo, solo se analizan las posibles inserciones para cada iteración del método, a diferencia de analizar las $n!$ combinaciones que son posibles en un principio, siendo n el número de trabajos a procesar.

Para ilustrar el algoritmo de NEH se procede a realizar un ejemplo donde se mostrarán los pasos a seguir. La tabla 1 muestra el tiempo que se tarda en ejecutar los 3 trabajos en cada una de las 5 máquinas existentes.

	M_1	M_2	M_3	M_4	M_5
J_1	1	2	1	2	2
J_2	3	4	1	1	2
J_3	2	3	1	2	1

Tabla 1: Matriz de tiempos de ejecución de la tarea J_i en la máquina M_j

1) Se suma el tiempo que tarda cada tarea en pasar por todas las máquinas.

Para la matriz de tiempos dada en la tabla 1 se tiene:

i. $J_1 = 1 + 2 + 1 + 2 + 2 = 8$

ii. $J_2 = 3 + 4 + 1 + 1 + 2 = 11$

iii. $J_3 = 2 + 3 + 1 + 2 + 1 = 9$

2) Una vez obtenidos los tiempos se procede a organizar de forma decreciente los trabajos. Para este ejemplo el orden es: $J_2 - J_3 - J_1$.

3) Se seleccionan las tareas J_2 y J_3 de la lista ordenada y se procede a calcular el makespan teniendo en cuenta solo estas dos tareas.

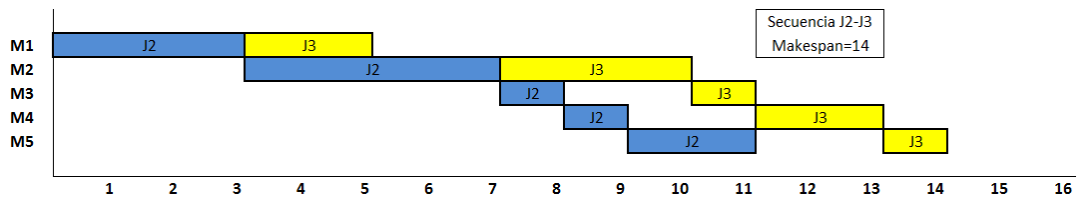


Figura 3: Cálculo del makespan para la secuencia $J_2 - J_3$. Fuente: Elaboración propia.

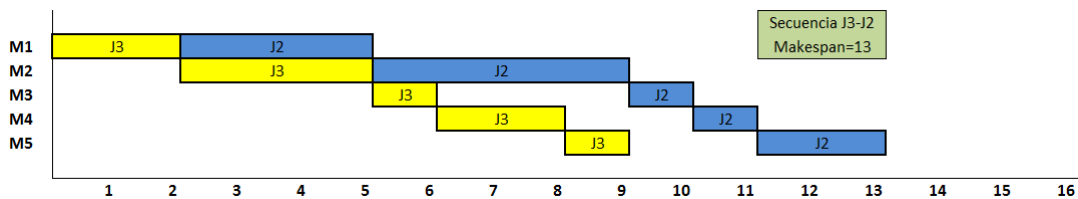


Figura 4: Cálculo del makespan para la secuencia $J_3 - J_2$. Fuente: Elaboración propia.

Como se puede ver en las figuras 3 y 4 el cálculo del makespan se debe realizar para cada una de las posibilidades de ordenamiento de los trabajos. La mejor secuencia de trabajos es $J_3 - J_2$ con un makespan = 13. Esto indica que necesariamente al final el trabajo J_3 deberá ser ejecutado antes que la tarea J_2 pero no necesariamente debe ser la inmediatamente anterior.

4) Ahora se selecciona la siguiente tarea en la lista, J_1 , y se inserta en cada una de las posibles posiciones. Las figuras 5, 6, 7 muestran el cálculo del makespan para cada uno de las posibles combinaciones de trabajos. En la figura 5 se muestra la mejor secuencia una vez incluida la tarea J_1 , con un makespan = 14.

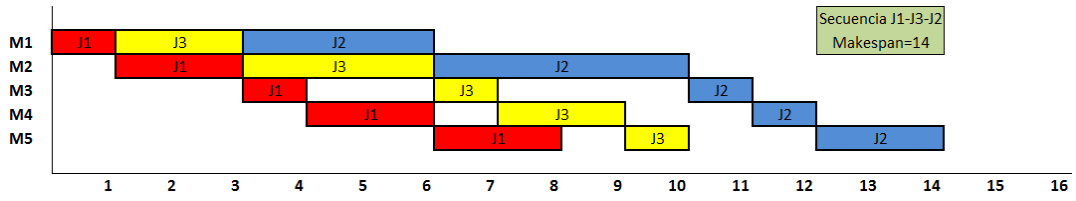


Figura 5: Cálculo del makespan para la secuencia $J_1 - J_3 - J_2$. Fuente: Elaboración propia.

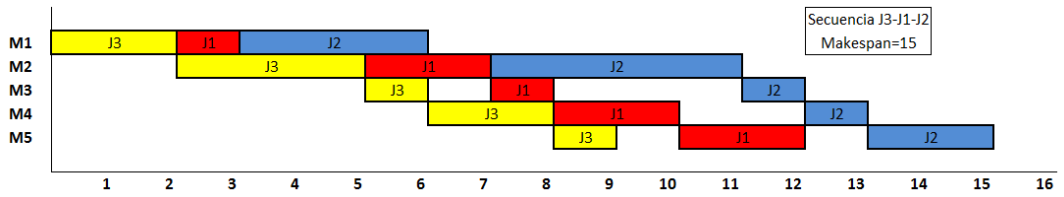


Figura 6: Cálculo del makespan para la secuencia $J_3 - J_1 - J_2$. Fuente: Elaboración propia.

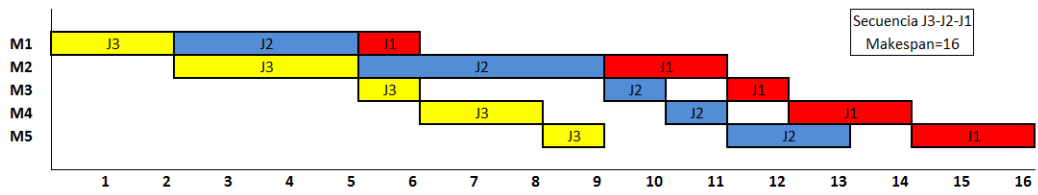


Figura 7: Cálculo del makespan para la secuencia $J_3 - J_2 - J_1$. Fuente: Elaboración propia.

- 5) Finalmente la secuencia $J_1 - J_3 - J_2$ es la secuencia solución de trabajos que nos proporciona el algoritmo NEH.

4.2 Pseudocódigos de los algoritmos

Decidir el código es la primera decisión que se debe tomar en el momento de implementar el algoritmo. El principal objetivo de la codificación reside en que los elementos característicos del problema se puedan representar de tal forma que resulte sencilla su implementación y comprensión. Los códigos, que son realizados por el autor de este trabajo, se encuentran en el Anexo 1 al final de este documento. A continuación se muestran los tres pseudocódigos para los respectivos algoritmos utilizados.

4.2.1 Pseudocódigo Random

En este apartado se describe el pseudocódigo perteneciente al algoritmo Random. El código detallado en C aparece en el anexo 1 de este documento, bajo el título de Código Algoritmo Random.

1. Declarar las variables.
2. Cargar los datos.
3. Generar secuencia aleatoria para los trabajos dados.
4. Calcular makespan.
5. Mostrar resultados.
6. Liberar memoria.

4.2.2 Pseudocódigo Tiempos medios

En esta unidad se representa el pseudocódigo referente al algoritmo basado en el cálculo de los tiempos medios. Su código detallado en C aparece en el anexo 1.

1. Declarar las variables.
2. Cargar los datos.
3. Calcular los tiempos medios de los trabajos.
4. Ordenar los trabajos de forma creciente respecto a sus tiempos medios.
5. Calcular makespan.
6. Mostrar los resultados.
7. Liberar memoria.

4.2.3 Pseudocódigo NEH

En esta sección se describe el pseudocódigo perteneciente al algoritmo basado en la heurística NEH. El código detallado en C aparece en el anexo 1 de este documento, bajo el título de Código Algoritmo NEH.

1. Declarar las variables.
2. Cargar los datos.
3. Calcular los tiempos de los trabajos en todas las máquinas.
4. Ordenar los trabajos de forma decreciente respecto a sus tiempos de procesos totales.
5. Seleccionar los dos primeros trabajos y programarlos para minimizar el makespan.
6. Para $k=3\dots n$ con $n = N^{\circ}$ de trabajos:
 - 6.1. Insertar el trabajo k en la posición en la que la secuencia parcial
 - 6.2. Entre las k posibles, escoger la que minimice el tiempo total de proceso o makespan.
7. Mostrar los resultados.
8. Liberar memoria.

5. Experimentación

5.1 Validación

En este apartado del trabajo se realizan una serie de experimentos o problemas donde utilizo los códigos de los algoritmos anteriormente descritos. Los valores escogidos para validar el trabajo son los siguientes:

- Porcentaje de ceros: 10-20-30-40.
- Número de trabajos: 50-150-250-400-500.
- Número de máquinas: 10-20-30-40-50.

El porcentaje de ceros representa el porcentaje de tiempos de proceso de un trabajo en una determinada máquina que van a ser ceros. El número de trabajos a procesar se va a encontrar en el rango manifestado anteriormente, donde se podrá analizar y obtener unos resultados fiables a la dimensión que pueda tener una empresa. Por último, el rango de máquinas que serán utilizadas por los trabajos se encuentra conforme a la extensión que se requiere del problema de la misma forma que sucede con el rango de trabajos.

Para alcanzar un resultado fiable del experimento se opta por efectuar treinta instancias para cada número de trabajo y número de máquina, ya que consideramos que es un valor lo suficientemente correcto para elaborar un análisis adecuado posteriormente. Un ejemplo de los resultados que se obtienen para uno de los experimentos se recoge en el Anexo 3 de este documento.

Los datos de dichos experimentos son obtenidos a través de un código que atendiendo a los rangos comentados anteriormente me genera un conjunto de archivos de texto donde se encuentra la matriz de tiempos de proceso de cada trabajo en cada máquina, así como el número de máquinas utilizadas y el número de trabajos demandados, declarados en las dos primeras filas respectivamente.

Es decir, el programa generará, dentro del porcentaje de ceros que presentarán los tiempos de procesamientos, 30 archivos de texto para el caso de 50 trabajos y 10

máquinas, 30 archivos de texto para el caso de 50 trabajos y 20 máquinas, y así sucesivamente hasta alcanzar los 500 trabajos y 50 máquinas. Se realizará esta operación para los casos en los que el porcentaje de ceros en los tiempos de proceso sea del 10%, 20% 30% y 40%.

Una vez obtenidos todos los datos en los que nos basaremos para hacer los experimentos, se realiza la ejecución de cada algoritmo identificado en este estudio para cada archivo de texto. Para ello utilizamos el programa símbolo del sistema (CMD) del ordenador. De las treinta instancias se realiza la media que corresponderá al resultado de ese experimento, y por otro lado, se obtiene la desviación típica. Para estas operaciones nos ayudamos de la herramienta Excel.

En el momento en el que se tienen recogidos todos los experimentos en sus respectivas hojas de Excel, nos disponemos a analizar los resultados en su conjunto. Para ello se realiza una nueva hoja Excel donde se manifiestan todos los resultados agrupándolos según número de trabajo y número de máquina. En ella queda recogido el valor promedio de las treinta instancias de cada porcentaje de ceros para cada algoritmo, la desviación típica hallada y el porcentaje de mejora de los algoritmos Random y tiempo medio respecto a NEH. A continuación se muestran todos los resultados obtenidos y recogidos en las tablas siguientes:

VALORES PROMEDIO j50 m10			
%CEROS	RANDOM	T. MEDIO	NEH
0,1	3663,600	3587,333	3025,367
0,2	3360,333	3326,533	2707,767
0,3	3110,467	3091,467	2437,200
0,4	2836,600	2805,667	2171,767

Tabla 2: Valores promedio para el experimento de 50 trabajos y 10 máquinas.

VALORES DESVIACIÓN j50 m10			
%CEROS	RANDOM	T. MEDIO	NEH
0,1	124,838	165,687	102,925
0,2	136,612	162,249	99,564
0,3	142,287	173,069	95,583
0,4	130,296	183,817	96,579

Tabla 3: Valores desviación para el experimento de 50 trabajos y 10 máquinas.

PORCENTAJES DE MEJORA		
%CEROS	RANDOM	T. MEDIO
0,1	21,096	18,575
0,2	24,100	22,852
0,3	27,625	26,845
0,4	30,613	29,188

Tabla 4: Porcentaje de mejora respecto a NEH para el experimento de 50 trabajos y 10 máquinas.

VALORES PROMEDIO j50 m20			
%CEROS	RANDOM	T. MEDIO	NEH
0,1	4555,000	4576,667	3760,900
0,2	4277,600	4298,200	3429,867
0,3	4009,567	3992,533	3127,933
0,4	3712,367	3683,333	2778,833

Tabla 5: Valores promedio para el experimento de 50 trabajos y 20 máquinas.

VALORES DESVIACIÓN j50 m20			
%CEROS	RANDOM	T. MEDIO	NEH
0,1	104,390	116,142	96,098
0,2	146,973	178,783	102,928
0,3	161,742	154,436	89,689
0,4	145,423	146,147	86,133

Tabla 6: Valores desviación para el experimento de 50 trabajos y 20 máquinas.

PORCENTAJES DE MEJORA		
%CEROS	RANDOM	T. MEDIO
0,1	21,115	21,691
0,2	24,716	25,317
0,3	28,186	27,641
0,4	33,594	32,550

Tabla 7: Porcentaje de mejora respecto a NEH para el experimento de 50 trabajos y 20 máquinas.

VALORES PROMEDIO j50 m30			
%CEROS	RANDOM	T. MEDIO	NEH
0,1	5388,200	5336,400	4554,800
0,2	5074,400	5069,967	4188,633
0,3	4776,833	4773,033	3818,233
0,4	4437,167	4398,933	3394,633

Tabla 8: Valores promedio para el experimento de 50 trabajos y 30 máquinas.

VALORES DESVIACIÓN j50 m30			
%CEROS	RANDOM	T. MEDIO	NEH
0,1	159,057	147,710	90,302
0,2	149,687	125,823	91,261
0,3	163,331	188,705	94,246
0,4	193,344	172,797	91,041

Tabla 9: Valores desviación para el experimento de 50 trabajos y 30 máquinas.

PORCENTAJES DE MEJORA		
%CEROS	RANDOM	T. MEDIO
0,1	18,297	17,160
0,2	21,147	21,041
0,3	25,106	25,006
0,4	30,711	29,585

Tabla 10: Porcentaje de mejora respecto a NEH para el experimento de 50 trabajos y 30 máquinas.

VALORES PROMEDIO j50 m40			
%CEROS	RANDOM	T. MEDIO	NEH
0,1	6126,867	6127,067	5210,933
0,2	5843,167	5800,300	4835,367
0,3	5454,800	5444,900	4450,200
0,4	5021,833	5000,033	3965,100

Tabla 11: Valores promedio para el experimento de 50 trabajos y 40 máquinas.

VALORES DESVIACIÓN j50 m40			
%CEROS	RANDOM	T. MEDIO	NEH
0,1	129,373	126,351	99,720
0,2	186,673	152,755	121,555
0,3	190,374	135,033	124,075
0,4	174,111	136,884	107,218

Tabla 12: Valores desviación para el experimento de 50 trabajos y 40 máquinas.

PORCENTAJES DE MEJORA		
%CEROS	RANDOM	T. MEDIO
0,1	17,577	17,581
0,2	20,842	19,956
0,3	22,574	22,352
0,4	26,651	26,101

Tabla 13: Porcentaje de mejora respecto a NEH para el experimento de 50 trabajos y 40 máquinas.

VALORES PROMEDIO j50 m50			
%CEROS	RANDOM	T. MEDIO	NEH
0,1	6845,233	6832,133	5934,833
0,2	6497,467	6477,933	5529,900
0,3	6114,900	6092,800	5028,567
0,4	5626,067	5621,233	4547,833

Tabla 14: Valores promedio para el experimento de 50 trabajos y 50 máquinas.

VALORES DESVIACIÓN j50 m50			
%CEROS	RANDOM	T. MEDIO	NEH
0,1	159,930	157,052	113,233
0,2	206,443	158,115	112,181
0,3	194,528	173,640	109,968
0,4	170,770	138,824	125,338

Tabla 15: Valores desviación para el experimento de 50 trabajos y 50 máquinas.

PORCENTAJES DE MEJORA		
%CEROS	RANDOM	T. MEDIO
0,1	15,340	15,119
0,2	17,497	17,144
0,3	21,603	21,164
0,4	23,709	23,602

Tabla 16: Porcentaje de mejora respecto a NEH para el experimento de 50 trabajos y 50 máquinas.

VALORES PROMEDIO j150 m10			
%CEROS	RANDOM	T. MEDIO	NEH
0,1	8850,400	8940,000	7829,167
0,2	8149,667	8180,533	7040,967
0,3	7495,300	7421,900	6252,200
0,4	6654,667	6670,333	5480,367

Tabla 17: Valores promedio para el experimento de 150 trabajos y 10 máquinas.

VALORES DESVIACIÓN j150 m10			
%CEROS	RANDOM	T. MEDIO	NEH
0,1	226,037	276,121	272,870
0,2	218,548	217,303	207,980
0,3	274,551	219,332	206,031
0,4	255,334	253,685	240,186

Tabla 18: Valores desviación para el experimento de 150 trabajos y 10 máquinas.

PORCENTAJES DE MEJORA		
%CEROS	RANDOM	T. MEDIO
0,1	13,044	14,188
0,2	15,746	16,185
0,3	19,883	18,709
0,4	21,427	21,713

Tabla 19: Porcentaje de mejora respecto a NEH para el experimento de 150 trabajos y 10 máquinas.

VALORES PROMEDIO j150 m20			
%CEROS	RANDOM	T. MEDIO	NEH
0,1	10288,600	10244,000	8693,267
0,2	9572,867	9582,333	7871,267
0,3	8822,433	8824,567	7102,733
0,4	8074,367	7937,867	6210,400

Tabla 20: Valores promedio para el experimento de 150 trabajos y 20 máquinas.

VALORES DESVIACIÓN j150 m20			
%CEROS	RANDOM	T. MEDIO	NEH
0,1	230,586	213,281	137,601
0,2	253,595	195,536	111,988
0,3	207,211	240,615	150,464
0,4	231,377	217,608	164,444

Tabla 21: Valores desviación para el experimento de 150 trabajos y 20 máquinas.

PORCENTAJES DE MEJORA		
%CEROS	RANDOM	T. MEDIO
0,1	18,351	17,838
0,2	21,618	21,738
0,3	24,212	24,242
0,4	30,014	27,816

Tabla 22: Porcentaje de mejora respecto a NEH para el experimento de 150 trabajos y 20 máquinas.

VALORES PROMEDIO j150 m30			
%CEROS	RANDOM	T. MEDIO	NEH
0,1	11412,167	11285,633	9572,867
0,2	10652,533	10559,700	8713,933
0,3	9945,700	9818,433	7893,033
0,4	8999,500	8919,200	6925,633

Tabla 23: Valores promedio para el experimento de 150 trabajos y 30 máquinas.

VALORES DESVIACIÓN j150 m30			
%CEROS	RANDOM	T. MEDIO	NEH
0,1	235,855	176,741	134,410
0,2	253,777	256,876	130,735
0,3	191,697	211,908	146,639
0,4	258,346	217,724	91,888

Tabla 24: Valores desviación para el experimento de 150 trabajos y 30 máquinas.

PORCENTAJES DE MEJORA		
%CEROS	RANDOM	T. MEDIO
0,1	19,214	17,892
0,2	22,247	21,182
0,3	26,006	24,394
0,4	29,945	28,785

Tabla 25: Porcentaje de mejora respecto a NEH para el experimento de 150 trabajos y 30 máquinas.

VALORES PROMEDIO j150 m40			
%CEROS	RANDOM	T. MEDIO	NEH
0,1	12405,433	12391,433	10511,400
0,2	11687,200	11629,733	9644,033
0,3	10838,733	10869,333	8725,767
0,4	9934,000	9827,300	7736,433

Tabla 26: Valores promedio para el experimento de 150 trabajos y 40 máquinas.

VALORES DESVIACIÓN j150 m40			
%CEROS	RANDOM	T. MEDIO	NEH
0,1	226,420	214,692	112,886
0,2	164,189	208,605	102,095
0,3	216,543	256,171	114,322
0,4	236,560	188,398	106,820

Tabla 27: Valores desviación para el experimento de 150 trabajos y 40 máquinas.

PORCENTAJES DE MEJORA		
%CEROS	RANDOM	T. MEDIO
0,1	18,019	17,886
0,2	21,186	20,590
0,3	24,215	24,566
0,4	28,405	27,026

Tabla 28: Porcentaje de mejora respecto a NEH para el experimento de 150 trabajos y 40 máquinas.

VALORES PROMEDIO j150 m50			
%CEROS	RANDOM	T. MEDIO	NEH
0,1	13227,600	13244,133	11427,167
0,2	12583,733	12595,767	10530,700
0,3	11687,967	11636,000	9565,067
0,4	10729,267	10818,700	8549,767

Tabla 29: Valores promedio para el experimento de 150 trabajos y 50 máquinas.

VALORES DESVIACIÓN j150 m50			
%CEROS	RANDOM	T. MEDIO	NEH
0,1	173,187	184,449	96,616
0,2	203,125	205,769	80,803
0,3	218,232	220,811	114,692
0,4	204,635	252,299	108,332

Tabla 30: Valores desviación para el experimento de 150 trabajos y 50 máquinas.

PORCENTAJES DE MEJORA		
%CEROS	RANDOM	T. MEDIO
0,1	15,756	15,900
0,2	19,496	19,610
0,3	22,194	21,651
0,4	25,492	26,538

Tabla 31: Porcentaje de mejora respecto a NEH para el experimento de 150 trabajos y 50 máquinas.

VALORES PROMEDIO j250 m10			
%CEROS	RANDOM	T. MEDIO	NEH
0,1	13932,100	13810,467	12561,733
0,2	12803,033	12810,867	11326,133
0,3	11550,600	11576,900	10034,033
0,4	10304,133	10261,600	8747,267

Tabla 32: Valores promedio para el experimento de 250 trabajos y 10 máquinas.

VALORES DESVIACIÓN j250 m10			
%CEROS	RANDOM	T. MEDIO	NEH
0,1	255,574	233,813	344,918
0,2	282,632	307,102	311,515
0,3	294,186	471,830	239,048
0,4	370,117	338,720	323,082

Tabla 33: Valores desviación para el experimento de 250 trabajos y 10 máquinas.

PORCENTAJES DE MEJORA		
%CEROS	RANDOM	T. MEDIO
0,1	10,909	9,941
0,2	13,040	13,109
0,3	15,114	15,376
0,4	17,798	17,312

Tabla 34: Porcentaje de mejora respecto a NEH para el experimento de 250 trabajos y 10 máquinas.

VALORES PROMEDIO j250 m20			
%CEROS	RANDOM	T. MEDIO	NEH
0,1	15715,467	15620,333	13470,567
0,2	14518,900	14491,167	12193,933
0,3	13300,900	13199,300	10925,733
0,4	11975,867	11891,233	9547,100

Tabla 35: Valores promedio para el experimento de 250 trabajos y 20 máquinas.

VALORES DESVIACIÓN j250 m20			
%CEROS	RANDOM	T. MEDIO	NEH
0,1	319,153	299,344	191,879
0,2	293,089	259,053	208,632
0,3	245,203	304,566	204,134
0,4	396,357	279,734	207,302

Tabla 36: Valores desviación para el experimento de 250 trabajos y 20 máquinas.

PORCENTAJES DE MEJORA		
%CEROS	RANDOM	T. MEDIO
0,1	16,665	15,959
0,2	19,067	18,839
0,3	21,739	20,809
0,4	25,440	24,553

Tabla 37: Porcentaje de mejora respecto a NEH para el experimento de 250 trabajos y 20 máquinas.

VALORES PROMEDIO j250 m30			
%CEROS	RANDOM	T. MEDIO	NEH
0,1	16992,600	17014,567	14495,467
0,2	15772,367	15793,767	13108,633
0,3	14544,200	14563,867	11757,067
0,4	13260,733	13160,567	10334,033

Tabla 38: Valores promedio para el experimento de 250 trabajos y 30 máquinas.

VALORES DESVIACIÓN j250 m30			
%CEROS	RANDOM	T. MEDIO	NEH
0,1	256,503	277,571	151,953
0,2	282,177	302,628	106,508
0,3	239,709	353,057	162,584
0,4	318,562	288,220	162,367

Tabla 39: Valores desviación para el experimento de 250 trabajos y 30 máquinas.

PORCENTAJES DE MEJORA		
%CEROS	RANDOM	T. MEDIO
0,1	17,227	17,379
0,2	20,320	20,484
0,3	23,706	23,873
0,4	28,321	27,352

Tabla 40: Porcentaje de mejora respecto a NEH para el experimento de 250 trabajos y 30 máquinas.

VALORES PROMEDIO j250 m40			
%CEROS	RANDOM	T. MEDIO	NEH
0,1	18116,467	18069,467	15485,500
0,2	16921,633	16922,600	14151,467
0,3	15729,533	15694,733	12720,933
0,4	14251,900	14320,733	11190,267

Tabla 41: Valores promedio para el experimento de 250 trabajos y 40 máquinas.

VALORES DESVIACIÓN j250 m40			
%CEROS	RANDOM	T. MEDIO	NEH
0,1	307,822	211,636	104,459
0,2	255,973	319,482	151,067
0,3	357,108	240,590	118,985
0,4	243,575	306,555	140,583

Tabla 42: Valores desviación para el experimento de 250 trabajos y 40 máquinas.

PORCENTAJES DE MEJORA		
%CEROS	RANDOM	T. MEDIO
0,1	16,990	16,686
0,2	19,575	19,582
0,3	23,651	23,377
0,4	27,360	27,975

Tabla 43: Porcentaje de mejora respecto a NEH para el experimento de 250 trabajos y 40 máquinas.

VALORES PROMEDIO j250 m50			
%CEROS	RANDOM	T. MEDIO	NEH
0,1	19082,500	19190,700	16470,900
0,2	18044,500	18010,933	15118,333
0,3	16728,167	16716,167	13674,100
0,4	15230,067	15336,633	12125,133

Tabla 44: Valores promedio para el experimento de 250 trabajos y 50 máquinas.

VALORES DESVIACIÓN j250 m50			
%CEROS	RANDOM	T. MEDIO	NEH
0,1	219,434	204,859	117,433
0,2	231,152	296,748	141,998
0,3	269,650	217,387	141,408
0,4	332,462	248,406	123,258

Tabla 45: Valores desviación para el experimento de 250 trabajos y 50 máquinas.

PORCENTAJES DE MEJORA		
%CEROS	RANDOM	T. MEDIO
0,1	15,856	16,513
0,2	19,355	19,133
0,3	22,335	22,247
0,4	25,607	26,486

Tabla 46: Porcentaje de mejora respecto a NEH para el experimento de 250 trabajos y 50 máquinas.

VALORES PROMEDIO j400 m10			
%CEROS	RANDOM	T. MEDIO	NEH
0,1	21433,900	21277,467	19753,967
0,2	19518,067	19514,933	17687,933
0,3	17533,100	17506,000	15725,733
0,4	15443,567	15570,567	13791,267

Tabla 47: Valores promedio para el experimento de 400 trabajos y 10 máquinas.

VALORES DESVIACIÓN j400 m10			
%CEROS	RANDOM	T. MEDIO	NEH
0,1	316,970	338,476	510,587
0,2	340,079	410,917	370,445
0,3	313,506	395,746	440,261
0,4	364,992	425,969	379,351

Tabla 48: Valores desviación para el experimento de 400 trabajos y 10 máquinas.

PORCENTAJES DE MEJORA		
%CEROS	RANDOM	T. MEDIO
0,1	8,504	7,712
0,2	10,347	10,329
0,3	11,493	11,321
0,4	11,981	12,902

Tabla 49: Porcentaje de mejora respecto a NEH para el experimento de 400 trabajos y 10 máquinas.

VALORES PROMEDIO j400 m20			
%CEROS	RANDOM	T. MEDIO	NEH
0,1	23466,567	23410,333	20598,700
0,2	21529,533	21548,633	18643,067
0,3	19580,700	19459,900	16527,000
0,4	17541,900	17498,800	14414,067

Tabla 50: Valores promedio para el experimento de 400 trabajos y 20 máquinas.

VALORES DESVIACIÓN j400 m20			
%CEROS	RANDOM	T. MEDIO	NEH
0,1	348,842	341,396	179,940
0,2	324,632	354,380	243,584
0,3	306,764	415,745	284,325
0,4	348,973	395,327	274,615

Tabla 51: Valores desviación para el experimento de 400 trabajos y 20 máquinas.

PORCENTAJES DE MEJORA		
%CEROS	RANDOM	T. MEDIO
0,1	13,923	13,650
0,2	15,483	15,585
0,3	18,477	17,746
0,4	21,700	21,401

Tabla 52: Porcentaje de mejora respecto a NEH para el experimento de 400 trabajos y 20 máquinas.

VALORES PROMEDIO j400 m30			
%CEROS	RANDOM	T. MEDIO	NEH
0,1	24915,467	24911,000	21601,700
0,2	23157,367	23068,267	19525,900
0,3	21141,567	21106,667	17404,867
0,4	18992,633	19013,633	15246,167

Tabla 53: Valores promedio para el experimento de 400 trabajos y 30 máquinas.

VALORES DESVIACIÓN j400 m30			
%CEROS	RANDOM	T. MEDIO	NEH
0,1	356,628	292,945	192,495
0,2	313,914	303,506	172,356
0,3	348,271	442,829	201,178
0,4	334,637	408,760	167,005

Tabla 54: Valores desviación para el experimento de 400 trabajos y 30 máquinas.

PORCENTAJES DE MEJORA		
%CEROS	RANDOM	T. MEDIO
0,1	15,340	15,320
0,2	18,598	18,142
0,3	21,469	21,269
0,4	24,573	24,711

Tabla 55: Porcentaje de mejora respecto a NEH para el experimento de 400 trabajos y 30 máquinas.

VALORES PROMEDIO j400 m40			
%CEROS	RANDOM	T. MEDIO	NEH
0,1	26354,033	26298,633	22717,233
0,2	24552,300	24528,200	20652,433
0,3	22428,900	22509,833	18470,900
0,4	20387,433	20363,200	16234,700

Tabla 56: Valores promedio para el experimento de 400 trabajos y 40 máquinas.

VALORES DESVIACIÓN j400 m40			
%CEROS	RANDOM	T. MEDIO	NEH
0,1	329,315	256,326	157,959
0,2	282,818	262,118	146,632
0,3	449,717	359,629	173,397
0,4	356,323	305,694	195,977

Tabla 57: Valores desviación para el experimento de 400 trabajos y 40 máquinas.

PORCENTAJES DE MEJORA		
%CEROS	RANDOM	T. MEDIO
0,1	16,009	15,765
0,2	18,883	18,767
0,3	21,428	21,866
0,4	25,579	25,430

Tabla 58: Porcentaje de mejora respecto a NEH para el experimento de 400 trabajos y 40 máquinas.

VALORES PROMEDIO j400 m50			
%CEROS	RANDOM	T. MEDIO	NEH
0,1	27571,733	27531,833	23858,100
0,2	25659,967	25700,467	21674,567
0,3	23766,033	23697,333	19511,800
0,4	21593,900	21573,567	17190,033

Tabla 59: Valores promedio para el experimento de 400 trabajos y 50 máquinas.

VALORES DESVIACIÓN j400 m50			
%CEROS	RANDOM	T. MEDIO	NEH
0,1	278,033	378,150	149,634
0,2	325,072	289,594	116,568
0,3	292,504	352,811	158,723
0,4	355,820	372,536	106,269

Tabla 60: Valores desviación para el experimento de 400 trabajos y 50 máquinas.

PORCENTAJES DE MEJORA		
%CEROS	RANDOM	T. MEDIO
0,1	15,566	15,398
0,2	18,387	18,574
0,3	21,803	21,451
0,4	25,619	25,500

Tabla 61: Porcentaje de mejora respecto a NEH para el experimento de 400 trabajos y 50 máquinas.

VALORES PROMEDIO j500 m10			
%CEROS	RANDOM	T. MEDIO	NEH
0,1	26186,700	26228,067	24575,400
0,2	23913,833	23914,333	21939,100
0,3	21486,100	21347,367	19737,267
0,4	18870,367	18842,067	16981,567

Tabla 62: Valores promedio para el experimento de 500 trabajos y 10 máquinas.

VALORES DESVIACIÓN j500 m10			
%CEROS	RANDOM	T. MEDIO	NEH
0,1	388,544	479,170	518,805
0,2	431,968	443,591	443,778
0,3	467,883	570,682	663,545
0,4	397,923	476,501	584,469

Tabla 63: Valores desviación para el experimento de 500 trabajos y 10 máquinas.

PORCENTAJES DE MEJORA		
%CEROS	RANDOM	T. MEDIO
0,1	6,557	6,725
0,2	9,001	9,003
0,3	8,861	8,158
0,4	11,123	10,956

Tabla 64: Porcentaje de mejora respecto a NEH para el experimento de 500 trabajos y 10 máquinas.

VALORES PROMEDIO j500 m20			
%CEROS	RANDOM	T. MEDIO	NEH
0,1	28342,167	28483,200	25453,867
0,2	25987,033	26193,067	22930,833
0,3	23664,467	23560,167	20339,300
0,4	21147,667	21017,533	17564,933

Tabla 65: Valores promedio para el experimento de 500 trabajos y 20 máquinas.

VALORES DESVIACIÓN j500 m20			
%CEROS	RANDOM	T. MEDIO	NEH
0,1	393,882	333,191	452,937
0,2	695,571	313,482	342,041
0,3	405,049	289,378	397,293
0,4	436,387	408,039	256,132

Tabla 66: Valores desviación para el experimento de 500 trabajos y 20 máquinas.

PORCENTAJES DE MEJORA		
%CEROS	RANDOM	T. MEDIO
0,1	11,347	11,901
0,2	13,328	14,226
0,3	16,348	15,836
0,4	20,397	19,656

Tabla 67: Porcentaje de mejora respecto a NEH para el experimento de 500 trabajos y 20 máquinas.

VALORES PROMEDIO j500 m30			
%CEROS	RANDOM	T. MEDIO	NEH
0,1	30082,800	30174,833	26395,367
0,2	27840,900	27783,767	23778,233
0,3	25360,567	25341,600	21212,233
0,4	22683,367	22685,367	18493,700

Tabla 68: Valores promedio para el experimento de 500 trabajos y 30 máquinas.

VALORES DESVIACIÓN j500 m30			
%CEROS	RANDOM	T. MEDIO	NEH
0,1	332,535	445,930	247,362
0,2	400,566	367,106	237,800
0,3	390,307	422,904	337,807
0,4	361,649	492,519	209,015

Tabla 69: Valores desviación para el experimento de 500 trabajos y 30 máquinas.

PORCENTAJES DE MEJORA		
%CEROS	RANDOM	T. MEDIO
0,1	13,970	14,319
0,2	17,086	16,845
0,3	19,556	19,467
0,4	22,655	22,665

Tabla 70: Porcentaje de mejora respecto a NEH para el experimento de 500 trabajos y 30 máquinas.

VALORES PROMEDIO j500 m40			
%CEROS	RANDOM	T. MEDIO	NEH
0,1	31567,133	31615,367	27449,000
0,2	29313,200	29287,933	24863,967
0,3	26855,167	26827,567	22210,167
0,4	24173,367	24182,333	19389,967

Tabla 71: Valores promedio para el experimento de 500 trabajos y 40 máquinas.

VALORES DESVIACIÓN j500 m40			
%CEROS	RANDOM	T. MEDIO	NEH
0,1	347,906	374,813	192,575
0,2	313,028	332,507	214,415
0,3	442,279	383,203	170,030
0,4	404,444	449,613	185,845

Tabla 72: Valores desviación para el experimento de 500 trabajos y 40 máquinas.

PORCENTAJES DE MEJORA		
%CEROS	RANDOM	T. MEDIO
0,1	15,003	15,179
0,2	17,894	17,793
0,3	20,914	20,790
0,4	24,669	24,716

Tabla 73: Porcentaje de mejora respecto a NEH para el experimento de 500 trabajos y 40 máquinas.

VALORES PROMEDIO j500 m50			
%CEROS	RANDOM	T. MEDIO	NEH
0,1	32937,400	32951,633	28626,800
0,2	30614,500	30628,800	25999,067
0,3	28152,500	28114,600	23316,400
0,4	25473,533	25458,533	20522,300

Tabla 74: Valores promedio para el experimento de 500 trabajos y 50 máquinas.

VALORES DESVIACIÓN j500 m50			
%CEROS	RANDOM	T. MEDIO	NEH
0,1	305,744	268,276	132,552
0,2	319,382	360,355	124,841
0,3	291,657	322,253	150,705
0,4	331,495	429,287	127,425

Tabla 75: Valores desviación para el experimento de 500 trabajos y 50 máquinas.

PORCENTAJES DE MEJORA		
%CEROS	RANDOM	T. MEDIO
0,1	15,058	15,108
0,2	17,752	17,807
0,3	20,741	20,579
0,4	24,126	24,053

Tabla 76: Porcentaje de mejora respecto a NEH para el experimento de 500 trabajos y 50 máquinas.

Del mismo modo se crean cuatro archivos de Excel, uno para cada porcentaje de ceros, para realizar el análisis de los resultados de un mismo tipo de tiempos de procesamientos, es decir, se muestran a través de dos gráficas la evolución de las medias de los experimentos y el promedio de los algoritmos utilizados para todo el conjunto de experimentos que abarca cada número de trabajo con cada número de máquinas haciendo un total de 25 experimentos.

Los datos obtenidos de makespan para los experimentos que abarcan desde que el 10% de los tiempos de procesamiento sean cero hasta el 40%, se representan a continuación.

EXPERIMENTO	RANDOM	T. MEDIO	NEH
j50 m10 - 0,1	3663,600	3587,333	3025,367
j50 m20- 0,1	4555,000	4576,667	3760,900
j50 m30- 0,1	5388,200	5336,400	4554,800
j50 m40- 0,1	6126,867	6127,067	5210,933
j50 m50- 0,1	6845,233	6832,133	5934,833
j150 m10- 0,1	8850,400	8940,000	7829,167
j150 m20- 0,1	10288,600	10244,000	8693,267
j150 m30- 0,1	11412,167	11285,633	9572,867
j150 m40- 0,1	12405,433	12391,433	10511,400
j150 m50- 0,1	13227,600	13244,133	11427,167
j250 m10- 0,1	13932,100	13810,467	12561,733
j250 m20- 0,1	15715,467	15620,333	13470,567
j250 m30- 0,1	16992,600	17014,567	14495,467
j250 m40- 0,1	18116,467	18069,467	15485,500
j250 m50- 0,1	19082,500	19190,700	16470,900
j400 m10- 0,1	21433,900	21277,467	19753,967
j400 m20- 0,1	23466,567	23410,333	20598,700
j400 m30- 0,1	24915,467	24911,000	21601,700
j400 m40- 0,1	26354,033	26298,633	22717,233
j400 m50- 0,1	27571,733	27531,833	23858,100
j500 m10- 0,1	26186,700	26228,067	24575,400
j500 m20- 0,1	28342,167	28483,200	25453,867
j500 m30- 0,1	30082,800	30174,833	26395,367
j500 m40- 0,1	31567,133	31615,367	27449,000
j500 m50- 0,1	32937,400	32951,633	28626,800
MAKESPAN	17578,405	17566,108	15361,400

Tabla 77: Valores medios makespan de las 30 instancias de cada experimento para cada algoritmo utilizado, en el tipo en el que el 10% de los tiempos de procesamientos son cero.

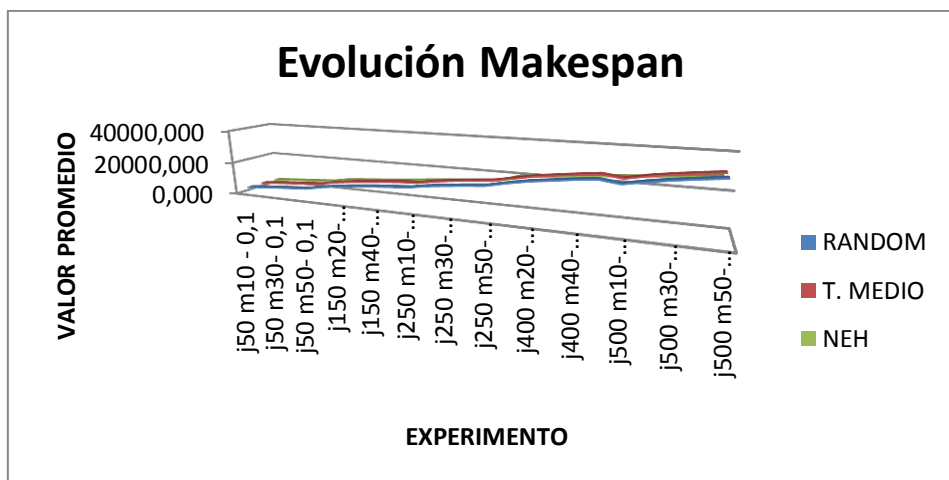


Figura 8: Evolución del makespan en los experimentos para cada algoritmo, en el tipo en el que el 10% de los tiempos de procesamientos son cero.

EXPERIMENTO	RANDOM	T. MEDIO	NEH
j50 m10 - 0,2	3360,333	3326,533	2707,767
j50 m20- 0,2	4277,600	4298,200	3429,867
j50 m30- 0,2	5074,400	5069,967	4188,633
j50 m40- 0,2	5843,167	5800,300	4835,367
j50 m50- 0,2	6497,467	6477,933	5529,900
j150 m10- 0,2	8149,667	8180,533	7040,967
j150 m20- 0,2	9572,867	9582,333	7871,267
j150 m30- 0,2	10652,533	10559,700	8713,933
j150 m40- 0,1	11687,200	11629,733	9644,033
j150 m50- 0,2	12583,733	12595,767	10530,700
j250 m10- 0,2	12803,033	12810,867	11326,133
j250 m20- 0,2	14518,900	14491,167	12193,933
j250 m30- 0,2	15772,367	15793,767	13108,633
j250 m40- 0,2	16921,633	16922,600	14151,467
j250 m50- 0,2	18044,500	18010,933	15118,333
j400 m10- 0,2	19518,067	19514,933	17687,933
j400 m20- 0,2	21529,533	21548,633	18643,067
j400 m30- 0,2	23157,367	23068,267	19525,900
j400 m40- 0,2	24552,300	24528,200	20652,433
j400 m50- 0,2	25659,967	25700,467	21674,567
j500 m10- 0,2	23913,833	23914,333	21939,100
j500 m20- 0,2	25987,033	26193,067	22930,833
j500 m30- 0,2	27840,900	27783,767	23778,233
j500 m40- 0,2	29313,200	29287,933	24863,967
j500 m50- 0,2	30614,500	30628,800	25999,067
PROMEDIO	16313,844	16308,749	13923,441

Tabla 78: Valores medios makespan de las 30 instancias de cada experimento para cada algoritmo utilizado, en el tipo en el que el 20% de los tiempos de procesamientos son cero.

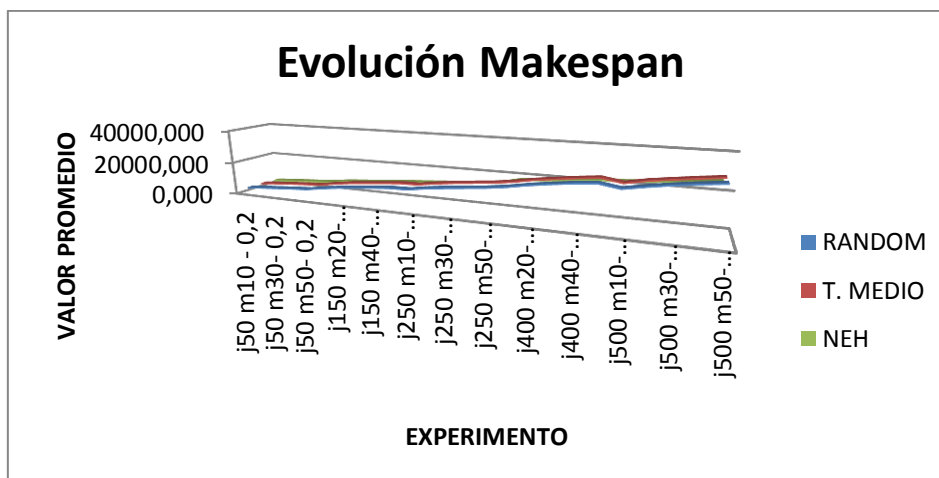


Figura 9: Evolución del makespan en los experimentos para cada algoritmo, en el tipo en el que el 20% de los tiempos de procesamientos son cero.

EXPERIMENTO	RANDOM	T. MEDIO	NEH
j50 m10 - 0,3	3110,467	3091,467	2437,200
j50 m20- 0,3	4009,567	3992,533	3127,933
j50 m30- 0,3	4776,833	4773,033	3818,233
j50 m40- 0,3	5454,800	5444,900	4450,200
j50 m50- 0,3	6114,900	6092,800	5028,567
j150 m10- 0,3	7495,300	7421,900	6252,200
j150 m20- 0,3	8822,433	8824,567	7102,733
j150 m30- 0,3	9945,700	9818,433	7893,033
j150 m40- 0,3	10838,733	10869,333	8725,767
j150 m50- 0,3	11687,967	11636,000	9565,067
j250 m10- 0,3	11550,600	11576,900	10034,033
j250 m20- 0,3	13300,900	13199,300	10925,733
j250 m30- 0,3	14544,200	14563,867	11757,067
j250 m40- 0,3	15729,533	15694,733	12720,933
j250 m50- 0,3	16728,167	16716,167	13674,100
j400 m10- 0,3	17533,100	17506,000	15725,733
j400 m20- 0,3	19580,700	19459,900	16527,000
j400 m30- 0,3	21141,567	21106,667	17404,867
j400 m40- 0,3	22428,900	22509,833	18470,900
j400 m50- 0,3	23766,033	23697,333	19511,800
j500 m10- 0,3	21486,100	21347,367	19737,267
j500 m20- 0,3	23664,467	23560,167	20339,300
j500 m30- 0,3	25360,567	25341,600	21212,233
j500 m40- 0,3	26855,167	26827,567	22210,167
j500 m50- 0,3	28152,500	28114,600	23316,400
PROMEDIO	14963,168	14927,479	12478,739

Tabla 79: Valores medios makespan de las 30 instancias de cada experimento para cada algoritmo utilizado, en el tipo en el que el 30% de los tiempos de procesamientos son cero.

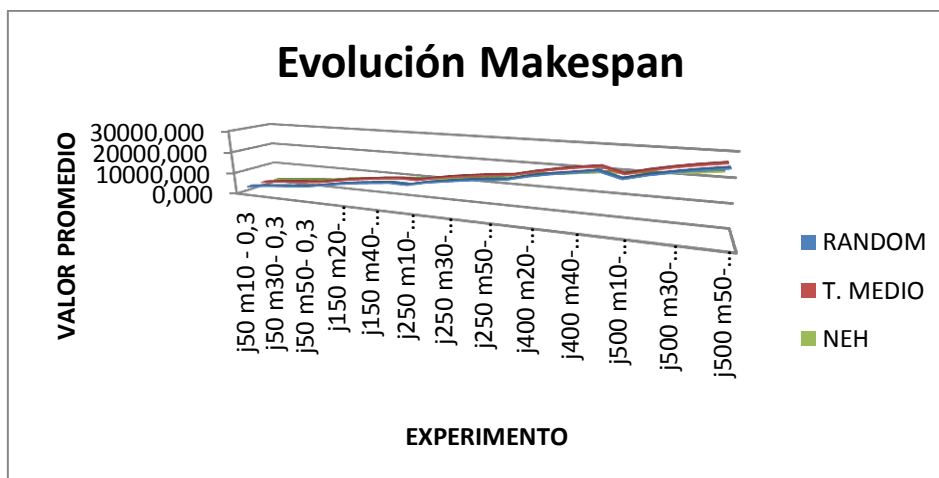


Figura 10: Evolución del makespan en los experimentos para cada algoritmo, en el tipo en el que el 30% de los tiempos de procesamientos son cero.

EXPERIMENTO	RANDOM	T. MEDIO	NEH
j50 m10 - 0,4	2836,600	2805,667	2171,767
j50 m20- 0,4	3712,367	3683,333	2778,833
j50 m30- 0,4	4437,167	4398,933	3394,633
j50 m40- 0,4	5021,833	5000,033	3965,100
j50 m50- 0,4	5626,067	5621,233	4547,833
j150 m10- 0,4	6654,667	6670,333	5480,367
j150 m20- 0,4	8074,367	7937,867	6210,400
j150 m30- 0,4	8999,500	8919,200	6925,633
j150 m40- 0,4	9934,000	9827,300	7736,433
j150 m50- 0,4	10729,267	10818,700	8549,767
j250 m10- 0,4	10304,133	10261,600	8747,267
j250 m20- 0,4	11975,867	11891,233	9547,100
j250 m30- 0,4	13260,733	13160,567	10334,033
j250 m40- 0,4	14251,900	14320,733	11190,267
j250 m50- 0,4	15230,067	15336,633	12125,133
j400 m10- 0,4	15443,567	15570,567	13791,267
j400 m20- 0,4	17541,900	17498,800	14414,067
j400 m30- 0,4	18992,633	19013,633	15246,167
j400 m40- 0,4	20387,433	20363,200	16234,700
j400 m50- 0,4	21593,900	21573,567	17190,033
j500 m10- 0,4	18870,367	18842,067	16981,567
j500 m20- 0,4	21147,667	21017,533	17564,933
j500 m30- 0,4	22683,367	22685,367	18493,700
j500 m40- 0,4	24173,367	24182,333	19389,967
j500 m50- 0,4	25473,533	25458,533	20522,300
PROMEDIO	13494,251	13474,359	10941,331

Tabla 80: Valores medios makespan de las 30 instancias de cada experimento para cada algoritmo utilizado, en el tipo en el que el 40% de los tiempos de procesamientos son cero.

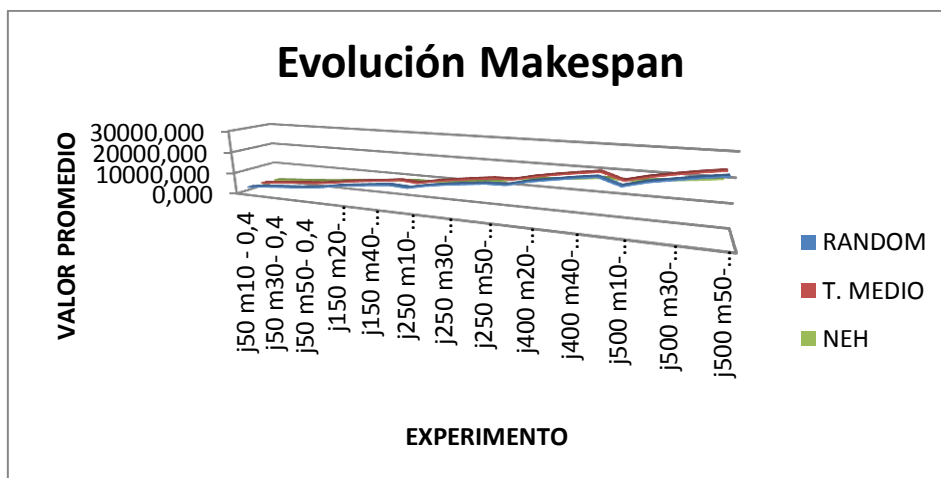


Figura 11: Evolución del makespan en los experimentos para cada algoritmo, en el tipo en el que el 40% de los tiempos de procesamientos son cero.

Como se ha podido observar, destacar que en las gráficas de la evolución del makespan (figuras 8, 9, 10 y 11) la tendencia que muestra es ascendente conforme aumenta el número de trabajos así como un pico descendente en el momento de analizar 400 trabajos-50 máquinas y 500 trabajos y 10 máquinas. El mismo patrón se repite para cada uno de los problemas definidos por el porcentaje de ceros.

Seguidamente se presentan las medias de makespan obtenidas para cada algoritmo en cada uno de los experimentos especificados por el porcentaje de ceros que forman sus tiempos de proceso.

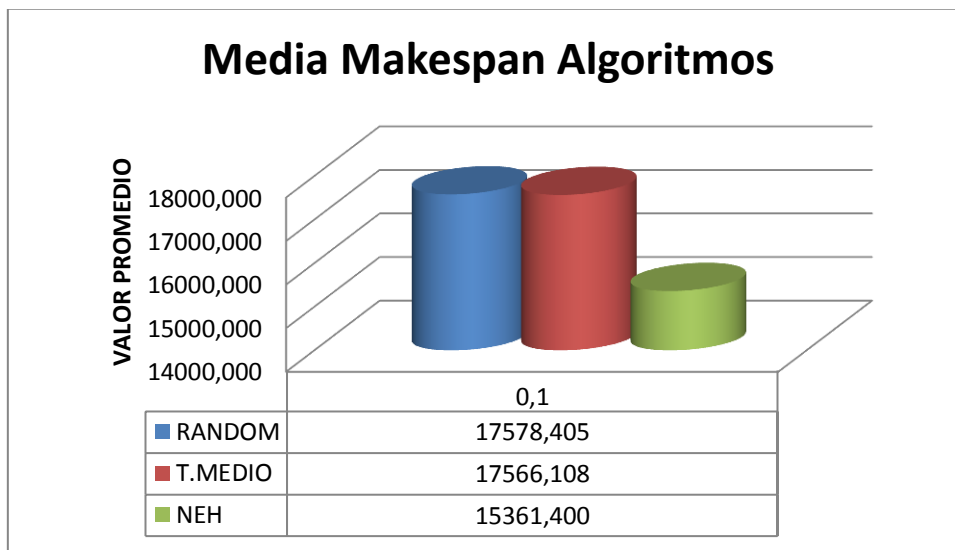


Figura 12: Valor medio del makespan de los algoritmos para el tipo de 10% de los tiempos ceros.

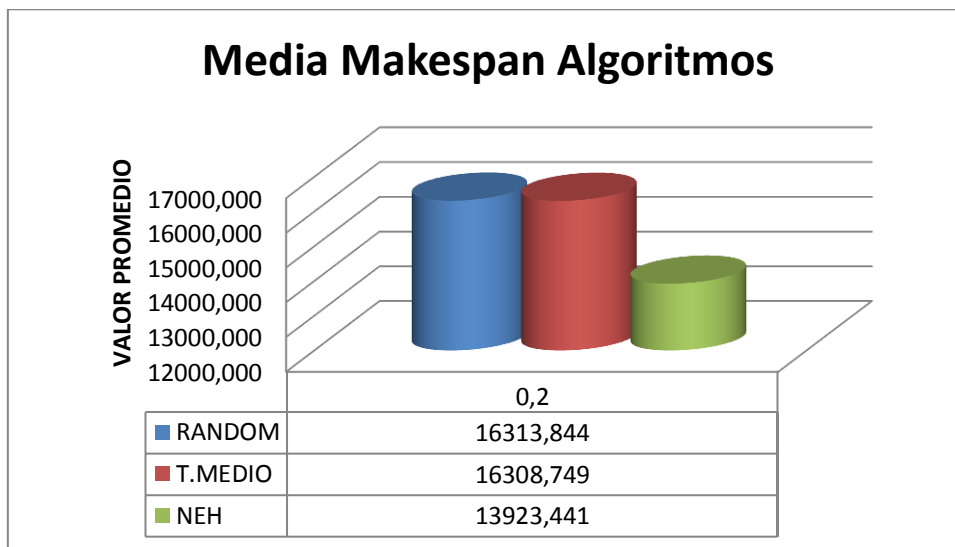


Figura 13: Valor medio del makespan de los algoritmos para el tipo de 20% de los tiempos ceros.

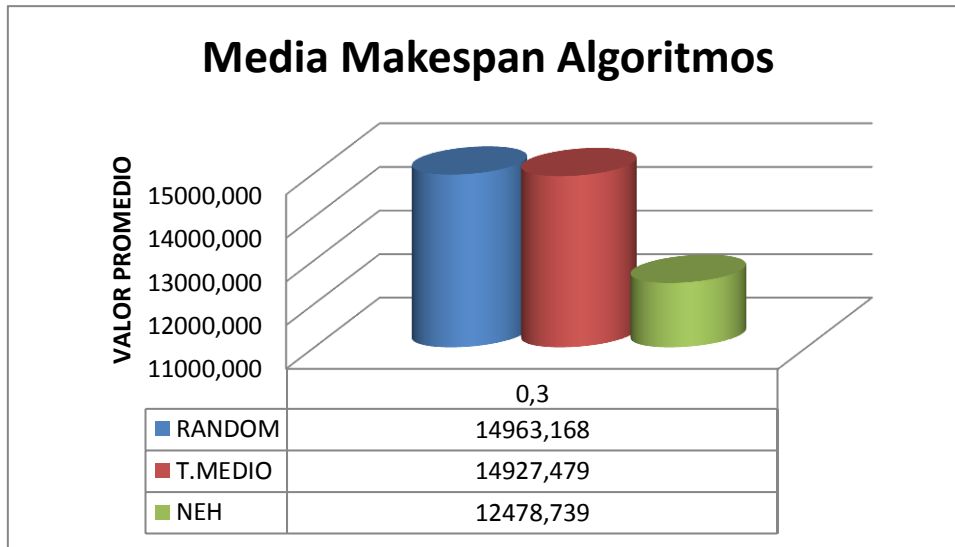


Figura 14: Valor medio del makespan de los algoritmos para el tipo de 30% de los tiempos ceros.

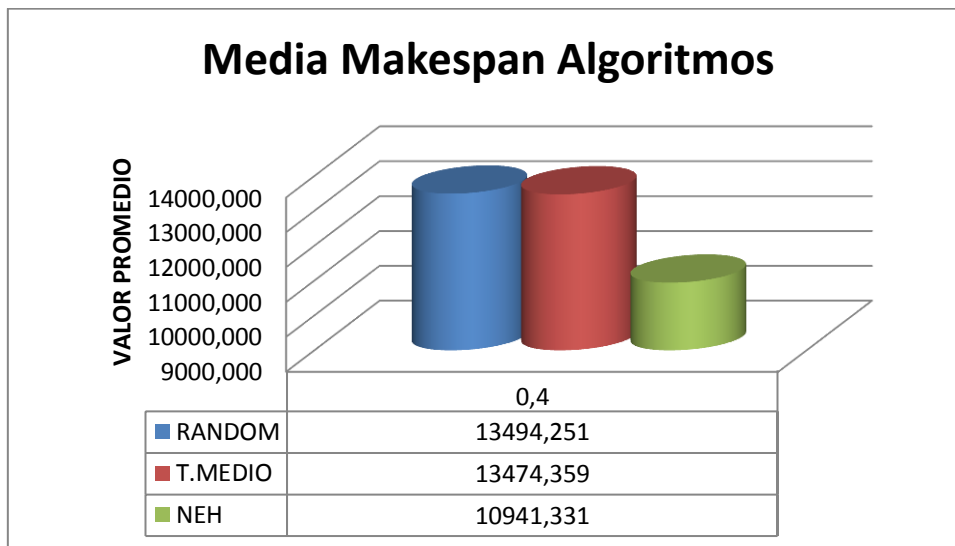


Figura 15: Valor medio del makespan de los algoritmos para el tipo de 40% de los tiempos ceros.

En las figuras 12, 13, 14 y 15 se aprecia la misma línea de resultados, siendo éstos parejos para los algoritmos Random y Tiempos medios. Los mejores resultados como era de prever se dan para el algoritmo basado en la heurística NEH. Y por último, incidir en la disminución del makespan obtenido conforme aumenta el número de tiempos que son ceros de los trabajos en las máquinas.

Los datos obtenidos de desviación típica para los experimentos que abarcan desde que el 10% de los tiempos de procesamiento sean cero hasta el 40%, son los siguientes.

EXPERIMENTO	RANDOM	T. MEDIO	NEH
j50 m10 - 0,1	124,838	165,687	102,925
j50 m20- 0,1	104,390	116,142	96,098
j50 m30- 0,1	159,057	147,710	90,302
j50 m40- 0,1	129,373	126,351	99,720
j50 m50- 0,1	159,930	157,052	113,233
j150 m10- 0,1	226,037	276,121	272,870
j150 m20- 0,1	230,586	213,281	137,601
j150 m30- 0,1	235,855	176,741	134,410
j150 m40- 0,1	226,420	214,692	112,886
j150 m50- 0,1	173,187	184,449	96,616
j250 m10- 0,1	255,574	233,813	344,918
j250 m20- 0,1	319,153	299,344	191,879
j250 m30- 0,1	256,503	277,571	151,953
j250 m40- 0,1	307,822	211,636	104,459
j250 m50- 0,1	219,434	204,859	117,433
j400 m10- 0,1	316,970	338,476	510,587
j400 m20- 0,1	348,842	341,396	179,940
j400 m30- 0,1	356,628	292,945	192,495
j400 m40- 0,1	329,315	256,326	157,959
j400 m50- 0,1	278,033	378,150	149,634
j500 m10- 0,1	388,544	479,170	518,805
j500 m20- 0,1	393,882	333,191	452,937
j500 m30- 0,1	332,535	445,930	247,362
j500 m40- 0,1	347,906	374,813	192,575
j500 m50- 0,1	305,744	268,276	132,552
DESVIACIÓN	261,062	260,565	196,086

Tabla 81: Valores medios de la desviación típica de las 30 instancias de cada experimento para cada algoritmo utilizado, en el tipo en el que el 10% de los tiempos de procesamientos son cero.

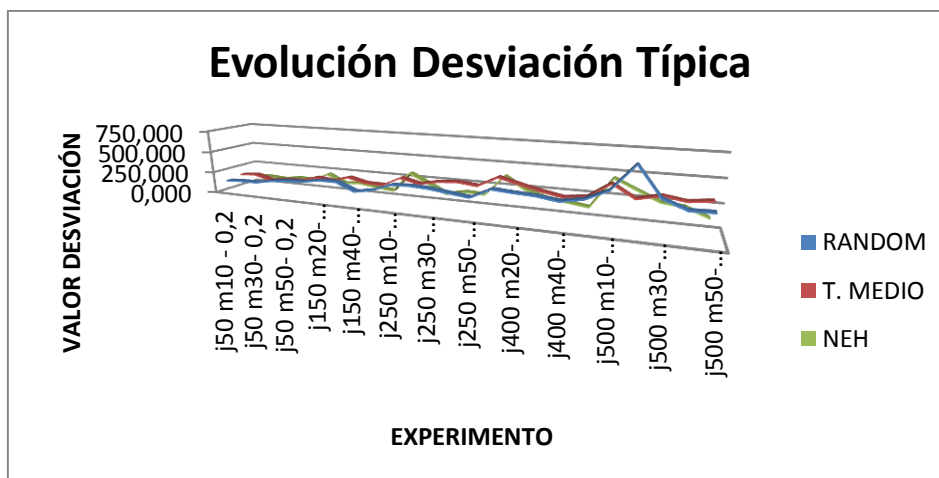


Figura 16: Evolución de la desviación típica en los experimentos para cada algoritmo, en el tipo en el que el 10% de los tiempos de procesamientos son cero.

EXPERIMENTO	RANDOM	T. MEDIO	NEH
j50 m10 - 0,2	136,612	162,249	99,564
j50 m20- 0,2	146,973	178,783	102,928
j50 m30- 0,2	149,687	125,823	91,261
j50 m40- 0,2	186,673	152,755	121,555
j50 m50- 0,2	206,443	158,115	112,181
j150 m10- 0,2	218,548	217,303	207,980
j150 m20- 0,2	253,595	195,536	111,988
j150 m30- 0,2	253,777	256,876	130,735
j150 m40- 0,1	164,189	208,605	102,095
j150 m50- 0,2	203,125	205,769	80,803
j250 m10- 0,2	282,632	307,102	311,515
j250 m20- 0,2	293,089	259,053	208,632
j250 m30- 0,2	282,177	302,628	106,508
j250 m40- 0,2	255,973	319,482	151,067
j250 m50- 0,2	231,152	296,748	141,998
j400 m10- 0,2	340,079	410,917	370,445
j400 m20- 0,2	324,632	354,380	243,584
j400 m30- 0,2	313,914	303,506	172,356
j400 m40- 0,2	282,818	262,118	146,632
j400 m50- 0,2	325,072	289,594	116,568
j500 m10- 0,2	431,968	443,591	443,778
j500 m20- 0,2	695,571	313,482	342,041
j500 m30- 0,2	400,566	367,106	237,800
j500 m40- 0,2	313,028	332,507	214,415
j500 m50- 0,2	319,382	360,355	124,841
DESVIACIÓN	280,467	271,375	179,731

Tabla 82: Valores medios de la desviación típica de las 30 instancias de cada experimento para cada algoritmo utilizado, en el tipo en el que el 20% de los tiempos de procesamientos son cero.

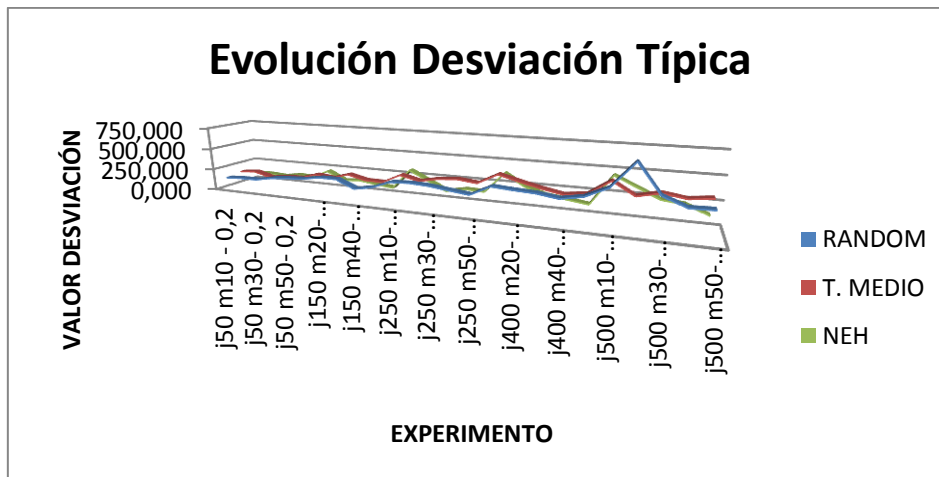


Figura 17: Evolución de la desviación típica en los experimentos para cada algoritmo, en el tipo en el que el 20% de los tiempos de procesamientos son cero.

EXPERIMENTO	RANDOM	T. MEDIO	NEH
j50 m10 - 0,3	142,287	173,069	95,583
j50 m20- 0,3	161,742	154,436	89,689
j50 m30- 0,3	163,331	188,705	94,246
j50 m40- 0,3	190,374	135,033	124,075
j50 m50- 0,3	194,528	173,640	109,968
j150 m10- 0,3	274,551	219,332	206,031
j150 m20- 0,3	207,211	240,615	150,464
j150 m30- 0,3	191,697	211,908	146,639
j150 m40- 0,3	216,543	256,171	114,322
j150 m50- 0,3	218,232	220,811	114,692
j250 m10- 0,3	294,186	471,830	239,048
j250 m20- 0,3	245,203	304,566	204,134
j250 m30- 0,3	239,709	353,057	162,584
j250 m40- 0,3	357,108	240,590	118,985
j250 m50- 0,3	269,650	217,387	141,408
j400 m10- 0,3	313,506	395,746	440,261
j400 m20- 0,3	306,764	415,745	284,325
j400 m30- 0,3	348,271	442,829	201,178
j400 m40- 0,3	449,717	359,629	173,397
j400 m50- 0,3	292,504	352,811	158,723
j500 m10- 0,3	467,883	570,682	663,545
j500 m20- 0,3	405,049	289,378	397,293
j500 m30- 0,3	390,307	422,904	337,807
j500 m40- 0,3	442,279	383,203	170,030
j500 m50- 0,3	291,657	322,253	150,705
DESVIACIÓN	282,972	300,653	203,565

Tabla 83: Valores medios de la desviación típica de las 30 instancias de cada experimento para cada algoritmo utilizado, en el tipo en el que el 30% de los tiempos de procesamientos son cero.

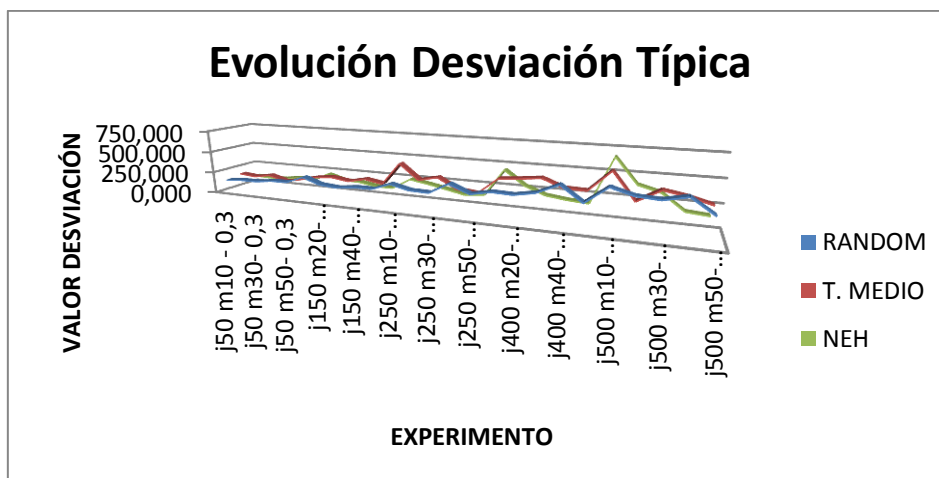


Figura 18: Evolución de la desviación típica en los experimentos para cada algoritmo, en el tipo en el que el 30% de los tiempos de procesamientos son cero.

EXPERIMENTO	RANDOM	T. MEDIO	NEH
j50 m10 - 0,4	130,296	183,817	96,579
j50 m20- 0,4	145,423	146,147	86,133
j50 m30- 0,4	193,344	172,797	91,041
j50 m40- 0,4	174,111	136,884	107,218
j50 m50- 0,4	170,770	138,824	125,338
j150 m10- 0,4	255,334	253,685	240,186
j150 m20- 0,4	231,377	217,608	164,444
j150 m30- 0,4	258,346	217,724	91,888
j150 m40- 0,4	236,560	188,398	106,820
j150 m50- 0,4	204,635	252,299	108,332
j250 m10- 0,4	370,117	338,720	323,082
j250 m20- 0,4	396,357	279,734	207,302
j250 m30- 0,4	318,562	288,220	162,367
j250 m40- 0,4	243,575	306,555	140,583
j250 m50- 0,4	332,462	248,406	123,258
j400 m10- 0,4	364,992	425,969	379,351
j400 m20- 0,4	348,973	395,327	274,615
j400 m30- 0,4	334,637	408,760	167,005
j400 m40- 0,4	356,323	305,694	195,977
j400 m50- 0,4	355,820	372,536	106,269
j500 m10- 0,4	397,923	476,501	584,469
j500 m20- 0,4	436,387	408,039	256,132
j500 m30- 0,4	361,649	492,519	209,015
j500 m40- 0,4	404,444	449,613	185,845
j500 m50- 0,4	331,495	429,287	127,425
DESVIACIÓN	294,156	301,363	186,427

Tabla 84: Valores medios de la desviación típica de las 30 instancias de cada experimento para cada algoritmo utilizado, en el tipo en el que el 40% de los tiempos de procesamientos son cero.

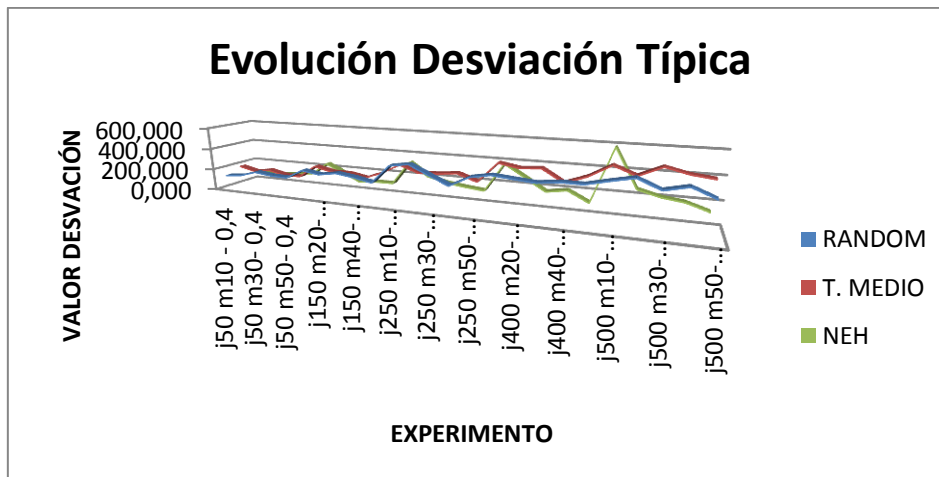


Figura 19: Evolución de la desviación típica en los experimentos para cada algoritmo, en el tipo en el que el 40% de los tiempos de procesamientos son cero.

La dispersión de puntos que se logra en las figuras 20, 21, 22 y 23 definen la evolución de la desviación típica para todos los experimentos relacionados con los algoritmos utilizados. Destacar que la desviación de los valores obtenidos para el algoritmo NEH es mayor respectivamente cuando el porcentaje de tiempos de proceso que son cero aumenta. Para los algoritmos Random y Tiempo medio la dispersión es semejante en todos los experimentos, alcanzándose algunos picos más notable cuando los tiempos de proceso que son cero son del 10% y el 20%.

A continuación, se muestran los resultados de las medias de la desviación típica obtenidas para cada algoritmo en cada uno de los experimentos especificados por el porcentaje de ceros que forman sus tiempos de proceso.

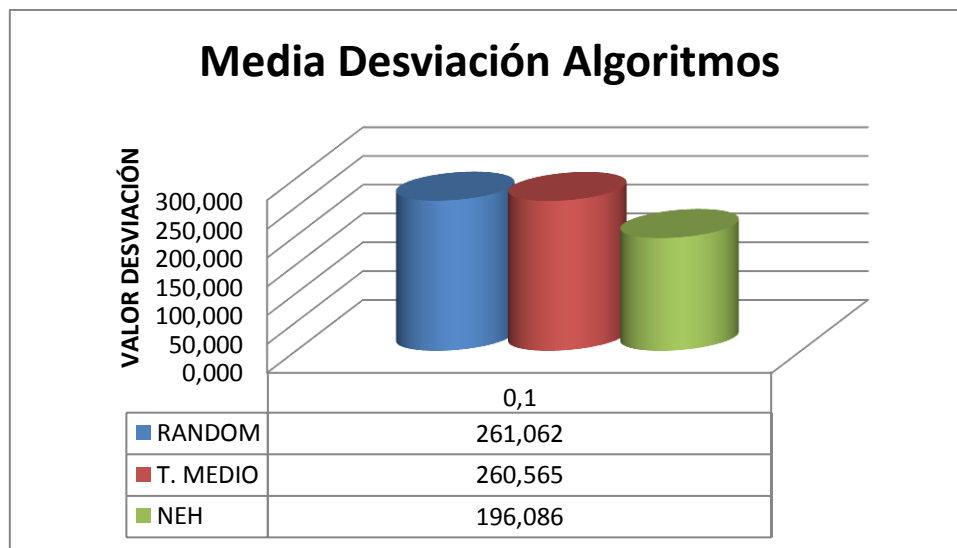


Figura 20: Valor medio de la desviación típica de los algoritmos para todos los experimentos y cada tipo de algoritmo, en el tipo en el que el 10% de los tiempos de procesamientos son cero.

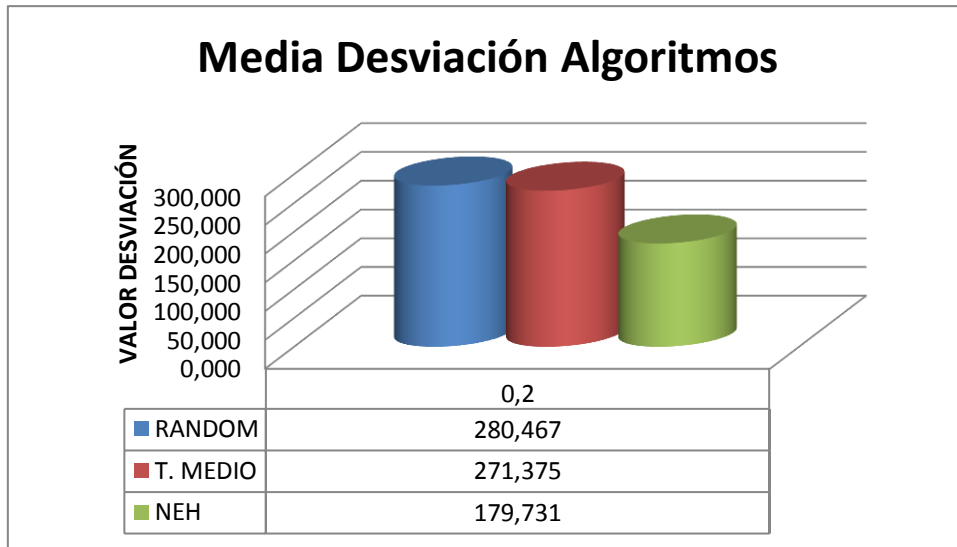


Figura 21: Valor medio de la desviación típica de los algoritmos para todos los experimentos y cada tipo de algoritmo, en el tipo en el que el 20% de los tiempos de procesamientos son cero.

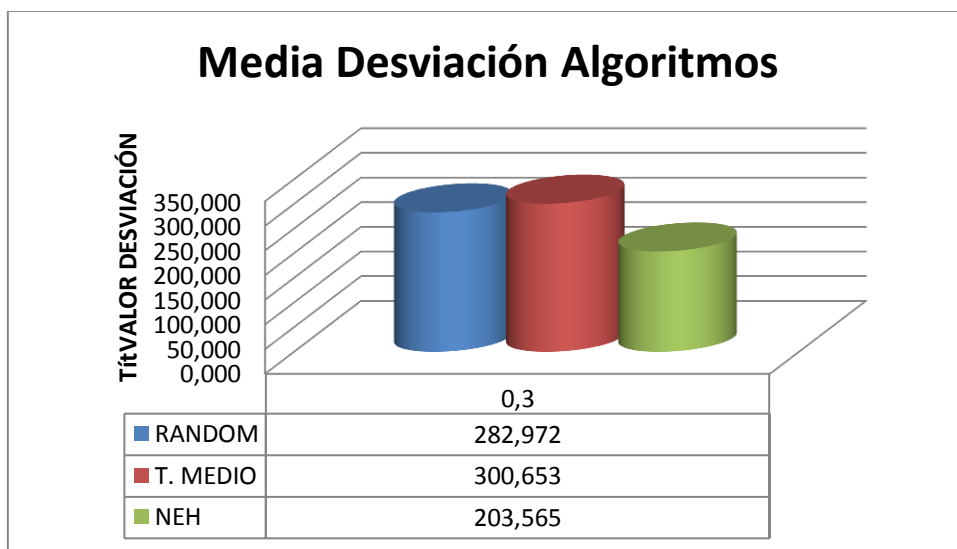


Figura 22: Valor medio de la desviación típica de los algoritmos para todos los experimentos y cada tipo de algoritmo, en el tipo en el que el 30% de los tiempos de procesamientos son cero.

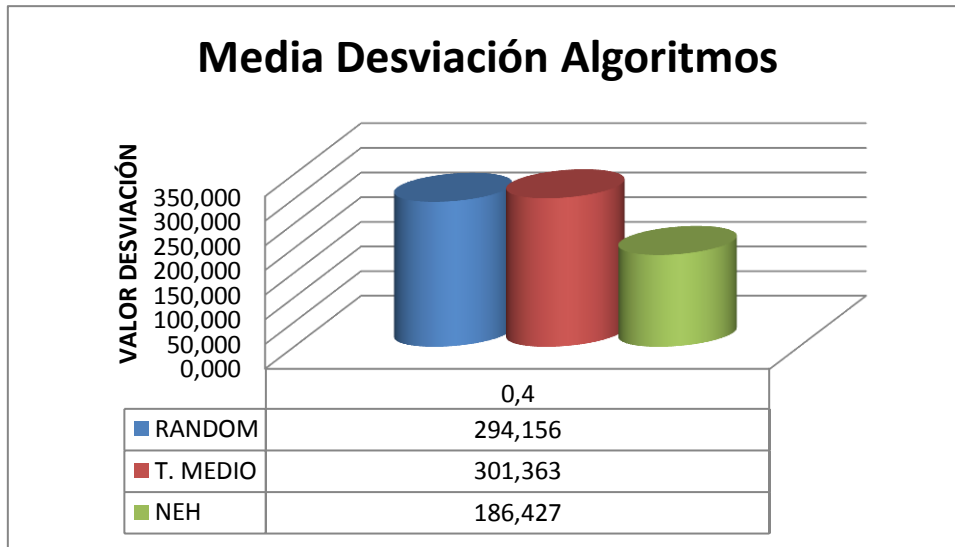


Figura 23: Valor medio de la desviación típica de los algoritmos para todos los experimentos y cada tipo de algoritmo, en el tipo en el que el 40% de los tiempos de procesamientos son cero.

En las figuras número 24, 25, 26 y 27 se observa la media de la desviación típica de los experimentos para los algoritmos utilizados en este estudio. Cabe destacar la gran diferencia que existe entre el algoritmo NEH respecto al algoritmo Random y algoritmo de los Tiempos medios. Aunque no es de gran interés el valor que recogen estos gráficos, nos sirve para mostrar la mejor solución que se obtiene de este estudio y que se encuentra para NEH, en todos los ámbitos.

5.2 Comparación de los algoritmos

El problema de Flow-Shop o problema de secuenciación de tareas de producción lineal, consiste en m máquinas dispuestas en serie, donde se procesan trabajos de flujo unidireccional que requieren m operaciones, cada una en una máquina distinta y siguiendo el mismo orden, esto es, primero en la máquina 1, después en la máquina 2, y así sucesivamente. Este problema, como bien se describió en apartados anteriores, en el campo de secuenciación de tareas es de difícil solución. Además en este estudio se analiza la programación de operaciones en un taller de flujo con operaciones perdidas. Por este motivo, observaremos en este apartado del trabajo el comportamiento de los algoritmos utilizados y los resultados obtenidos para su posterior análisis.

Comparando el algoritmo Random propuesto con el algoritmo basado en la heurística NEH, desarrollado para aplicaciones industriales, se puede observar que existe un error porcentual desfavorable para el algoritmo Random y que va creciendo exponencialmente conforme se aumenta el número de trabajos a procesar, aun así la componente de azar que presenta dicha técnica, la dota de alcanzar en algunos casos resultados mínimamente satisfactorios. Por lo que el algoritmo de búsqueda aleatoria no puede llegar a ser una opción considerable para resolver problemas reales, y en este estudio se ha tratado como la alternativa a no realizar ninguna acción referente a la planificación de los trabajos.

Del mismo modo el algoritmo de los tiempos medios presenta un código no muy complejo y junto con el algoritmo NEH dan mejores resultados que algoritmo Random, por lo que si tuviéramos que elegir alguno de ellos, las opciones pasarían por discutir entre estos dos mencionados.

A continuación se mostrarán los resultados obtenidos mediante gráficas que revelan las soluciones adquiridas por los tres algoritmos utilizados y en las que se profundiza en el valor promedio alcanzado para cada uno de ellos según el porcentaje de tiempos de procesamiento que son ceros, así como el valor de la

desviación típica que es conseguido y el porcentaje de mejora de los algoritmos Random y T. medio respecto a NEH.

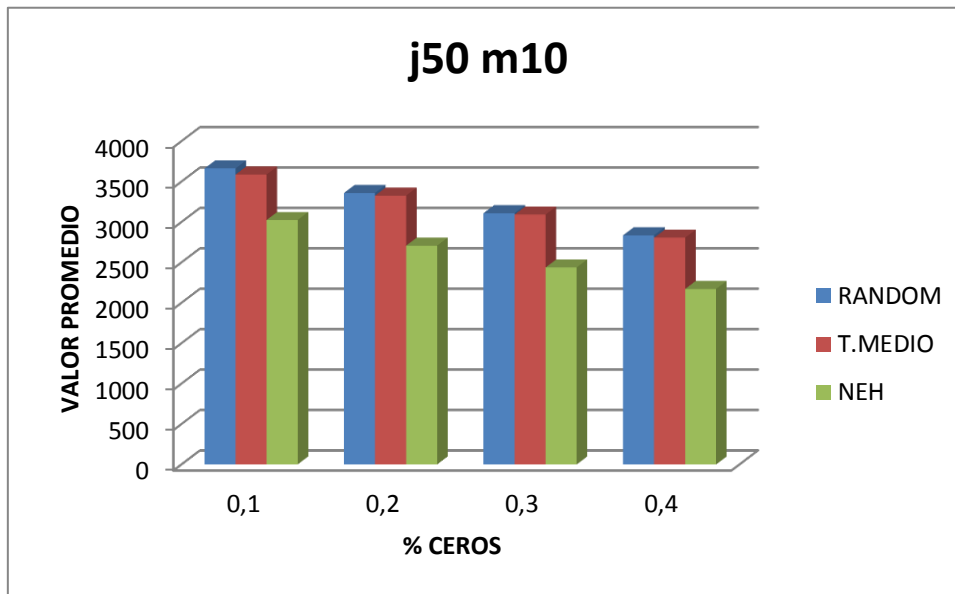


Figura 24: Valor promedio frente a porcentaje de cero para cada algoritmo del experimento de 50 trabajos y 10 máquinas.

Como podemos observar en la figura 24 para los cuatro valores de porcentaje de ceros los mejores resultados establecidos en minimizar el tiempo de finalización de todos los trabajos o makespan, los obtenemos para el algoritmo basado en la heurística NEH, hecho que siempre se va a producir durante todos los experimentos y que es debido a la mejor eficiencia para encontrar la mejor solución de este algoritmo. Apoyado en el primer experimento para 50 trabajos y 10 máquinas, observamos una diferencia considerable en el valor promedio para el algoritmo Random y el algoritmo de los tiempos medios con respecto a las secuencias más favorables y correctas alcanzadas con NEH.

Como este análisis será una constante durante todos los experimentos y solo va a cambiar el valor medio obtenido como resultado, se manifiesta que no habrá más observaciones respecto a este tipo de figuras, más allá de reflexionar que el tiempo de finalización de todos los trabajos va creciendo conforme aumenta el número de trabajos y el número de máquinas.

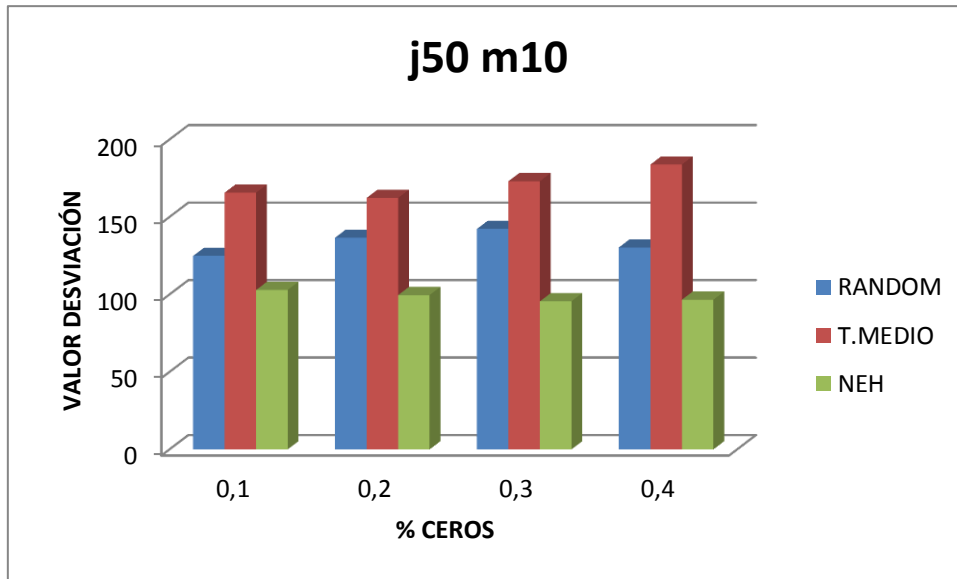


Figura 25: Valor desviación típica frente a porcentaje de cero para cada algoritmo del experimento de 50 trabajos y 10 máquinas.

El mismo tipo de gráfico anterior es utilizado en la figura 25, pero en este caso se reflejan los valores de la desviación típica alcanzada para los diferentes algoritmos. Se puede observar que el algoritmo basado en ordenar por tiempos medios alcanza una mayor desviación típica en sus valores y que el algoritmo NEH presenta la menor desviación en sus resultados.

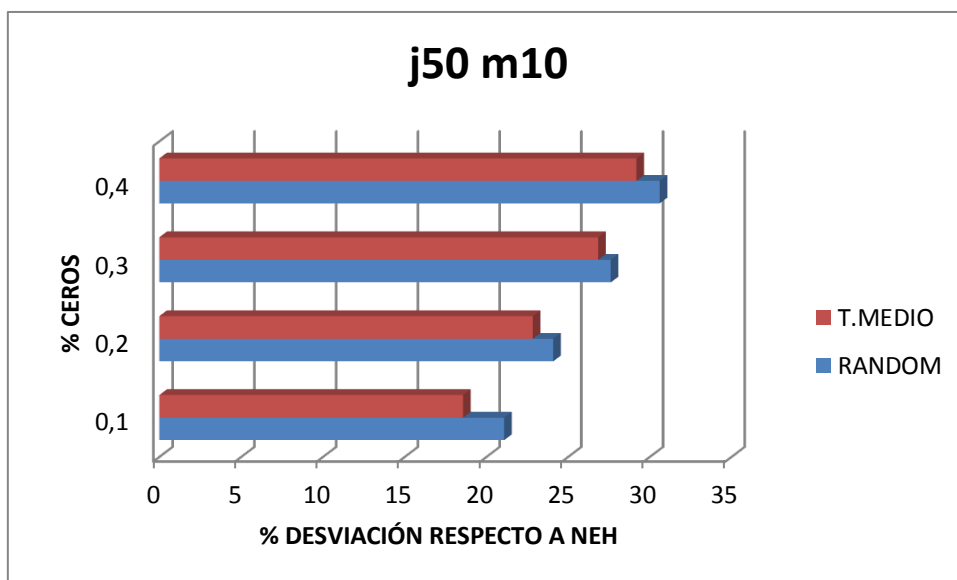


Figura 26: Porcentaje de mejora respecto a NEH de los algoritmos Random y T. Medios frente al porcentaje de ceros para el experimento de 50 trabajos y 10 máquinas.

La figura 26 refleja el porcentaje de desviación en los resultados de los algoritmos Random y tiempo medio respecto a la solución obtenida por NEH, siendo esta última la mejor solución para nuestro estudio. Destacar que conforme se aumenta el porcentaje de tiempos de procesamiento que son ceros, el error o desviación respecto a la mejor solución aumenta, siendo el algoritmo Random el que presenta mayor desviación en todos sus resultados.

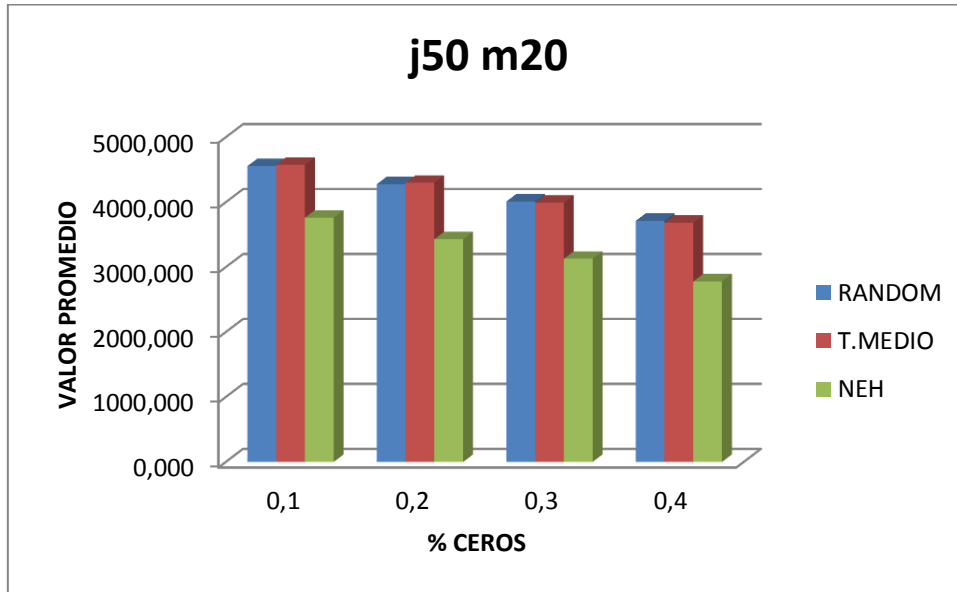


Figura 27: Valor promedio frente a porcentaje de cero para cada algoritmo del experimento de 50 trabajos y 20 máquinas.

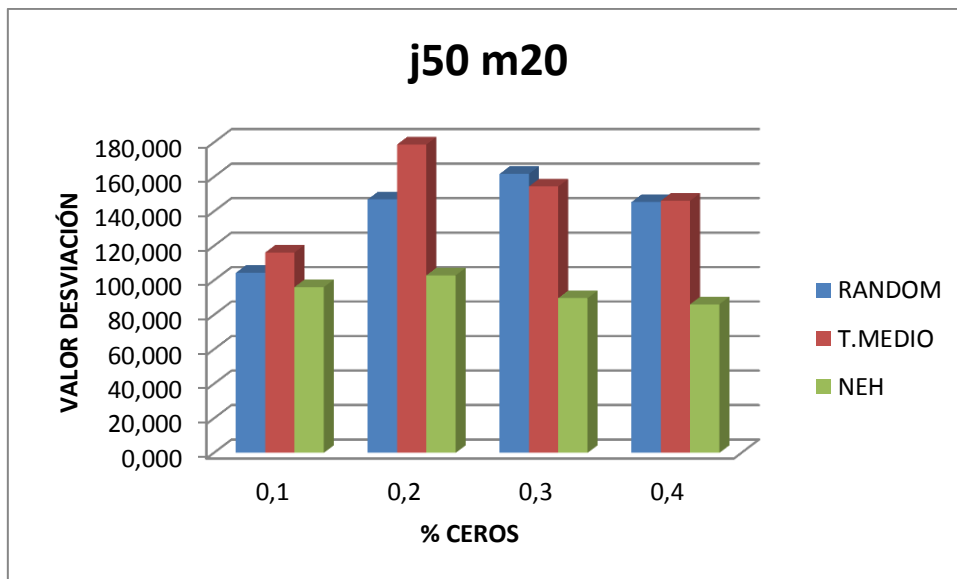


Figura 28: Valor desviación típica frente a porcentaje de cero para cada algoritmo del experimento de 50 trabajos y 20 máquinas.

A diferencia de los resultados de la figura 25, en la figura 28 se aprecia como el valor de la desviación típica para el algoritmo Random se va igualando o incluso superando a los valores obtenidos por el algoritmo tiempo medio. Del mismo modo el algoritmo NEH presenta la menor desviación típica en todos los experimentos.

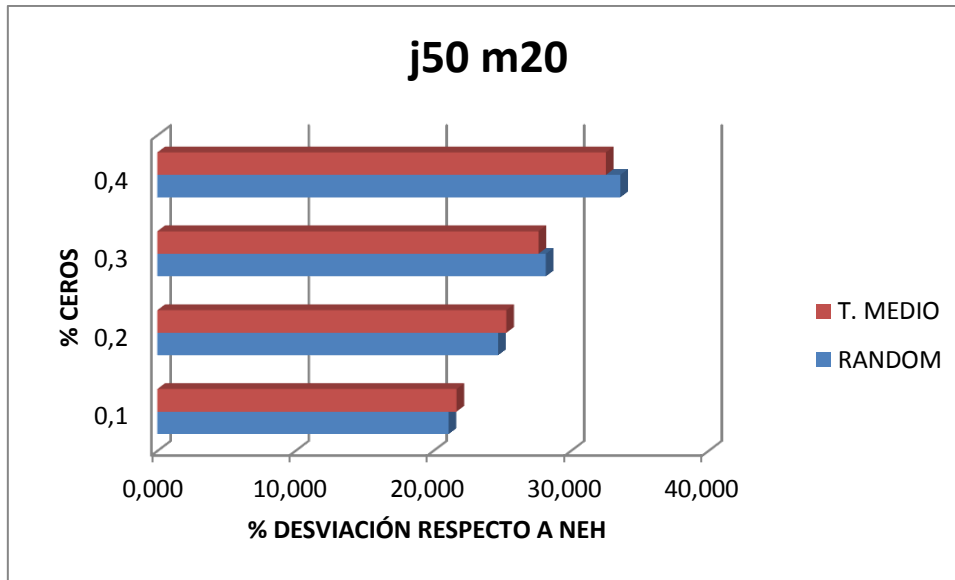


Figura 29: Porcentaje de mejora respecto a NEH de los algoritmos Random y T. Medios frente al porcentaje de ceros para el experimento de 50 trabajos y 20 máquinas.

La figura 29 refleja como el algoritmo Random obtiene peores resultados que el algoritmo basado en los tiempos medios acorde al aumento del porcentaje de tiempos de procesos que son ceros.

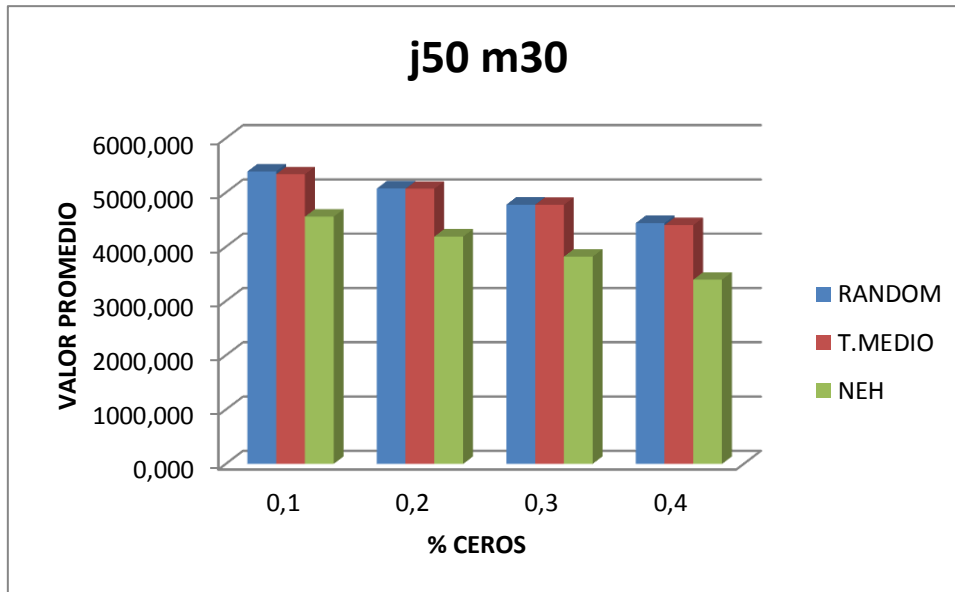


Figura 30: Valor promedio frente a porcentaje de cero para cada algoritmo del experimento de 50 trabajos y 30 máquinas.

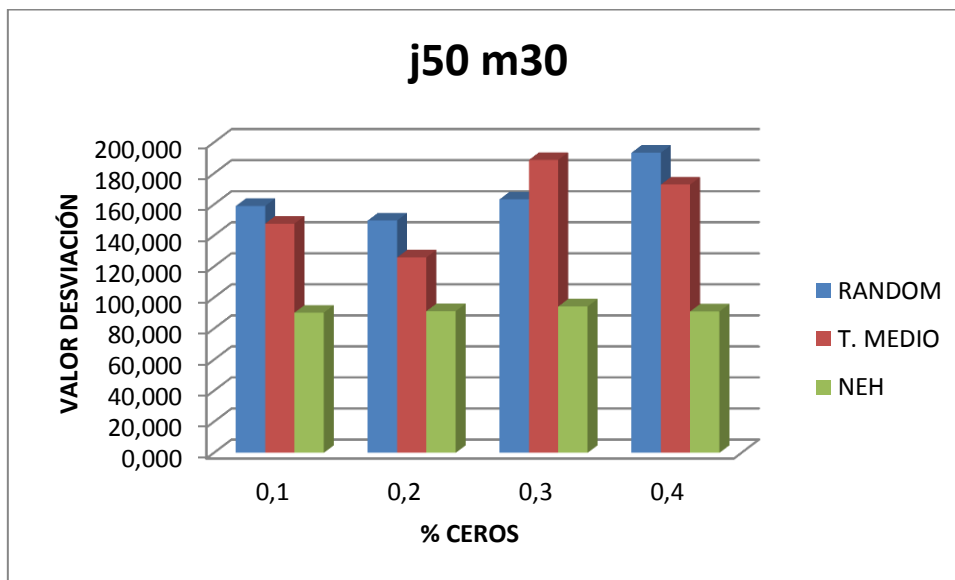


Figura 31: Valor desviación típica frente a porcentaje de cero para cada algoritmo del experimento de 50 trabajos y 30 máquinas.

La figura 31 muestra que los valores obtenidos para la desviación típica en el algoritmo Random superan a los valores conseguidos por el algoritmo tiempo medio salvo para el experimento del 30%. Por lo que se confirma la evolución que ya estaba presentando la figura 28 respecto a la figura 25.

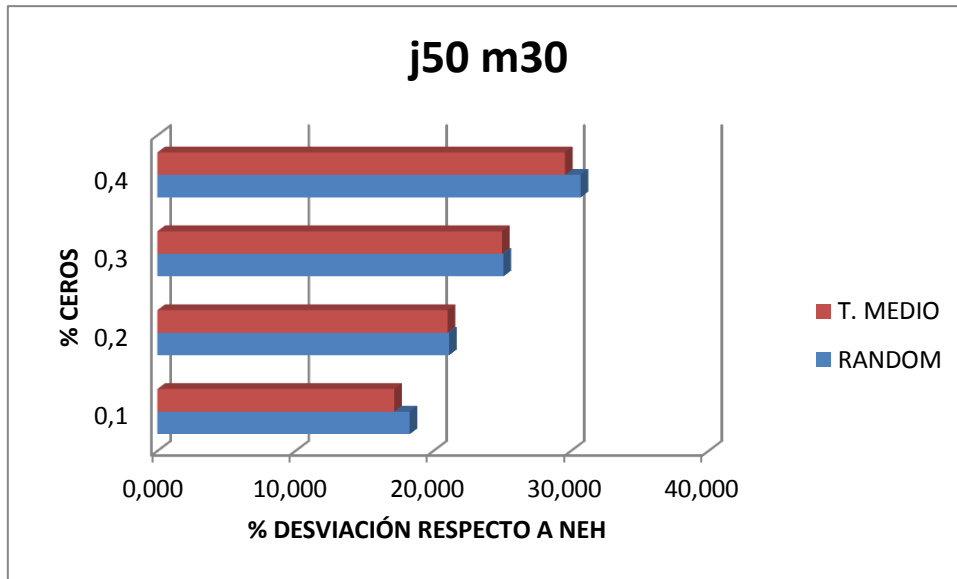


Figura 32: Porcentaje de mejora respecto a NEH de los algoritmos Random y T. Medios frente al porcentaje de ceros para el experimento de 50 trabajos y 30 máquinas.

Podemos observar en la figura 32 como permanece el mismo análisis comentado anteriormente, donde continúa el algoritmo Random presentando los peores resultados igualándose en algunas ocasiones con el algoritmo de los tiempos medios.

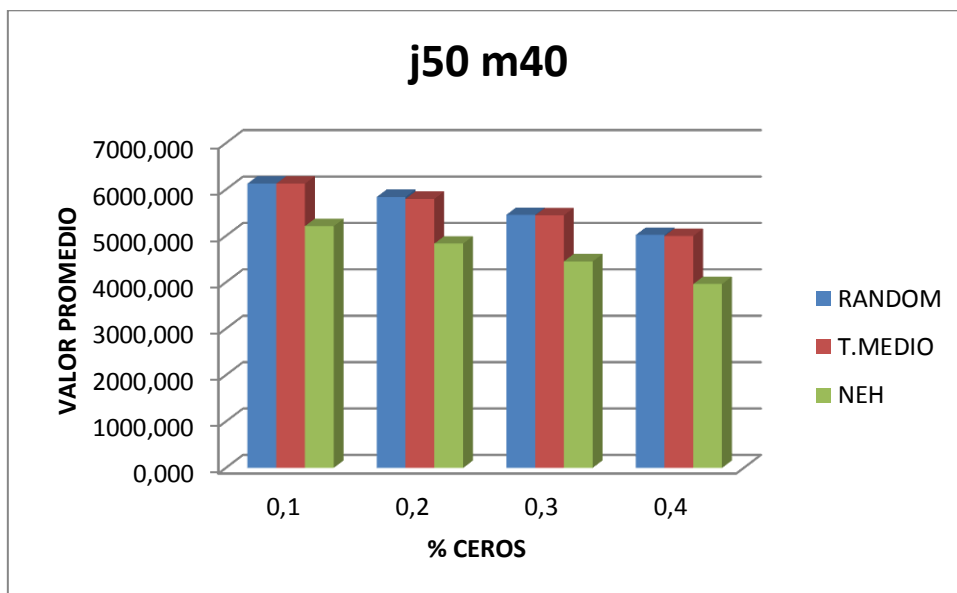


Figura 33: Valor promedio frente a porcentaje de cero para cada algoritmo del experimento de 50 trabajos y 40 máquinas.

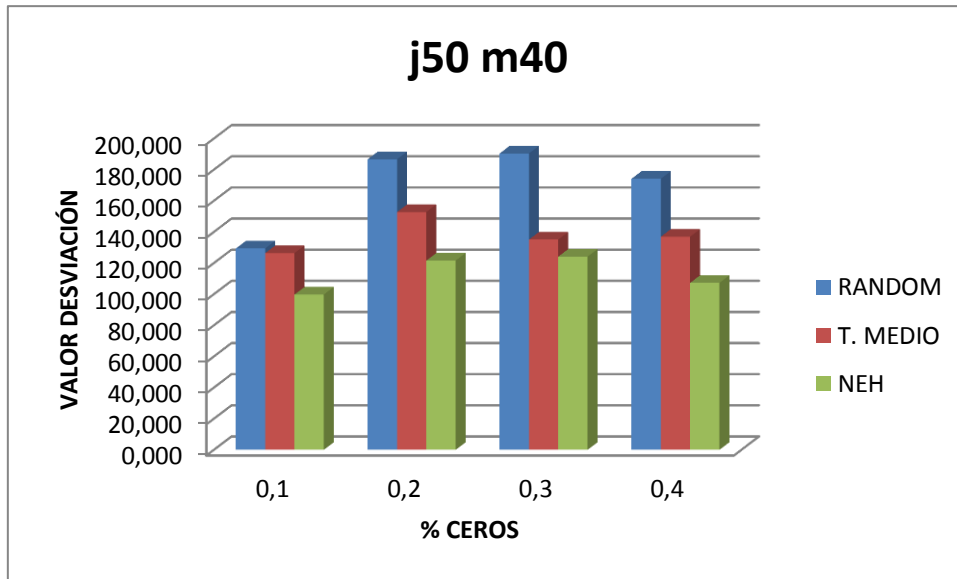


Figura 34: Valor desviación típica frente a porcentaje de cero para cada algoritmo del experimento de 50 trabajos y 40 máquinas.

En la figura 34 se refleja como el algoritmo Random obtiene mayores desviaciones que los otros dos algoritmos utilizados en este estudio.

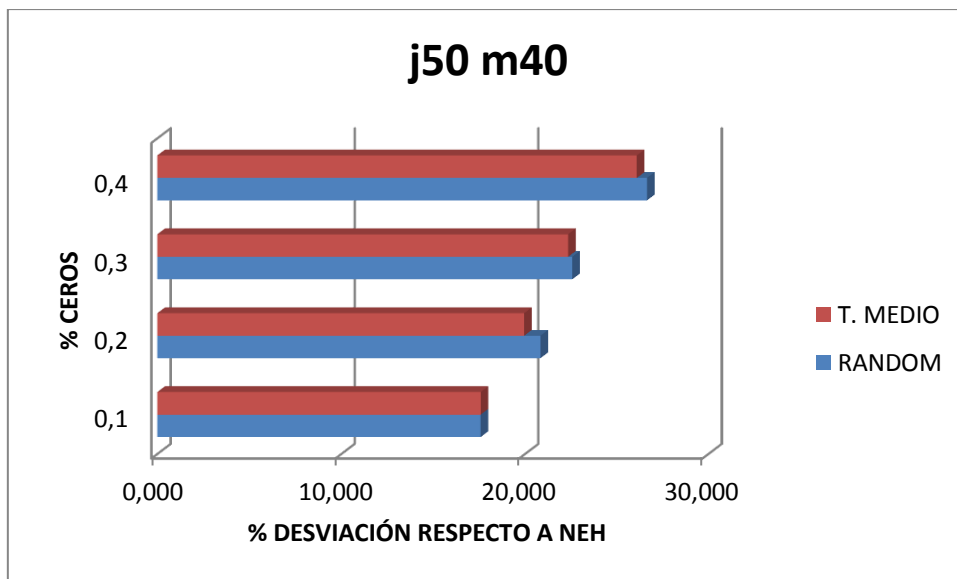


Figura 35: Porcentaje de mejora respecto a NEH de los algoritmos Random y T. Medios frente al porcentaje de ceros para el experimento de 50 trabajos y 40 máquinas.

La figura 35 muestra las mismas observaciones que se ha realizado para este gráfico en los anteriores experimentos.

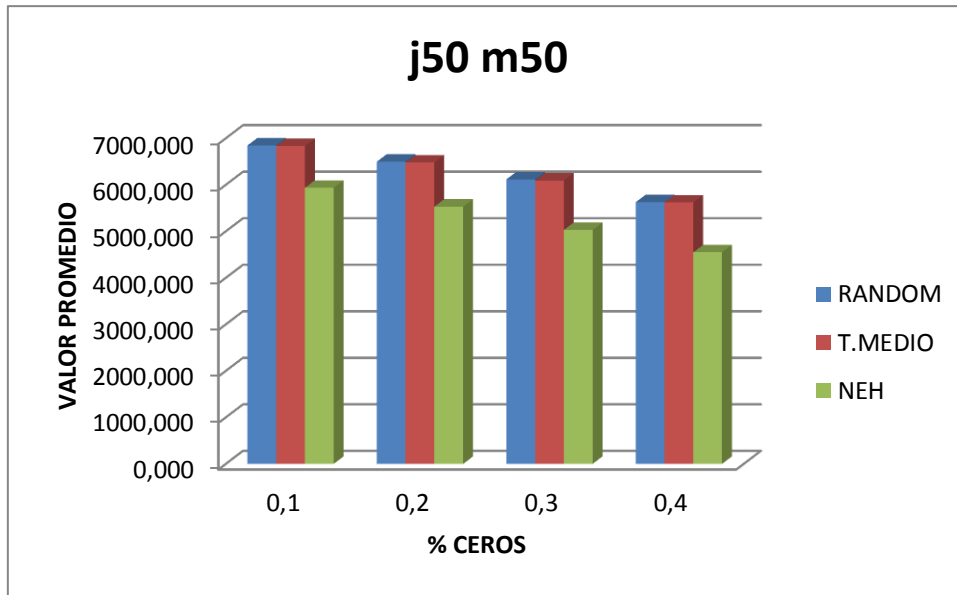


Figura 36: Valor promedio frente a porcentaje de cero para cada algoritmo del experimento de 50 trabajos y 50 máquinas.

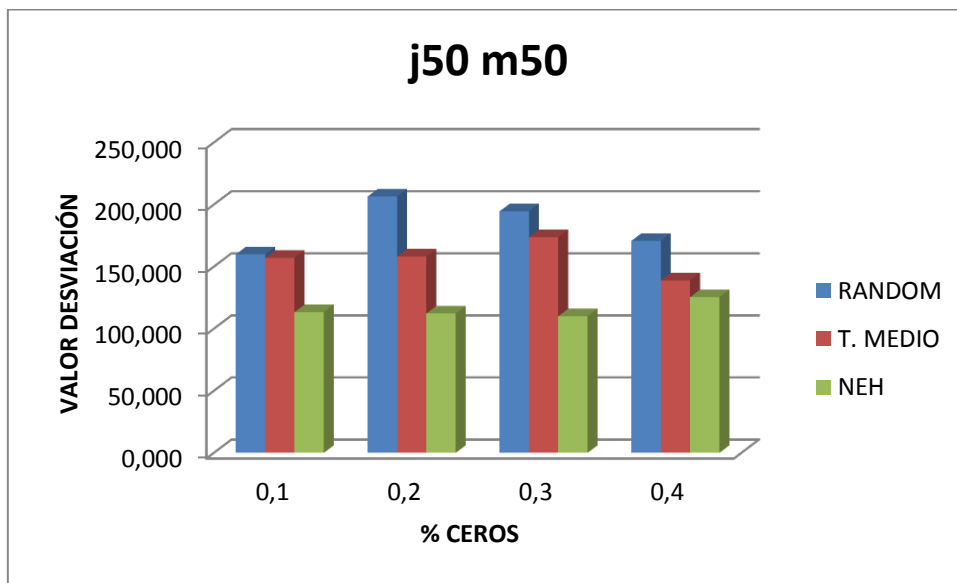


Figura 37: Valor desviación típica frente a porcentaje de cero para cada algoritmo del experimento de 50 trabajos y 50 máquinas.

La figura 37 refleja como el algoritmo Random obtiene mayores desviaciones que los otros dos algoritmos utilizados en este estudio, igual que se ha citado en los anteriores análisis.

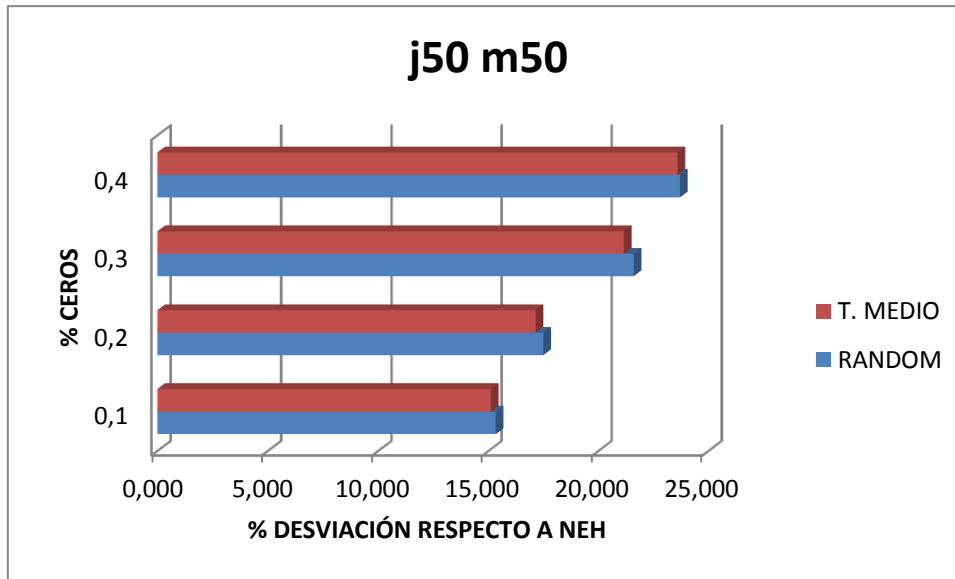


Figura 38: Porcentaje de mejora respecto a NEH de los algoritmos Random y T. Medios frente al porcentaje de ceros para el experimento de 50 trabajos y 50 máquinas.

Destacar, una vez observado la figura 38, que el porcentaje de error que muestran los dos algoritmos respecto a NEH son muy semejantes para los experimentos tratados en 50 trabajos.

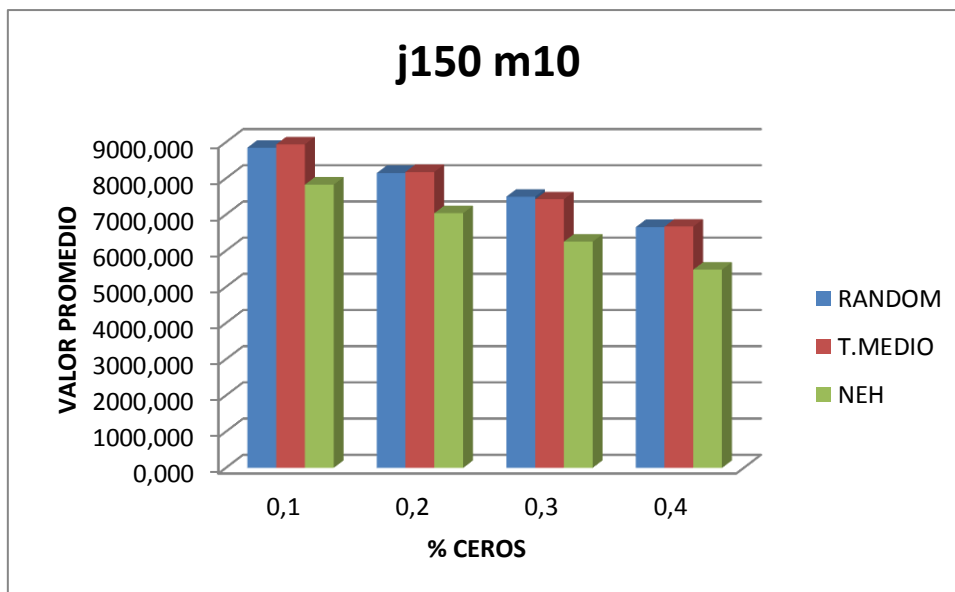


Figura 39: Valor promedio frente a porcentaje de cero para cada algoritmo del experimento de 150 trabajos y 10 máquinas.

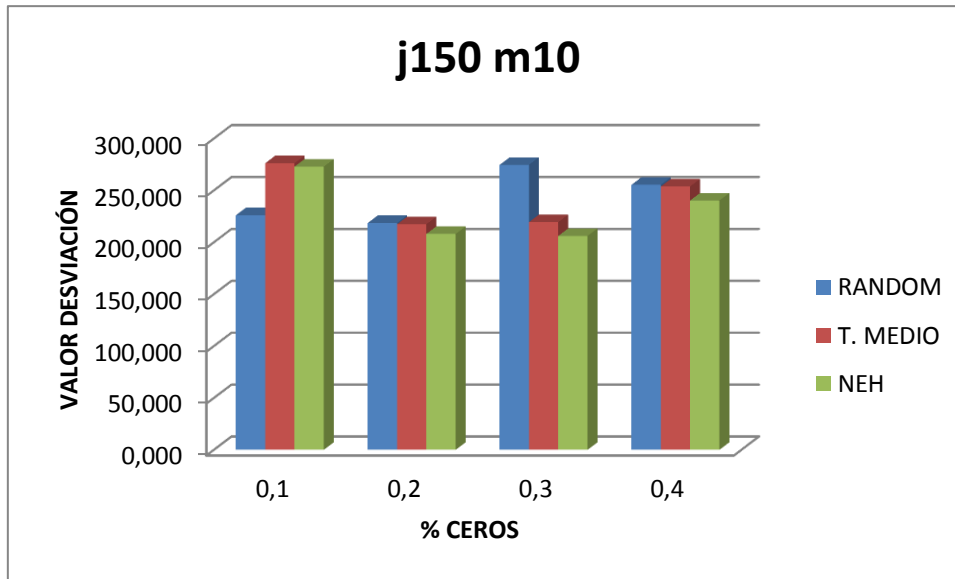


Figura 40: Valor desviación típica frente a porcentaje de cero para cada algoritmo del experimento de 150 trabajos y 10 máquinas.

En la figura 40 se aprecia como el algoritmo NEH presenta una alta desviación típica para el caso en el que el 10% de los tiempos de proceso de los trabajos en las máquinas son cero. En el resto de experimentos de este tipo alcanza el menor valor.

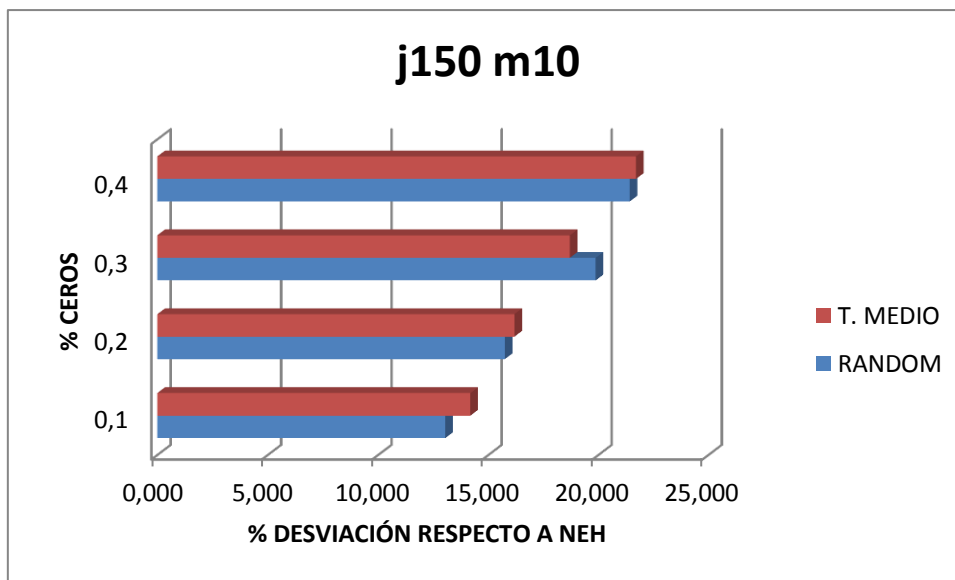


Figura 41: Porcentaje de mejora respecto a NEH de los algoritmos Random y T. Medios frente al porcentaje de ceros para el experimento de 150 trabajos y 10 máquinas.

Se puede observar en la figura 41 que el algoritmo de los tiempos medios obtiene una secuencia de trabajos peor, en la mayoría de los casos, que el algoritmo que consigue la secuencia de forma aleatoria. Puede ser un ejemplo donde se

manifiesta que este algoritmo tampoco alcanza una solución satisfactoria y adecuada para la programación de las operaciones.

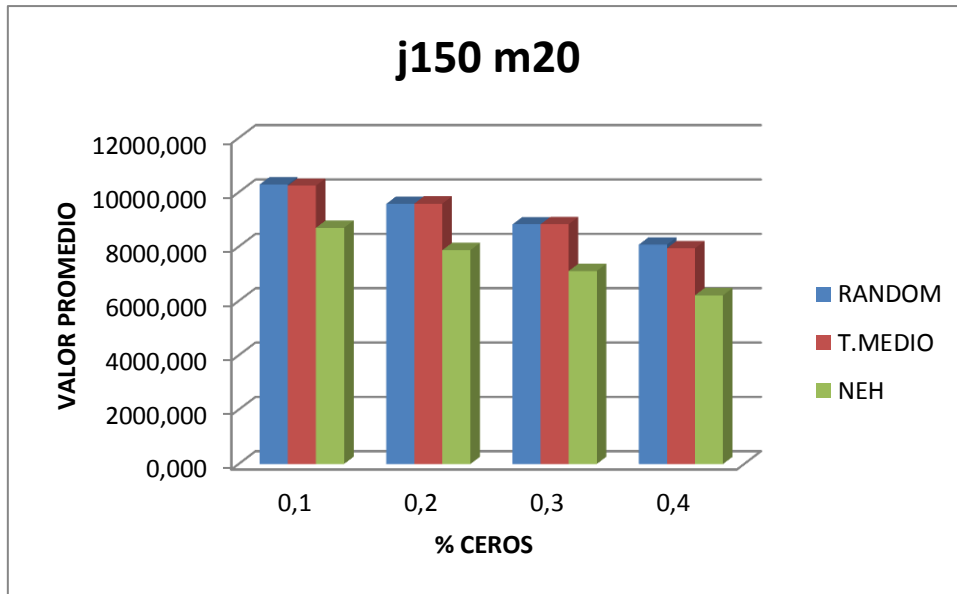


Figura 42: Valor promedio frente a porcentaje de cero para cada algoritmo del experimento de 150 trabajos y 20 máquinas.

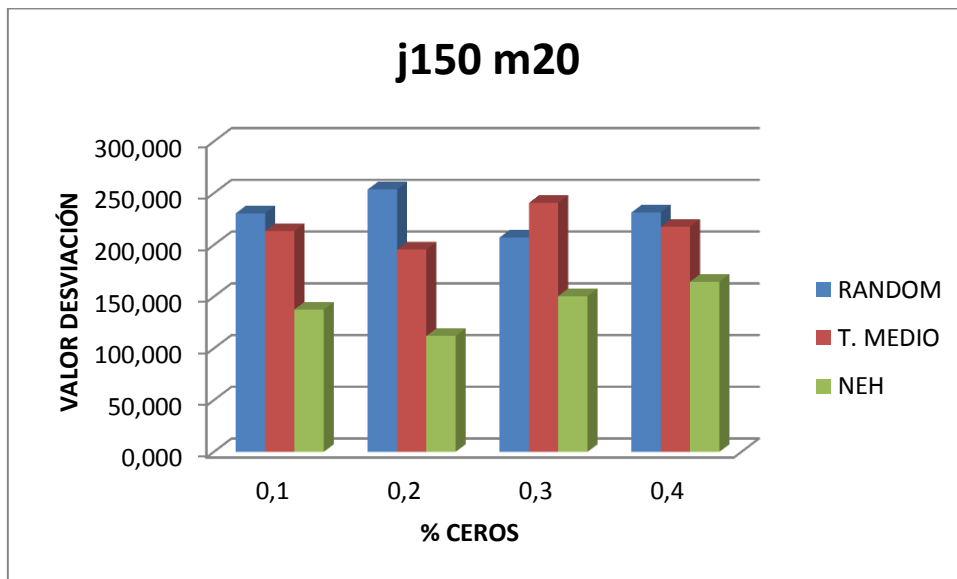


Figura 43: Valor desviación típica frente a porcentaje de cero para cada algoritmo del experimento de 150 trabajos y 20 máquinas.

La figura 43 exhibe las mismas observaciones citadas con anterioridad.

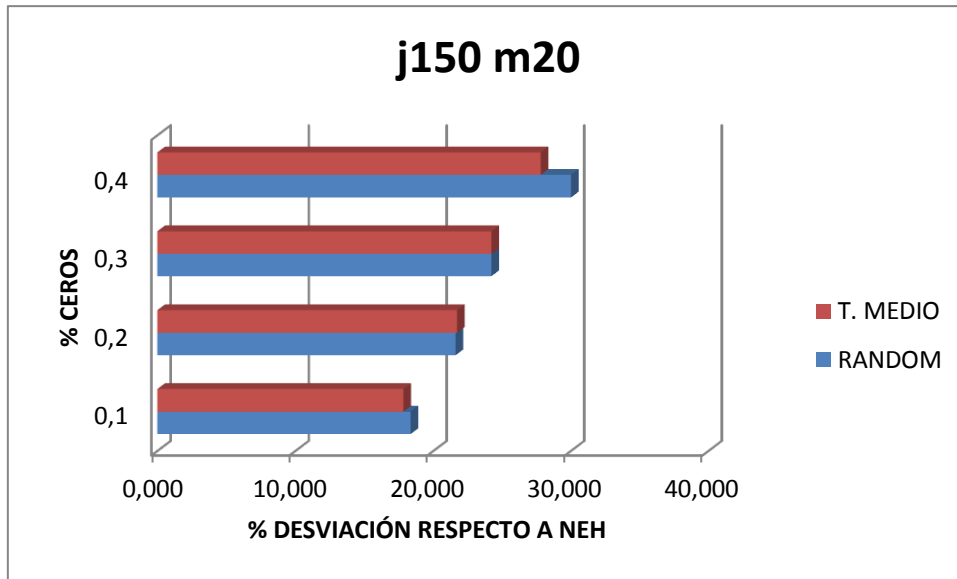


Figura 44: Porcentaje de mejora respecto a NEH de los algoritmos Random y T. Medios frente al porcentaje de ceros para el experimento de 150 trabajos y 20 máquinas.

La figura 44 muestra unos resultados muy semejantes en los algoritmos Random y tiempos medios, hecho que se repite en numerosas ocasiones.

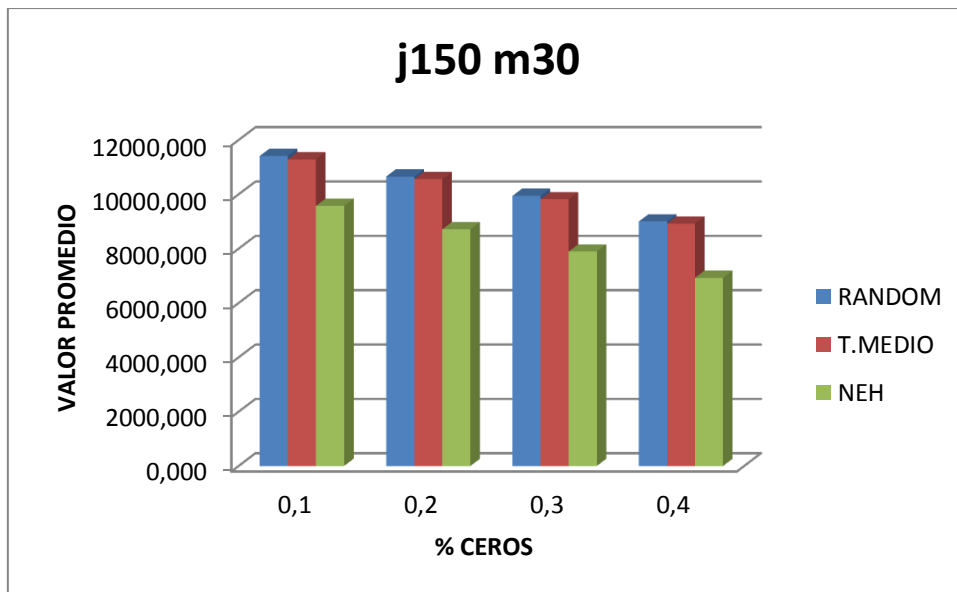


Figura 45: Valor promedio frente a porcentaje de cero para cada algoritmo del experimento de 150 trabajos y 30 máquinas.

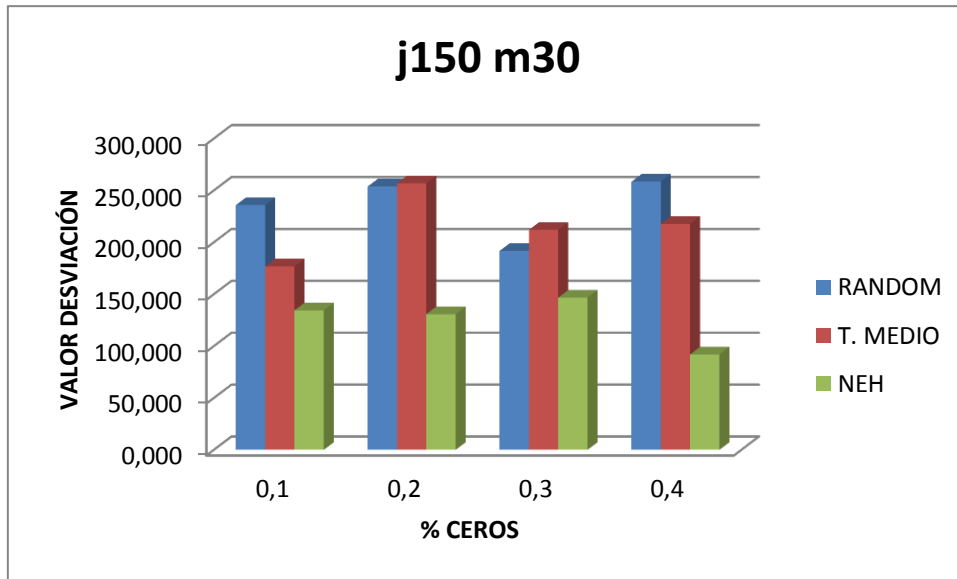


Figura 46: Valor desviación típica frente a porcentaje de cero para cada algoritmo del experimento de 150 trabajos y 30 máquinas.

La figura 46 refleja como el algoritmo Random obtiene mayores desviaciones que los otros dos algoritmos utilizados en este estudio, aunque para el caso en el que 30% de los tiempos de proceso son ceros, el algoritmo de los tiempos medios muestra una mayor desviación.

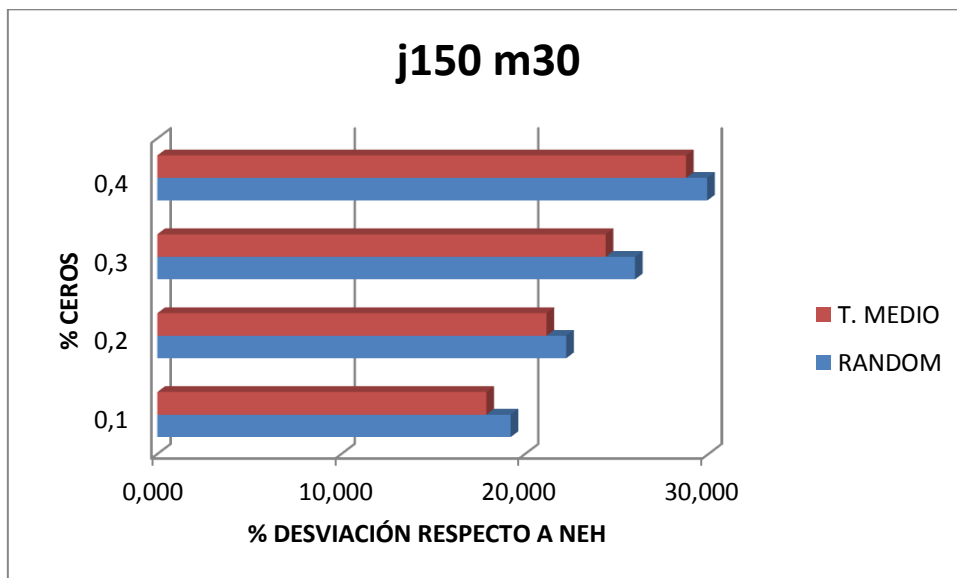


Figura 47: Porcentaje de mejora respecto a NEH de los algoritmos Random y T. Medios frente al porcentaje de ceros para el experimento de 150 trabajos y 30 máquinas.

Podemos observar en la figura 32 como permanece el mismo análisis comentado anteriormente, donde continúa el algoritmo Random presentando los peores resultados igualándose en algunas ocasiones con el algoritmo de los tiempos medios.

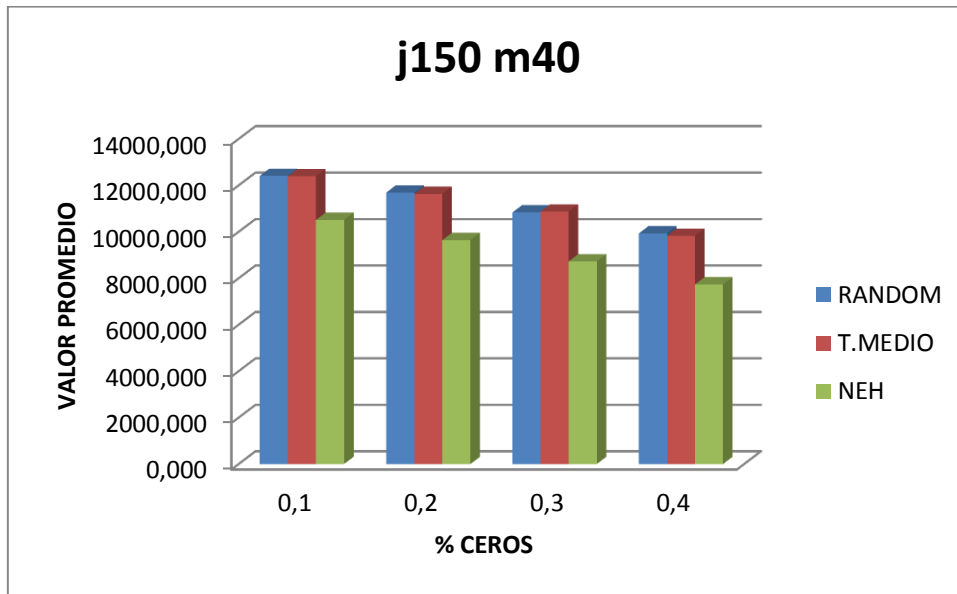


Figura 48: Valor promedio frente a porcentaje de cero para cada algoritmo del experimento de 150 trabajos y 40 máquinas.

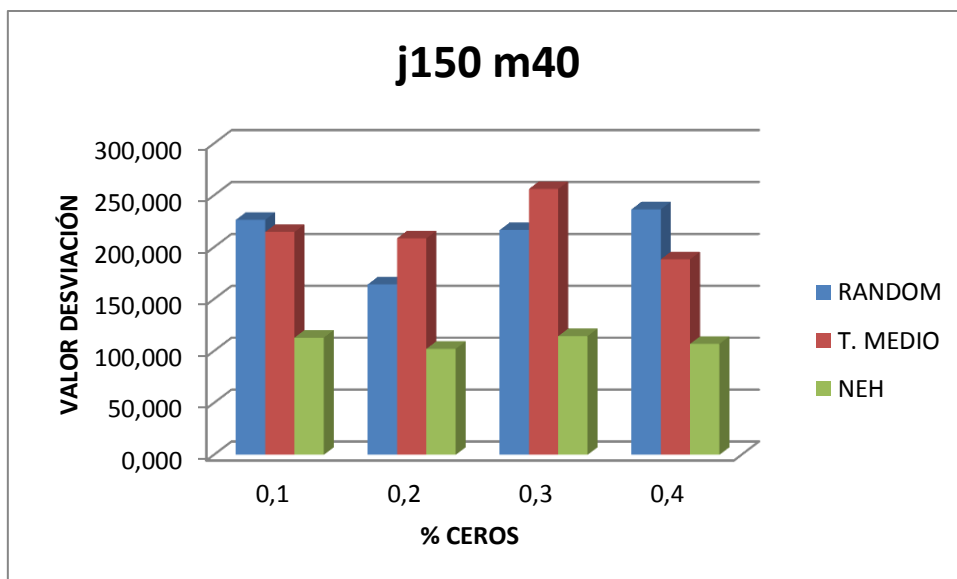


Figura 49: Valor desviación típica frente a porcentaje de cero para cada algoritmo del experimento de 150 trabajos y 40 máquinas.

En la figura 49 hay alternancia de cara a alcanzar la mayor desviación. Se observa que para los casos limites (10% y 40%) la desviación es mayor en el algoritmo Random. El algoritmo NEH refleja una desviación muy inferior respecto a los otros dos métodos utilizados.

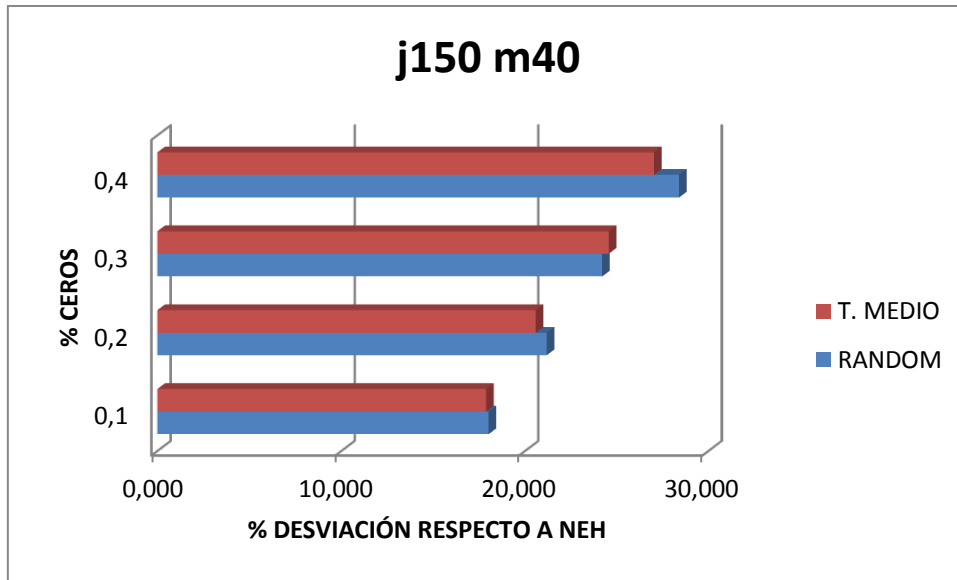


Figura 50: Porcentaje de mejora respecto a NEH de los algoritmos Random y T. Medios frente al porcentaje de ceros para el experimento de 150 trabajos y 40 máquinas.

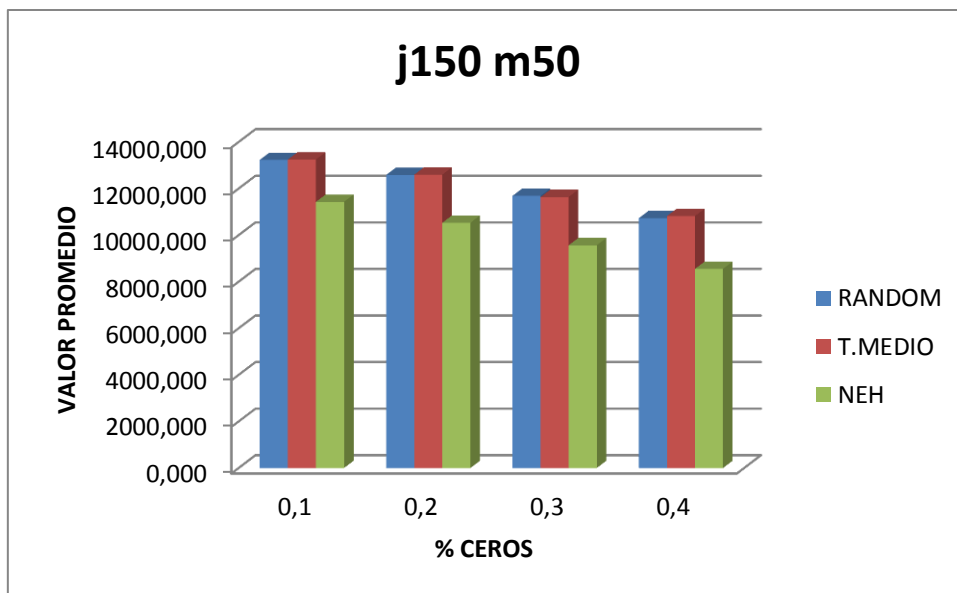


Figura 51: Valor promedio frente a porcentaje de cero para cada algoritmo del experimento de 150 trabajos y 50 máquinas.

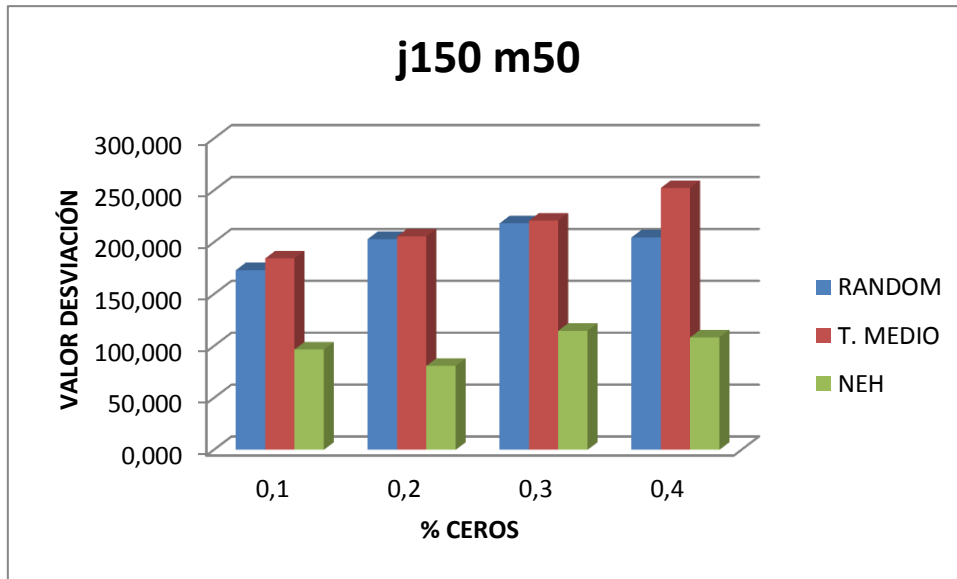


Figura 52: Valor desviación típica frente a porcentaje de cero para cada algoritmo del experimento de 150 trabajos y 50 máquinas.

Se observa en la figura 52 que el algoritmo de los tiempos medios presenta en todos los experimentos la mayor desviación. Destacando que se produce de forma lineal y ascendente conforme aumenta el número de tiempos de proceso que son ceros. De igual forma destacar que el algoritmo NEH presenta unas desviaciones en sus resultados que llegan a ser la mitad que las obtenidas por las otras dos técnicas.

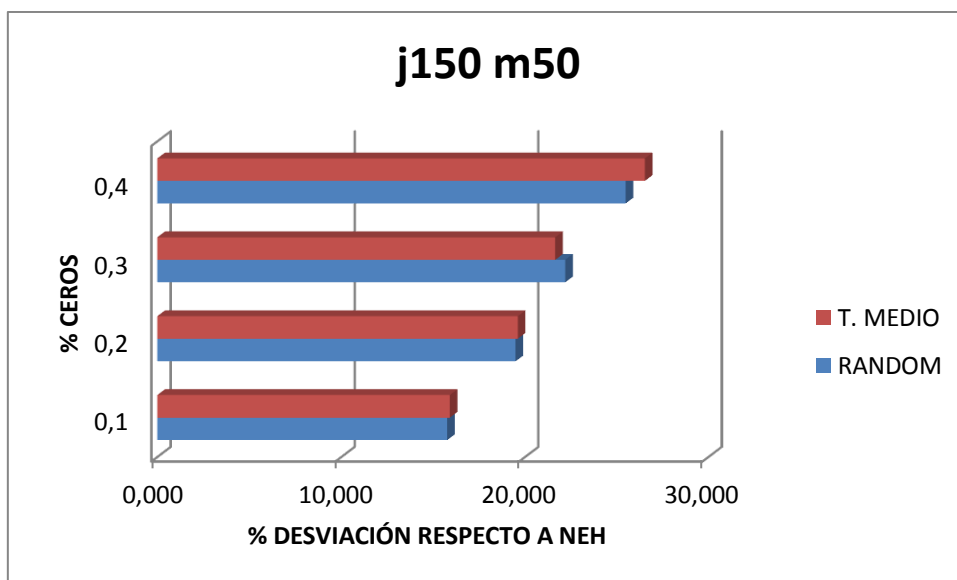


Figura 53: Porcentaje de mejora respecto a NEH de los algoritmos Random y T. Medios frente al porcentaje de ceros para el experimento de 150 trabajos y 50 máquinas.

La figura 53 refleja como el algoritmo Random y el algoritmo basado en los tiempos medios obtiene peores resultados o similares acorde al aumento del porcentaje de tiempos de procesos que son ceros.

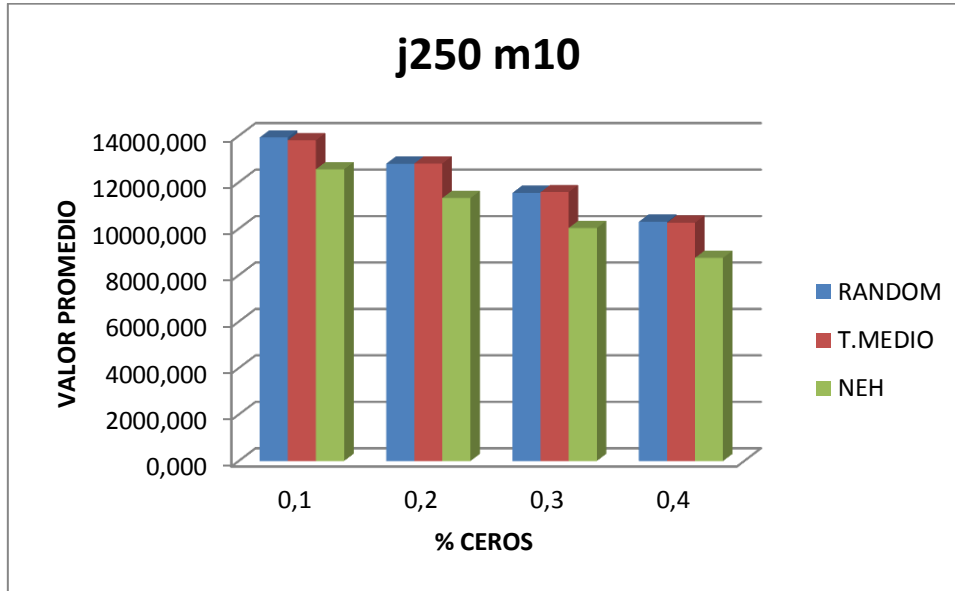


Figura 54: Valor promedio frente a porcentaje de cero para cada algoritmo del experimento de 250 trabajos y 10 máquinas.

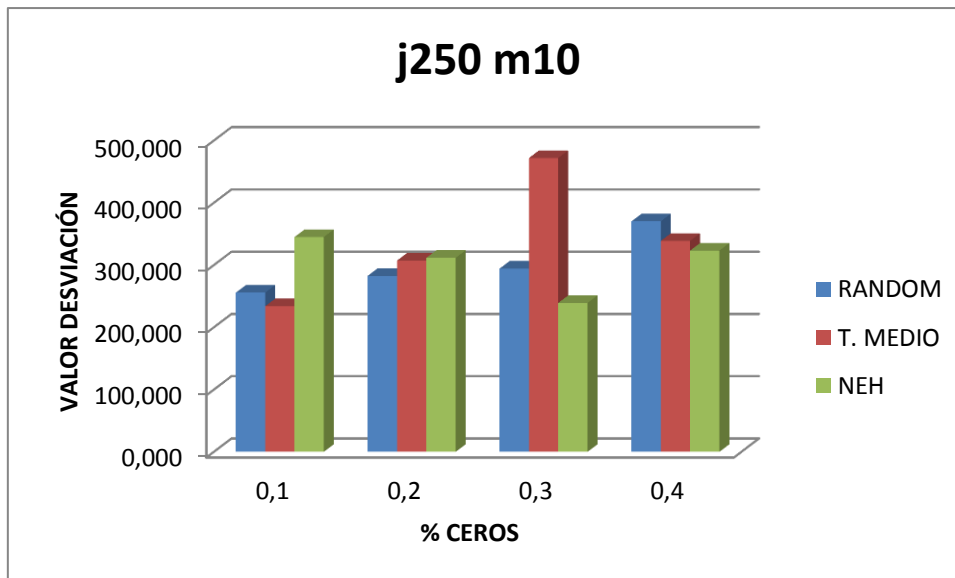


Figura 55: Valor desviación típica frente a porcentaje de cero para cada algoritmo del experimento de 250 trabajos y 10 máquinas.

Se puede observar en la figura 55 como la desviación de los resultados que se obtiene por el algoritmo NEH alcanza unos valores significativos, puesto que llega a

presentar una dispersión en sus resultados más alta que para el resto de algoritmo. Ésto sucede para el caso en el que el 10% de los tiempos de proceso son cero, para el caso de 20% se iguala al algoritmo tiempo medio y para 30% y 40% llega a ser el de menor desviación.

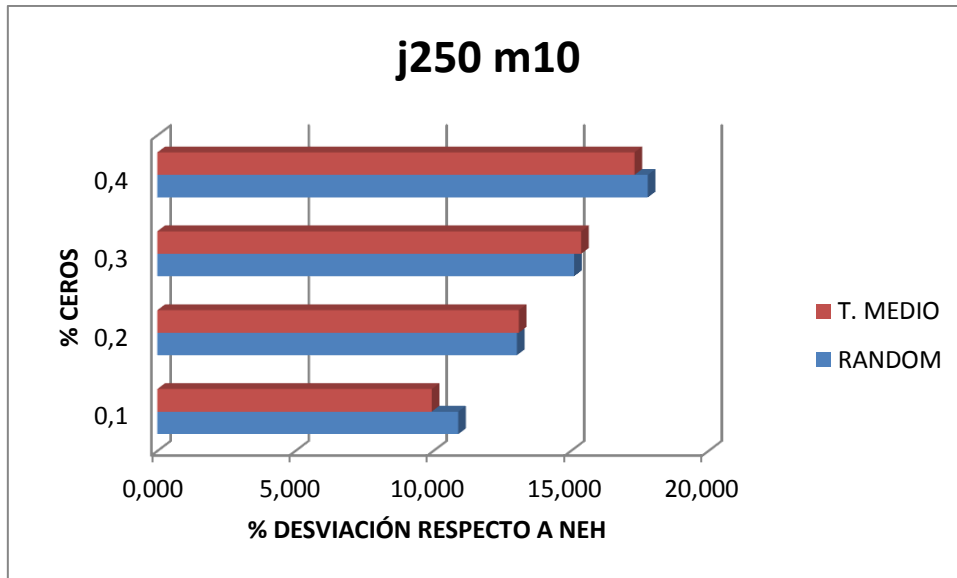


Figura 56: Porcentaje de mejora respecto a NEH de los algoritmos Random y T. Medios frente al porcentaje de ceros para el experimento de 250 trabajos y 10 máquinas.

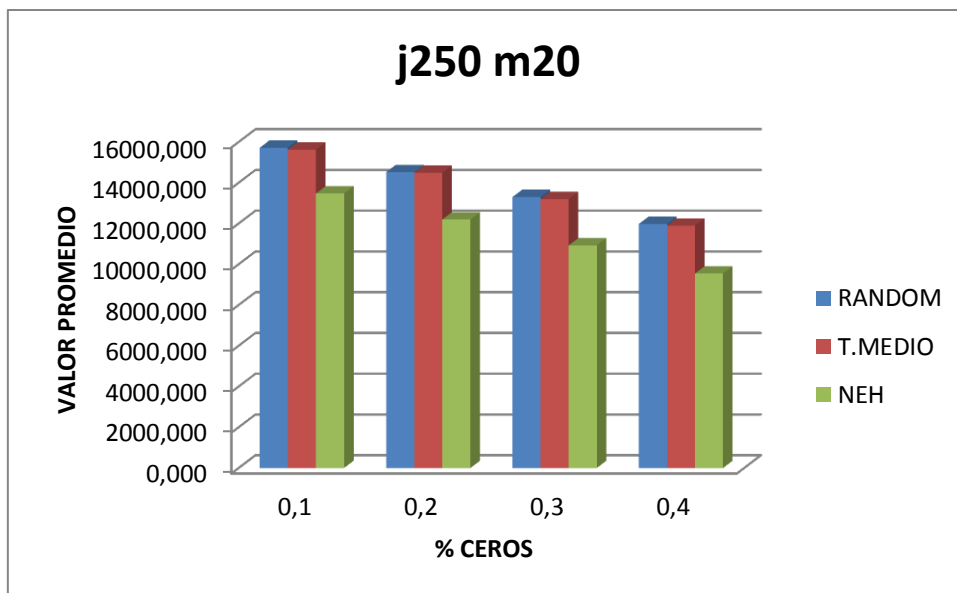


Figura 57: Valor promedio frente a porcentaje de cero para cada algoritmo del experimento de 250 trabajos y 20 máquinas.

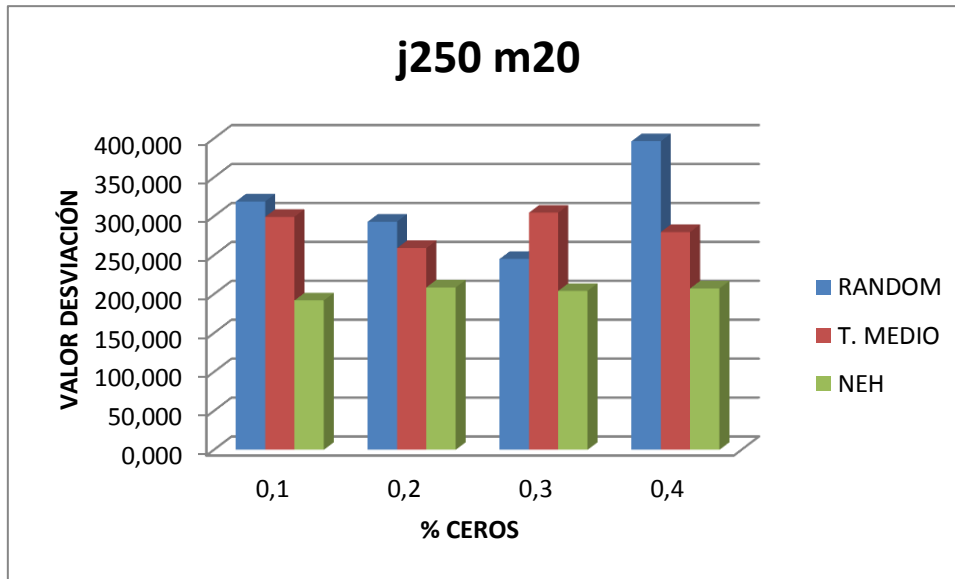


Figura 58: Valor desviación típica frente a porcentaje de cero para cada algoritmo del experimento de 250 trabajos y 20 máquinas.

La figura 58 muestra los valores de desviación típica obtenidos para el experimento de 250 trabajos y 20 máquinas. En este caso, es significativo el valor de desviación que alcanza el algoritmo Random cuando el 40% de los tiempos de proceso son cero. En lo demás presenta el mismo patrón de observaciones que se han comentado con anterioridad.

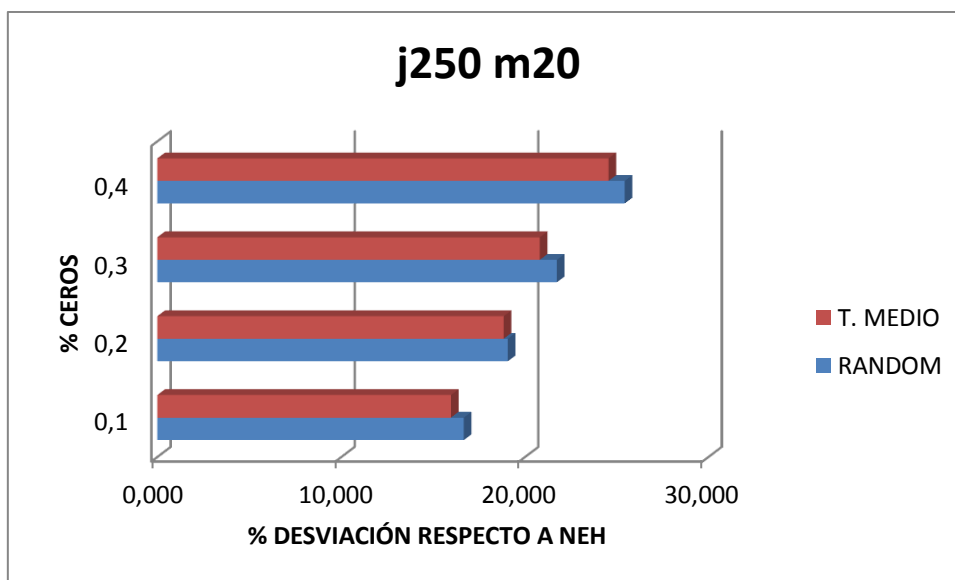


Figura 59: Porcentaje de mejora respecto a NEH de los algoritmos Random y T. Medios frente al porcentaje de ceros para el experimento de 250 trabajos y 20 máquinas.

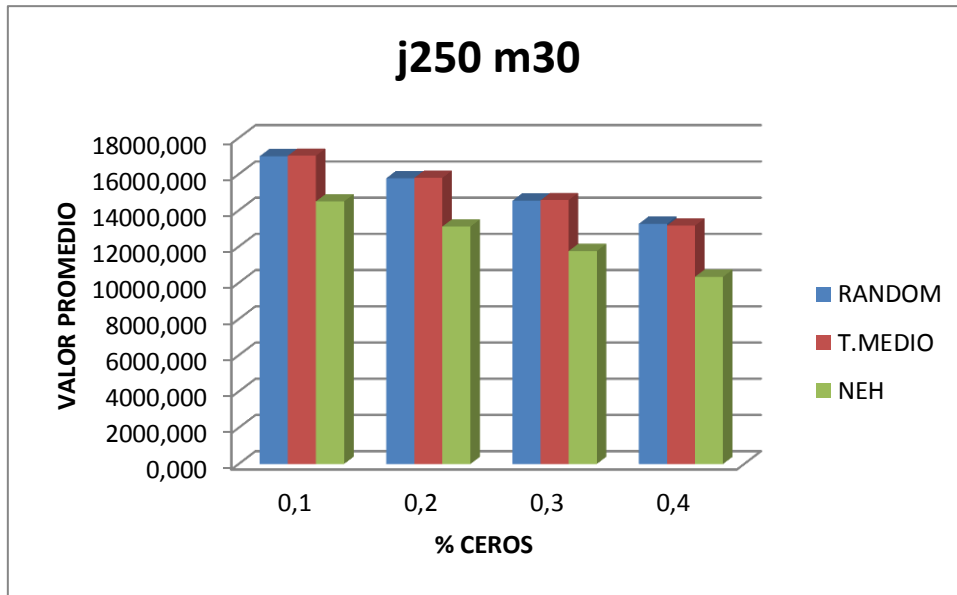


Figura 60: Valor promedio frente a porcentaje de cero para cada algoritmo del experimento de 250 trabajos y 30 máquinas.

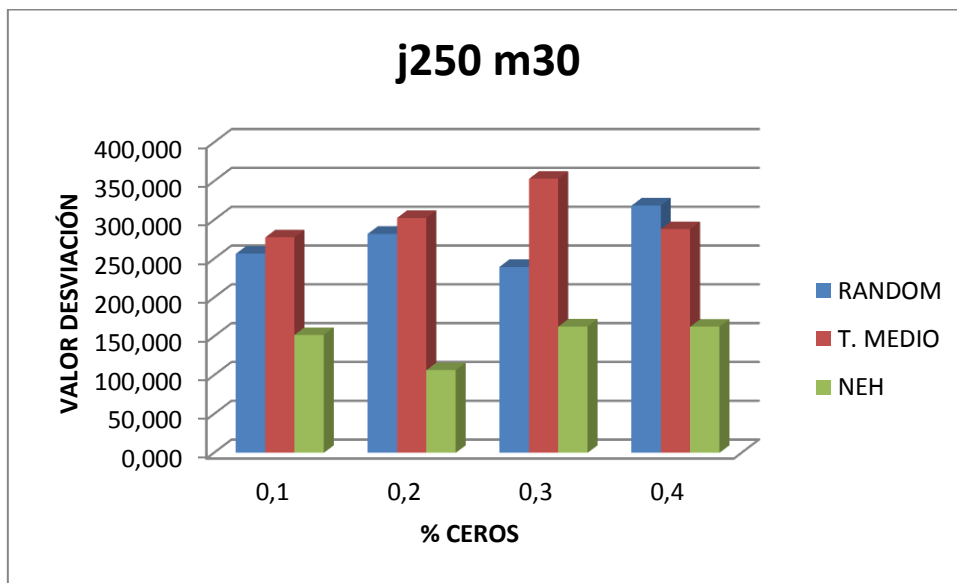


Figura 61: Valor desviación típica frente a porcentaje de cero para cada algoritmo del experimento de 250 trabajos y 30 máquinas.

Se observa en la figura 61, de la misma forma que sucediera en el experimento para 150 trabajos y 50 máquinas, que el algoritmo de los tiempos medios presenta en todos los experimentos la mayor desviación a excepción del caso en el que el 40% de los tiempos de proceso son cero. Destacar que el algoritmo NEH presenta unas

desviaciones en sus resultados que llegan a ser la mitad que las obtenidas por las otras dos técnicas.

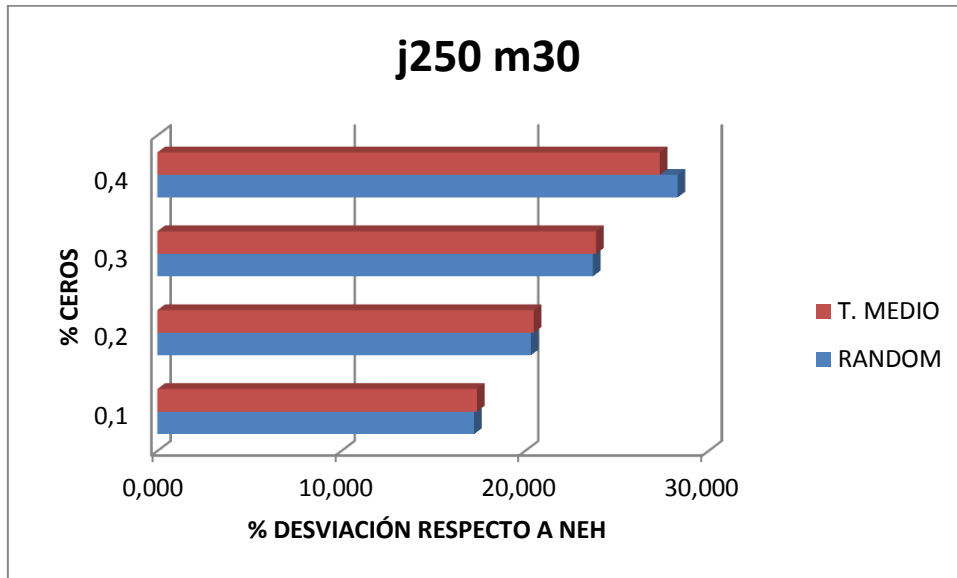


Figura 62: Porcentaje de mejora respecto a NEH de los algoritmos Random y T. Medio frente al porcentaje de ceros para el experimento de 250 trabajos y 30 máquinas.

Destacar, una vez observado la figura 62, que el porcentaje de error que muestran los dos algoritmos respecto a NEH son muy semejantes para los experimentos tratados en 50 trabajos.

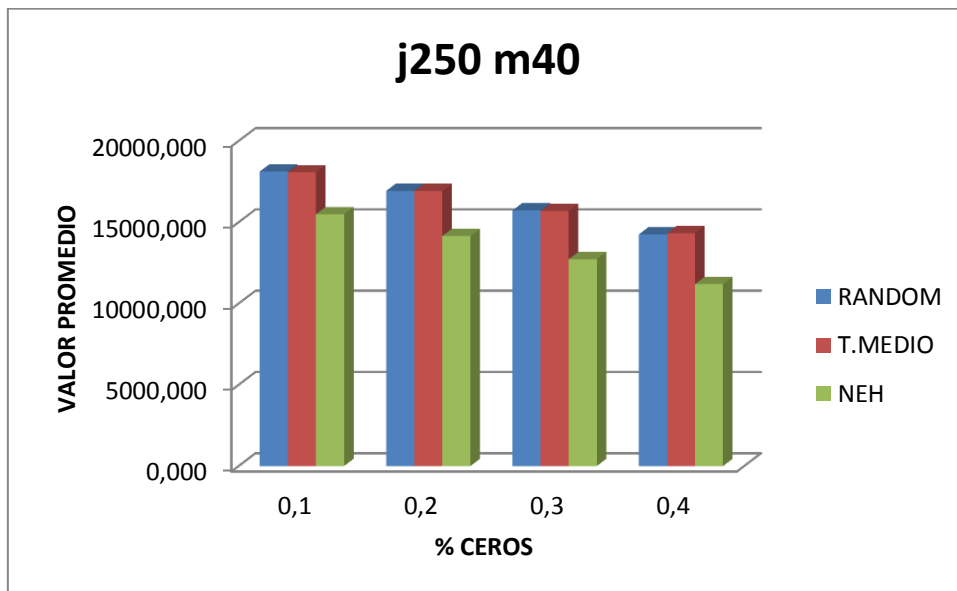


Figura 63: Valor promedio frente a porcentaje de cero para cada algoritmo del experimento de 250 trabajos y 40 máquinas.

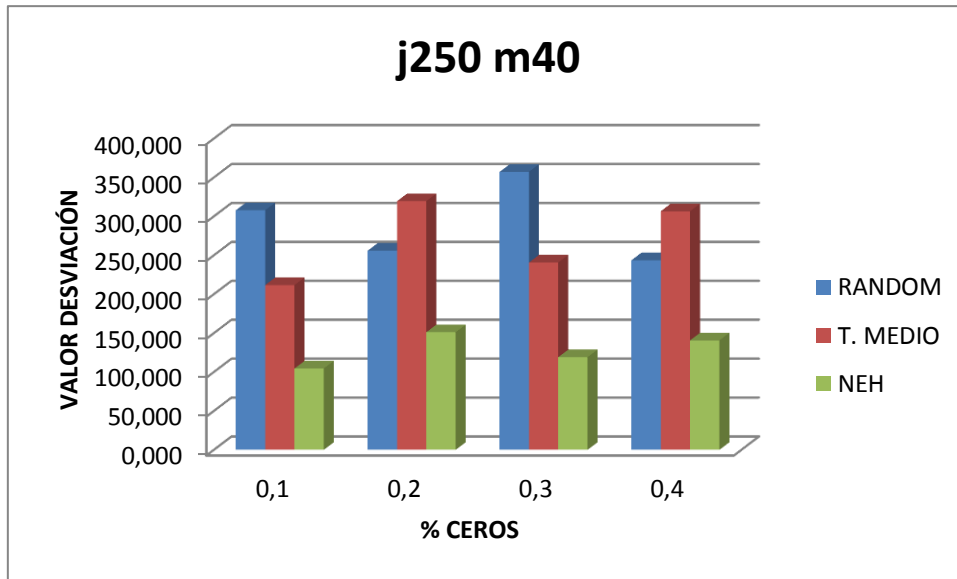


Figura 64: Valor desviación típica frente a porcentaje de cero para cada algoritmo del experimento de 250 trabajos y 40 máquinas.

En la figura 64 se observa una alternancia en los valores máximos de desviación para cada experimento entre los algoritmos Random y tiempos medios.

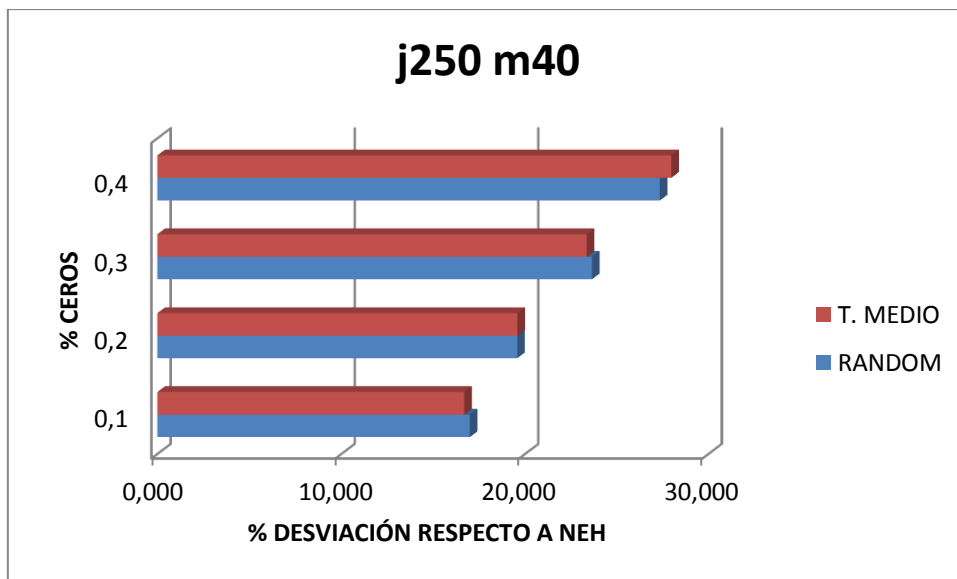


Figura 65: Porcentaje de mejora respecto a NEH de los algoritmos Random y T. Medio frente al porcentaje de ceros para el experimento de 250 trabajos y 40 máquinas.

Presenta el mismo análisis que en la figura 62.

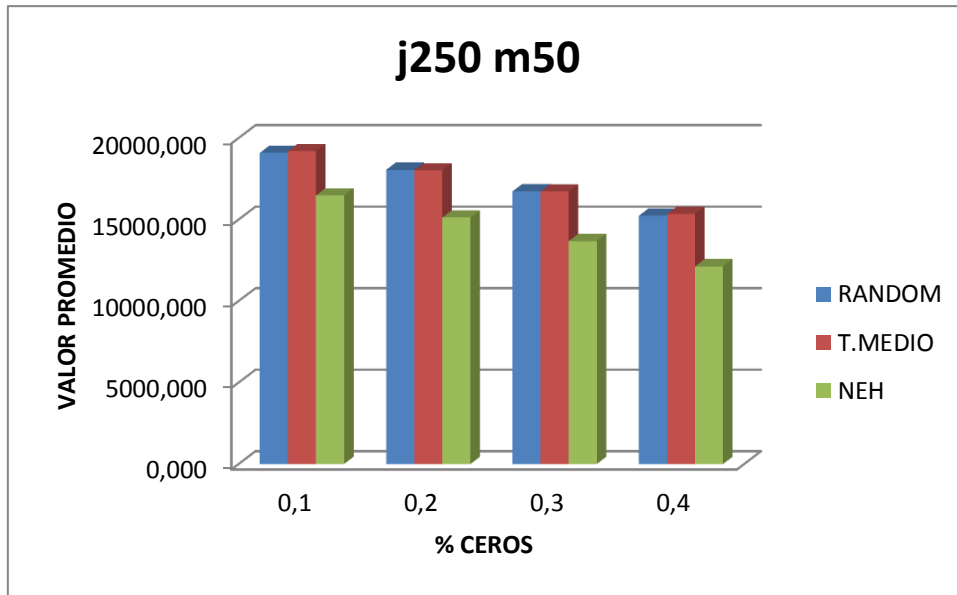


Figura 66: Valor promedio frente a porcentaje de cero para cada algoritmo del experimento de 250 trabajos y 50 máquinas.

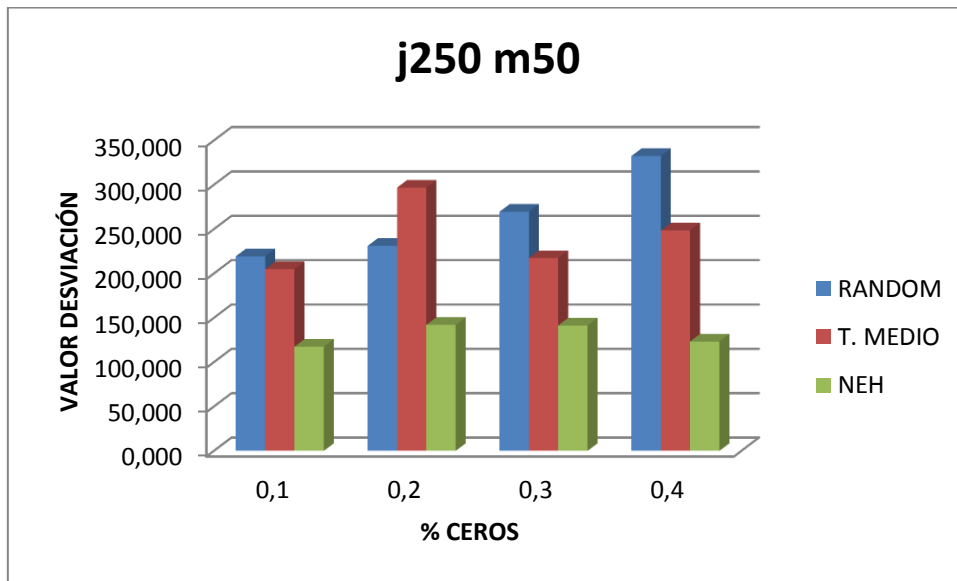


Figura 67: Valor desviación típica frente a porcentaje de cero para cada algoritmo del experimento de 250 trabajos y 50 máquinas.

La figura 67 refleja las mismas observaciones citadas con anterioridad que presentan este tipo de gráficos.

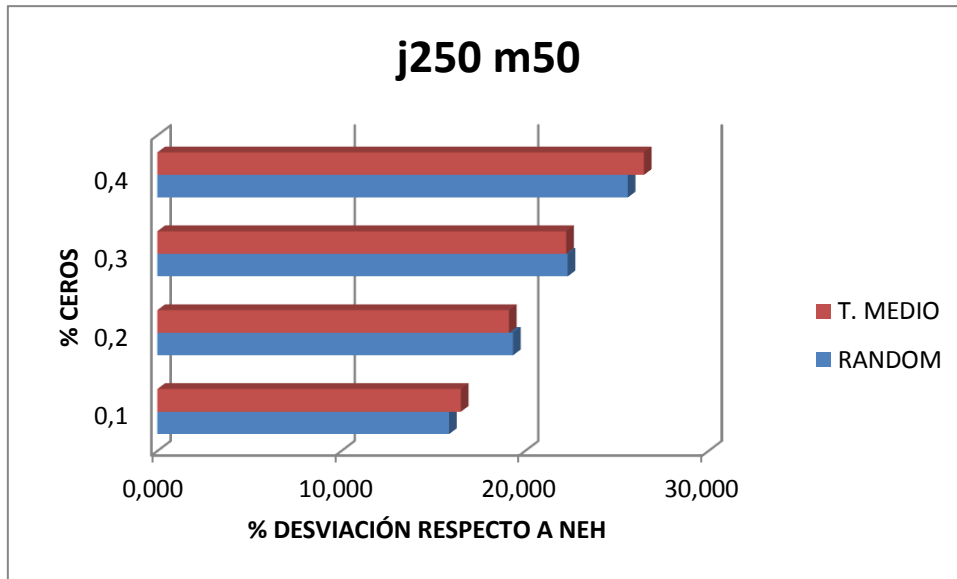


Figura 68: Porcentaje de mejora respecto a NEH de los algoritmos Random y T. Medio frente al porcentaje de ceros para el experimento de 250 trabajos y 50 máquinas.

Se observa en la figura 68, y al igual que durante todo el experimento tratado con 250 trabajos, que el porcentaje de error que muestran los dos algoritmos respecto a NEH son muy similares.

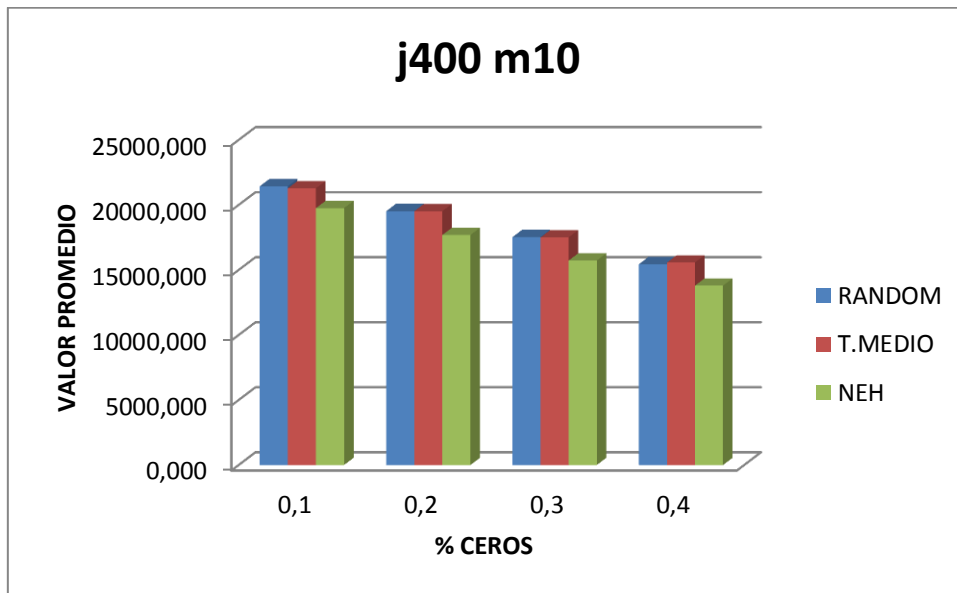


Figura 69: Valor promedio frente a porcentaje de cero para cada algoritmo del experimento de 400 trabajos y 10 máquinas.

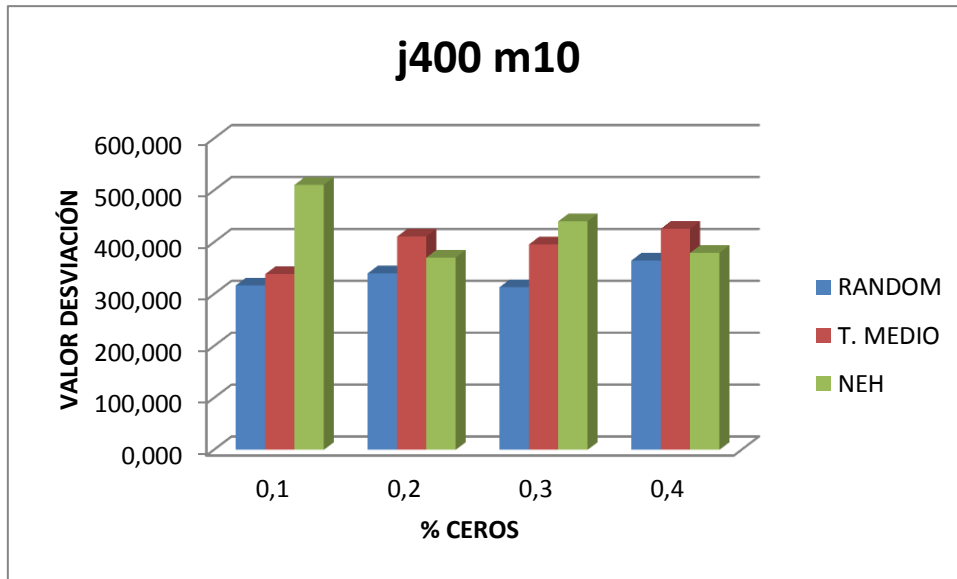


Figura 70: Valor desviación típica frente a porcentaje de cero para cada algoritmo del experimento de 400 trabajos y 10 máquinas.

Como se puede observar, la figura 70 muestra como para este experimento de 400 trabajos y 10 máquinas el algoritmo basado en la heurística NEH presenta una desviación mayor tanto para el 10% de los tiempos de proceso que son ceros como para el 30%. Esto puede ser debido al incremento de trabajos cuya influencia se ha visto reflejada en la dispersión de los resultados que obtiene.

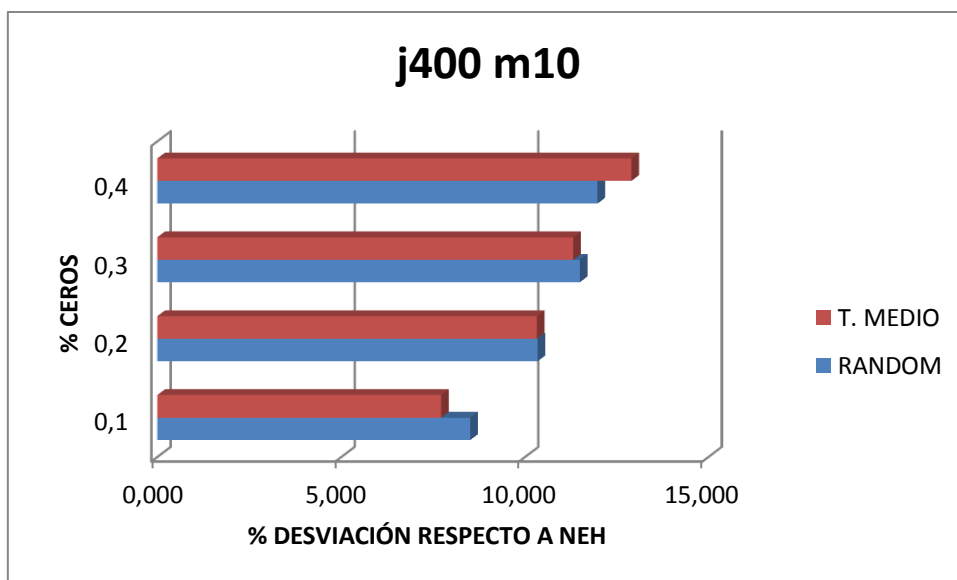


Figura 71: Porcentaje de mejora respecto a NEH de los algoritmos Random y T. Medio frente al porcentaje de ceros para el experimento de 400 trabajos y 10 máquinas.

La figura 71 refleja, al igual que ocurriera en la figura 45 para el caso de 150 trabajos y 10 máquinas, que el algoritmo de los tiempos medios obtiene una secuencia de trabajos peores o similares, en la mayoría de los casos, que el algoritmo que consigue la secuencia de forma aleatoria.

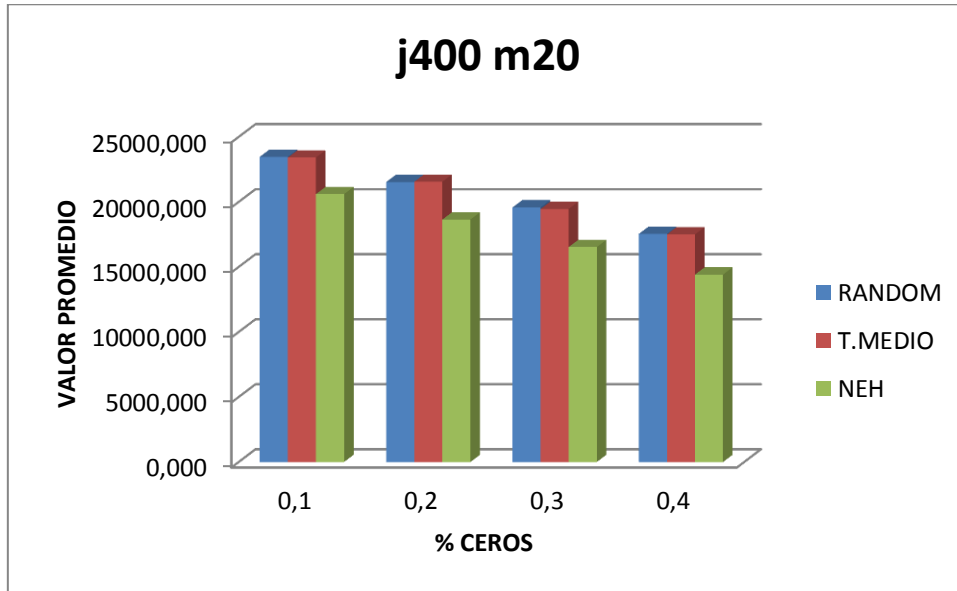


Figura 72: Valor promedio frente a porcentaje de cero para cada algoritmo del experimento de 400 trabajos y 20 máquinas.

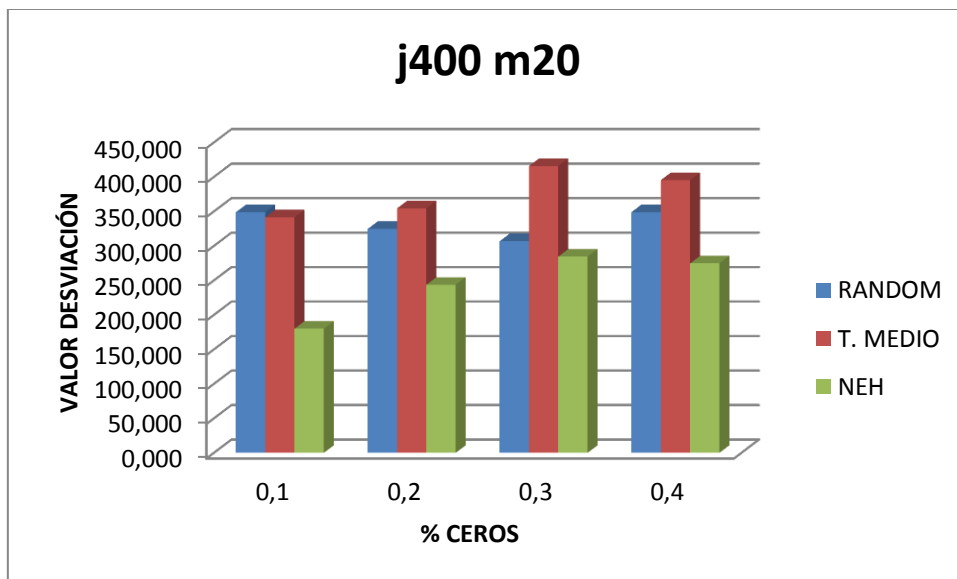


Figura 73: Valor desviación típica frente a porcentaje de cero para cada algoritmo del experimento de 400 trabajos y 20 máquinas.

Para esta ocasión y como bien muestra la figura 73 la desviación del algoritmo NEH vuelve a ser la menor y se incrementa conforme aumenta el porcentaje de tiempos que son ceros. Otro aspecto a señalar son los valores tan elevados que se reflejan para el algoritmo de los tiempos medios.

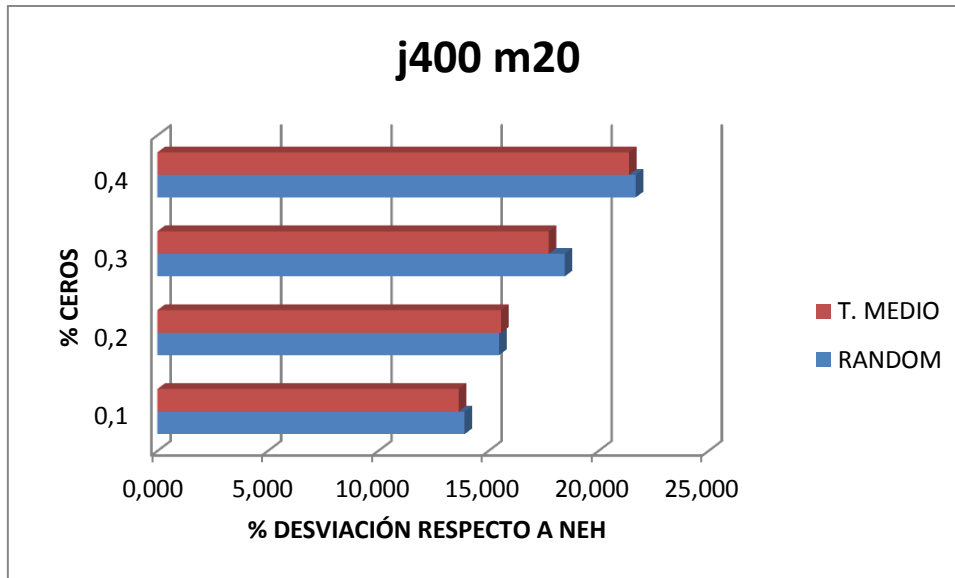


Figura 74: Porcentaje de mejora respecto a NEH de los algoritmos Random y T. Medio frente al porcentaje de ceros para el experimento de 400 trabajos y 20 máquinas.

La figura 74 refleja como el algoritmo Random obtiene peores resultados que el algoritmo basado en los tiempos medios acorde al aumento del porcentaje de tiempos de procesos que son ceros.

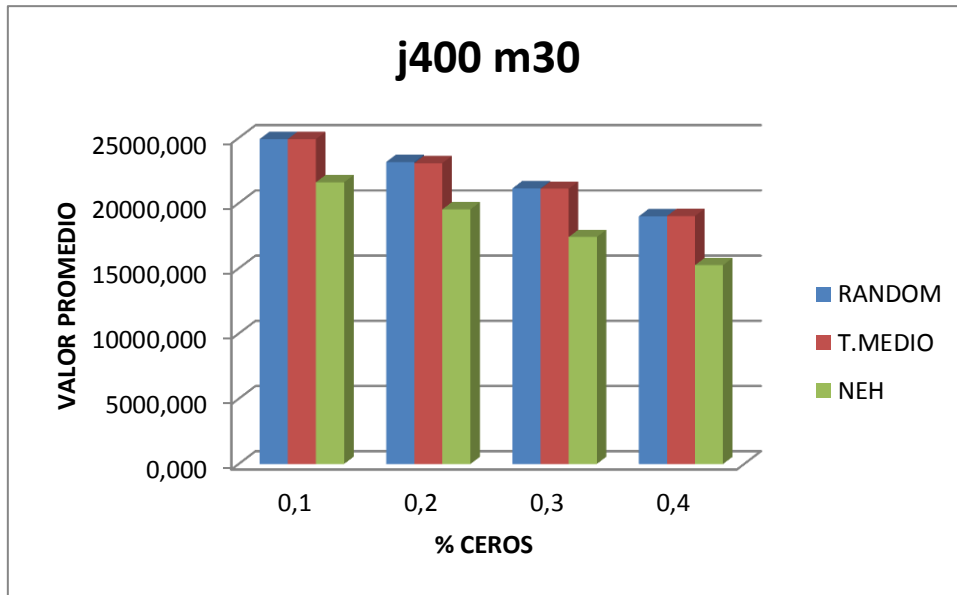


Figura 75: Valor promedio frente a porcentaje de cero para cada algoritmo del experimento de 400 trabajos y 30 máquinas.

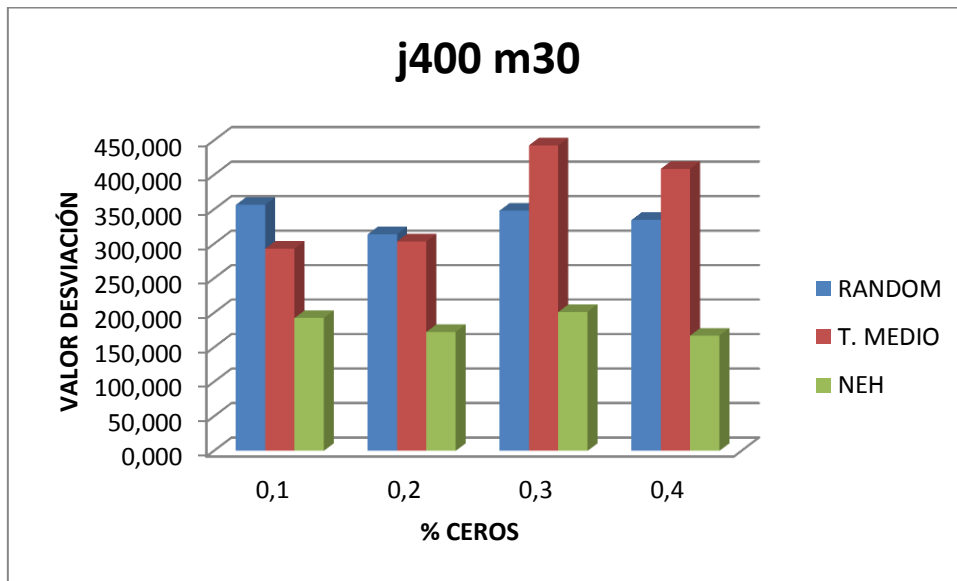


Figura 76: Valor desviación típica frente a porcentaje de cero para cada algoritmo del experimento de 400 trabajos y 30 máquinas.

La figura 76 refleja como el algoritmo de los tiempos medios vuelve a ser el que presente el valor más elevado para los casos de 30% y 40% de los tiempos de proceso iguales a cero. Por lo que se repite y continúa la misma línea de los últimos experimentos. El algoritmo NEH vuelve a mantenerse constante en torno a un valor

para todos los experimentos que abarcan este problema de 400 trabajos y 30 máquinas.

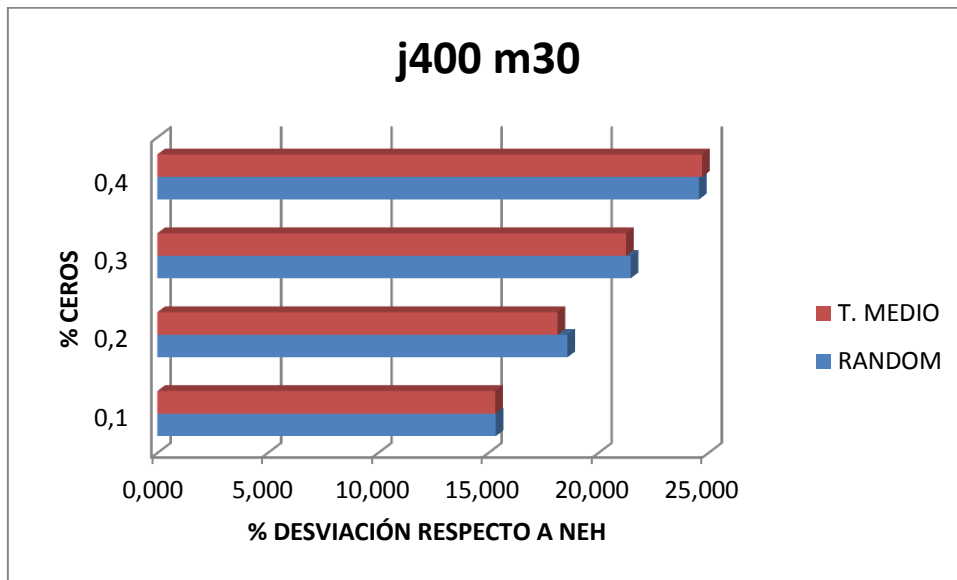


Figura 77: Porcentaje de mejora respecto a NEH de los algoritmos Random y T. Medio frente al porcentaje de ceros para el experimento de 400 trabajos y 30 máquinas.

La figura 77 no presenta ninguna observación a destacar respecto a otros análisis que ya se hayan hecho anteriormente.

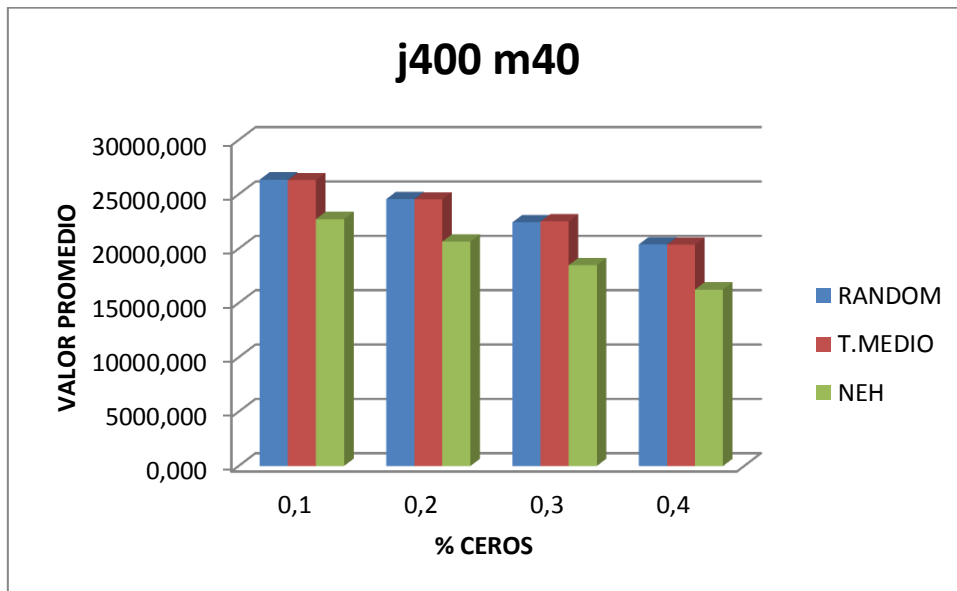


Figura 78: Valor promedio frente a porcentaje de cero para cada algoritmo del experimento de 400 trabajos y 40 máquinas.

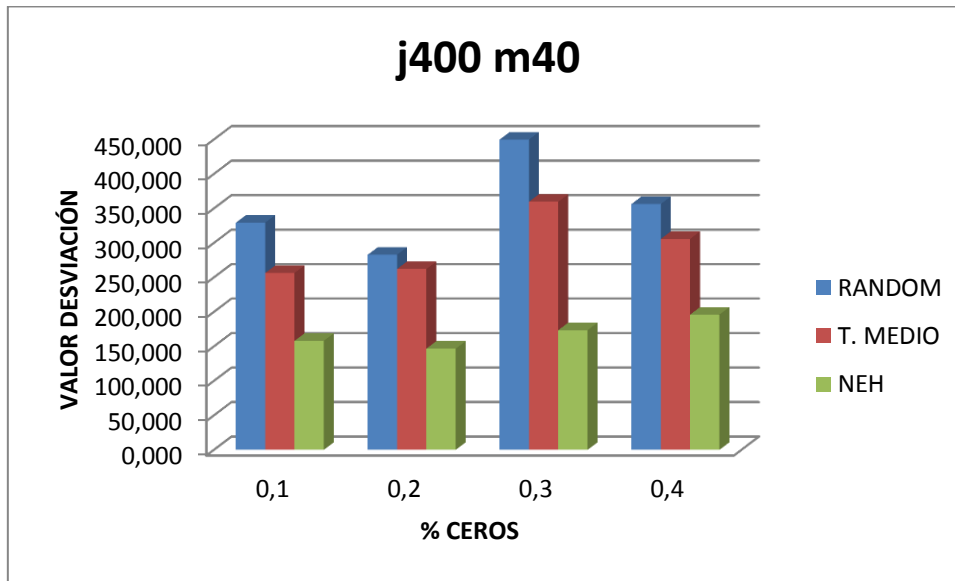


Figura 79: Valor desviación típica frente a porcentaje de cero para cada algoritmo del experimento de 400 trabajos y 40 máquinas.

La figura 79 refleja como el algoritmo Random obtiene mayores desviaciones que los otros dos algoritmos utilizados en este estudio, igual que se ha citado en otros análisis anteriores.

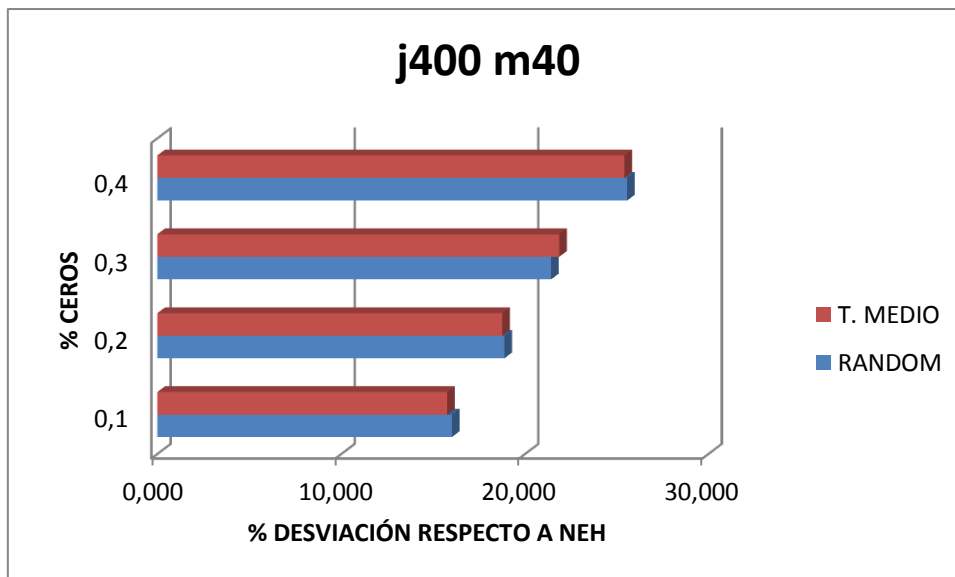


Figura 80: Porcentaje de mejora respecto a NEH de los algoritmos Random y T. Medio frente al porcentaje de ceros para el experimento de 400 trabajos y 40 máquinas.

Se observa en la figura 80, de la misma forma que durante todo el experimento tratado con 400 trabajos, que el porcentaje de error que muestran los dos algoritmos respecto a NEH son muy similares.

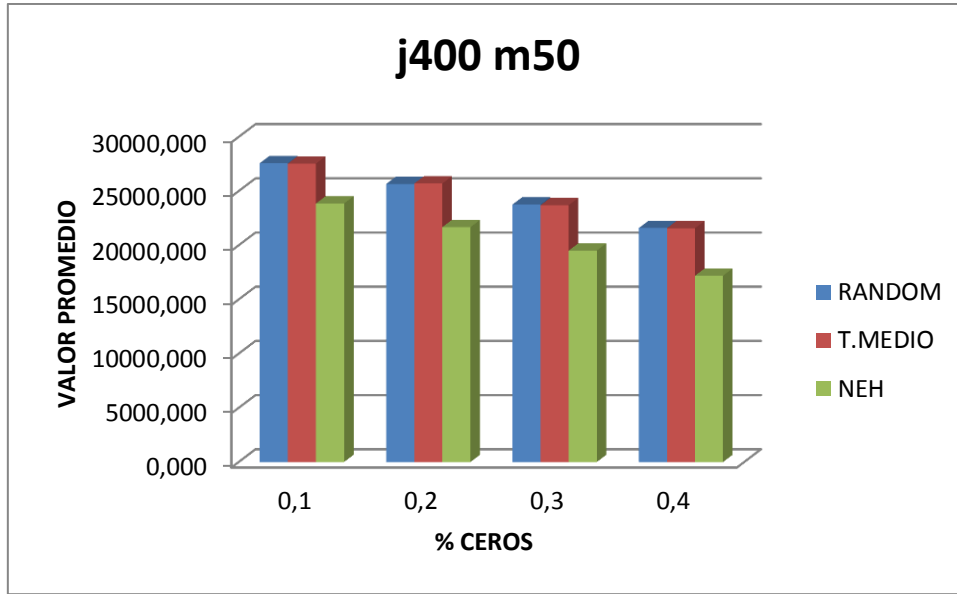


Figura 81: Valor promedio frente a porcentaje de cero para cada algoritmo del experimento de 400 trabajos y 50 máquinas.

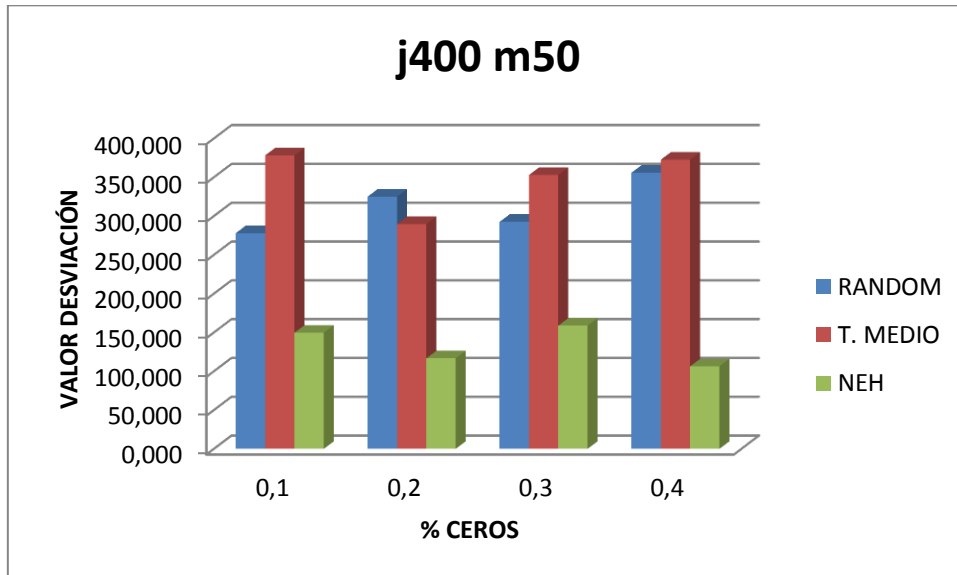


Figura 82: Valor desviación típica frente a porcentaje de cero para cada algoritmo del experimento de 400 trabajos y 50 máquinas.

En la figura 82, se reflejan los valores de la desviación típica alcanzada para los diferentes algoritmos. Se puede observar que el algoritmo basado en ordenar por tiempos medios alcanza una mayor desviación típica en sus valores y que el algoritmo NEH presenta la menor desviación en sus resultados, siendo éstos últimos aproximadamente la mitad de los valores que alcanzan el resto de algoritmos.

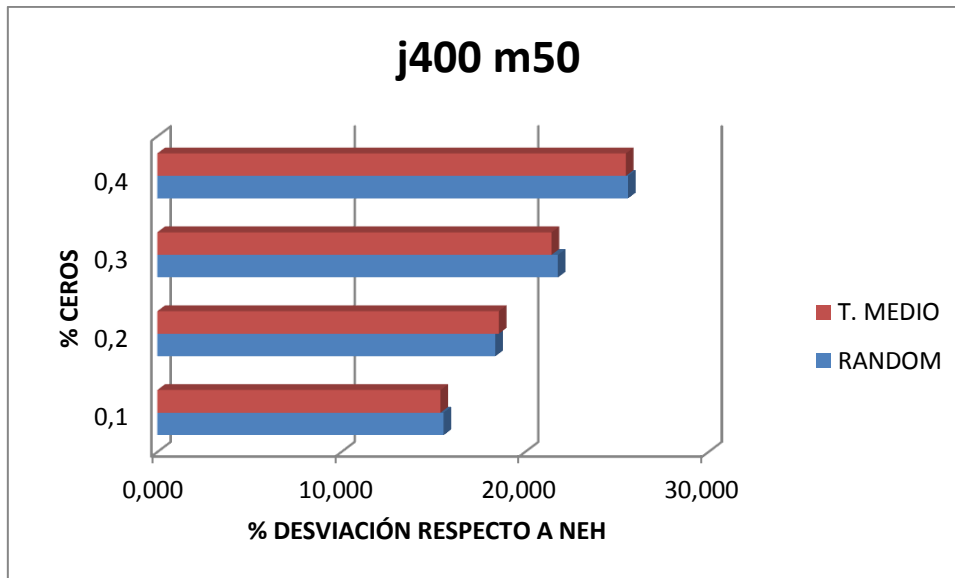


Figura 83: Porcentaje de mejora respecto a NEH de los algoritmos Random y T. Medio frente al porcentaje de ceros para el experimento de 400 trabajos y 50 máquinas.

Destacar, una vez observado la figura 83, que el porcentaje de error que muestran los dos algoritmos respecto a NEH son muy semejantes para los experimentos tratados cuando se tienen 400 trabajos.

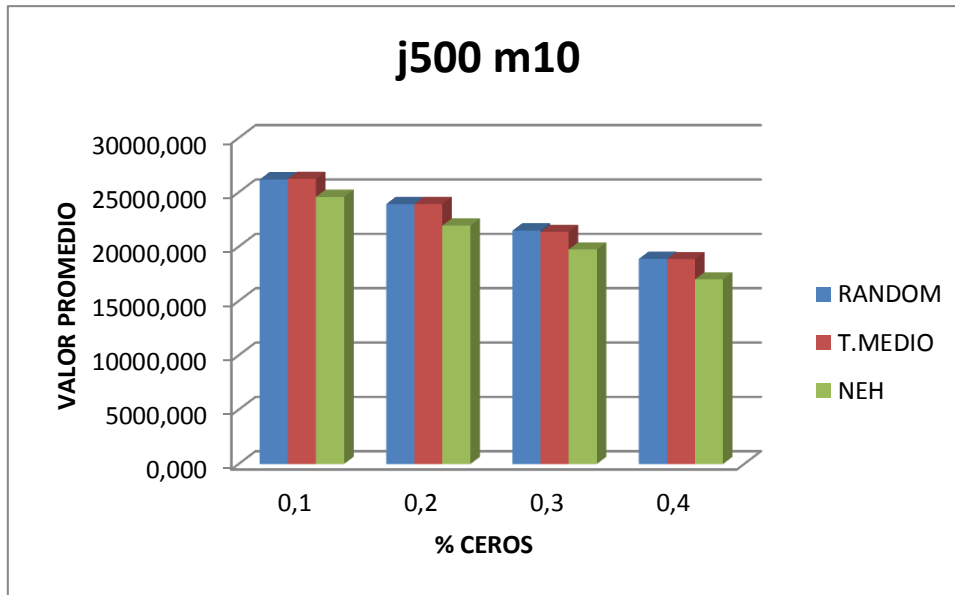


Figura 84: Valor promedio frente a porcentaje de cero para cada algoritmo del experimento de 500 trabajos y 10 máquinas.

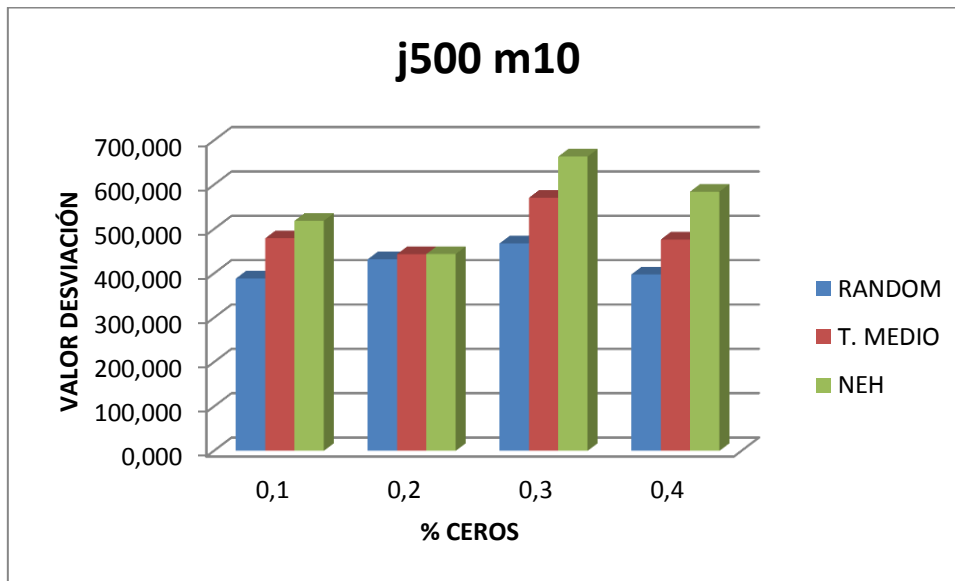


Figura 85: Valor desviación típica frente a porcentaje de cero para cada algoritmo del experimento de 500 trabajos y 10 máquinas.

Como se puede observar, la figura 85 muestra como para este experimento de 500 trabajos y 10 máquinas el algoritmo basado en la heurística NEH presenta una desviación mayor para todos los experimentos. Esto puede ser debido al incremento de trabajos cuya influencia se ha visto reflejada en mayor medida en este algoritmo expresado en la dispersión de los resultados que obtiene.

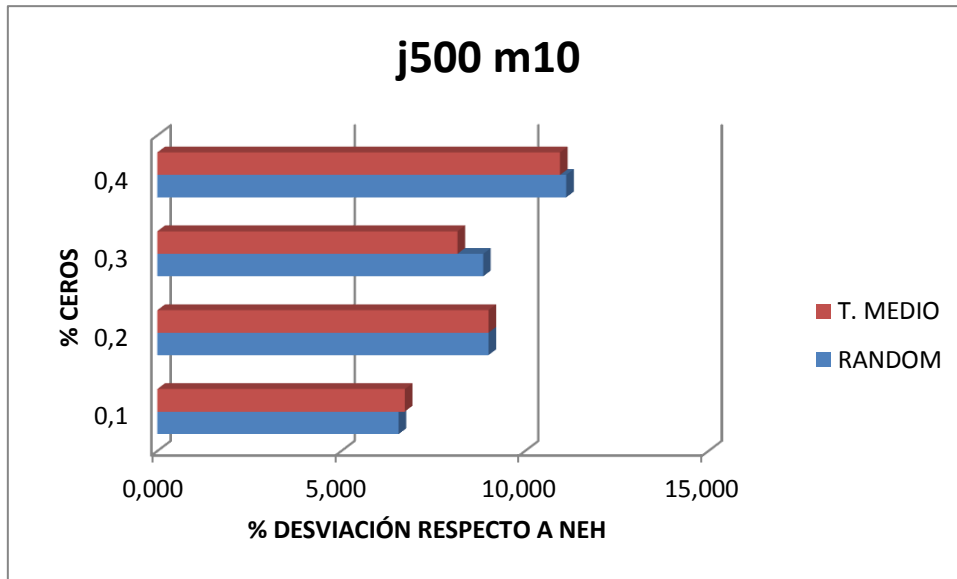


Figura 86: Porcentaje de mejora respecto a NEH de los algoritmos Random y T. Medio frente al porcentaje de ceros para el experimento de 500 trabajos y 10 máquinas.

Destacar en la figura 86 y como se puede observar también en la figura 84 que la secuencia solución obtenida por NEH no difiere, como en otros experimentos hasta ahora analizados, en gran valor respecto a los otros dos algoritmos utilizados en este estudio.

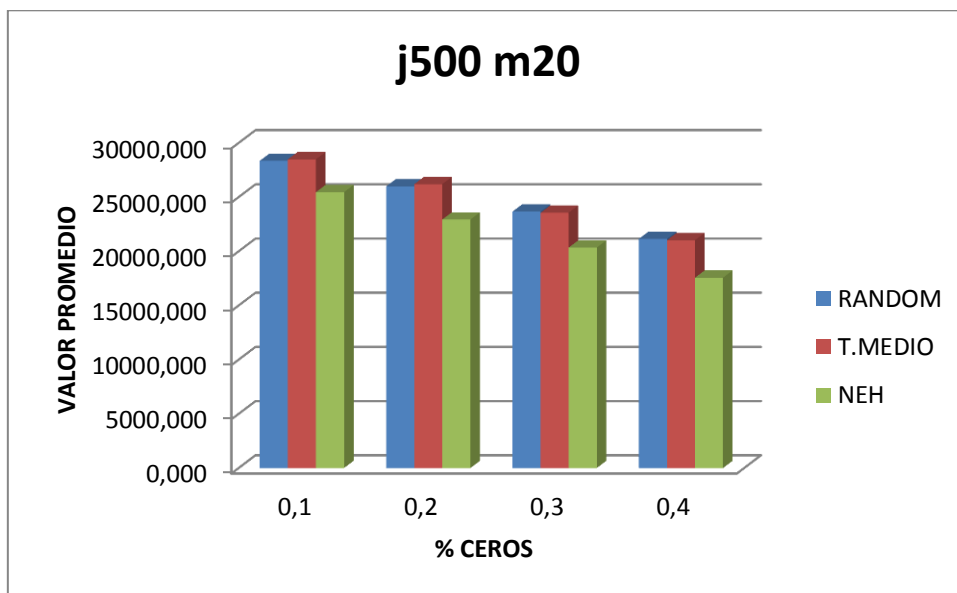


Figura 87: Valor promedio frente a porcentaje de cero para cada algoritmo del experimento de 500 trabajos y 20 máquinas.

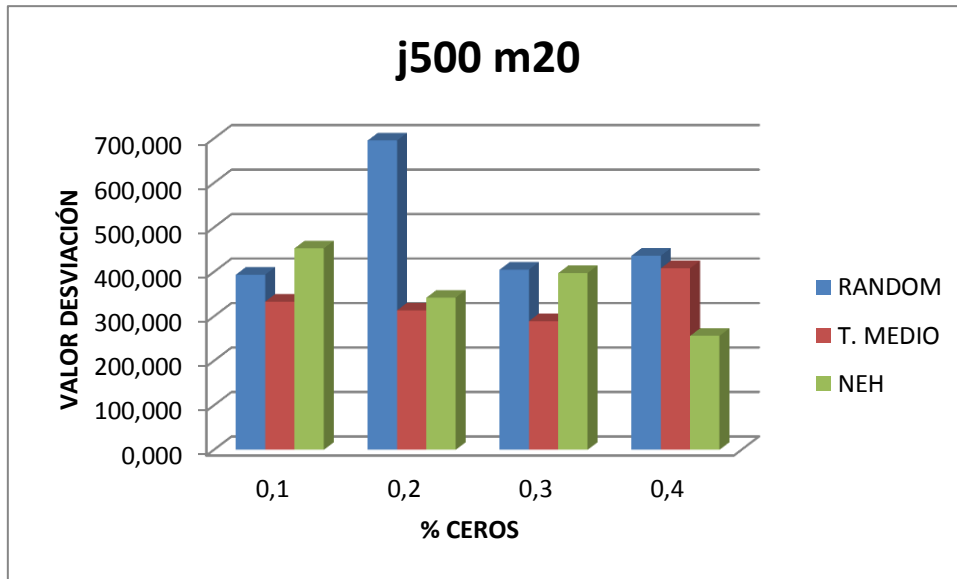


Figura 88: Valor desviación típica frente a porcentaje de cero para cada algoritmo del experimento de 500 trabajos y 20 máquinas.

La figura 88 refleja como el algoritmo Random obtiene una desviación muy elevada respecto a los otros dos algoritmos en el caso en el que el 20% de los tiempos de proceso son cero. Para el caso en el que el 10% de los tiempos de proceso son cero, el que presenta mayor desviación es el algoritmo NEH. Para el caso en el que el 30% de los tiempos de proceso son cero, el algoritmo NEH y el algoritmo Random se igualan.

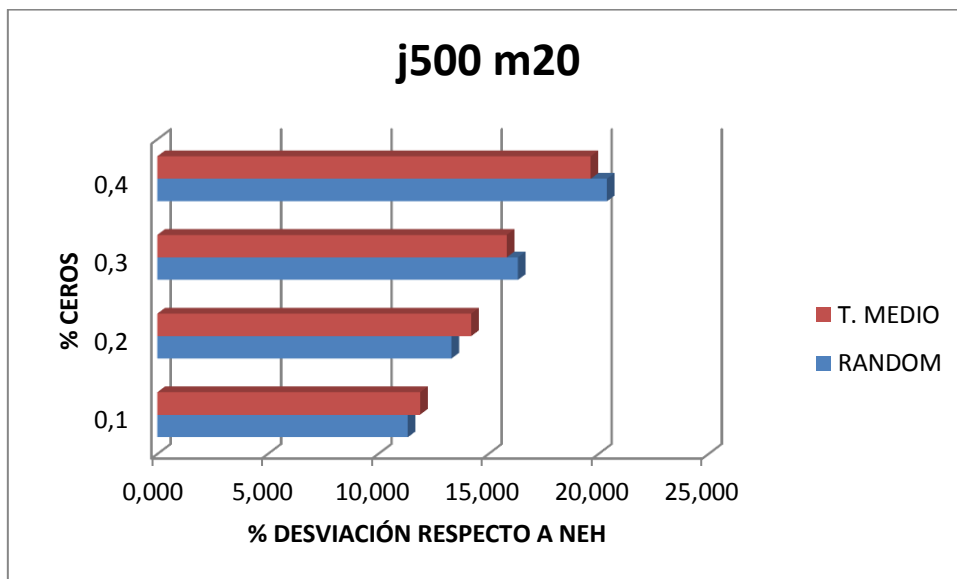


Figura 89: Porcentaje de mejora respecto a NEH de los algoritmos Random y T. Medio frente al porcentaje de ceros para el experimento de 500 trabajos y 20 máquinas.

Se puede observar en la figura 89 como ya se retoman valores más normales de diferencia en la solución entre el algoritmo NEH y los algoritmos Random y tiempos medios.

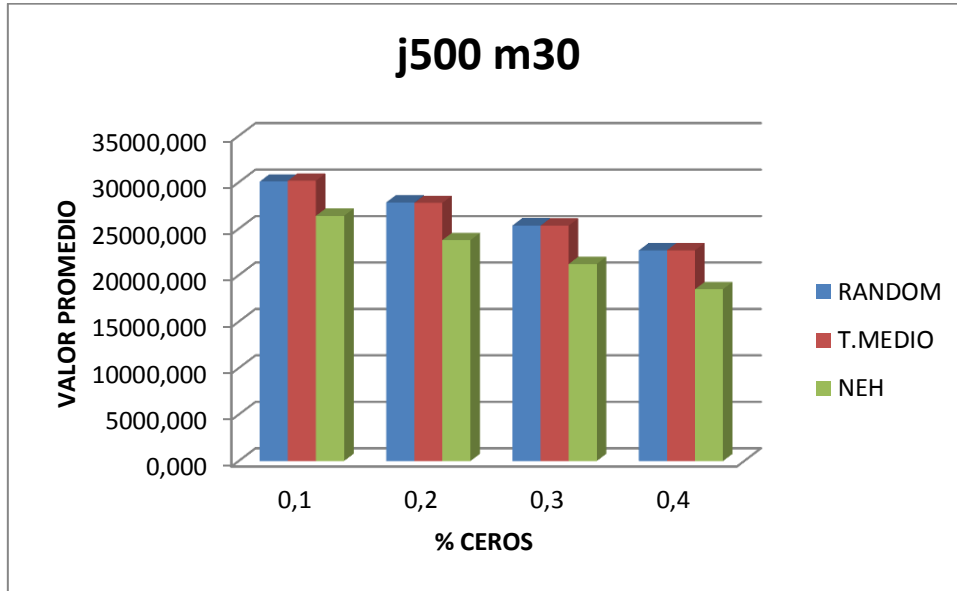


Figura 90: Valor promedio frente a porcentaje de cero para cada algoritmo del experimento de 500 trabajos y 30 máquinas.

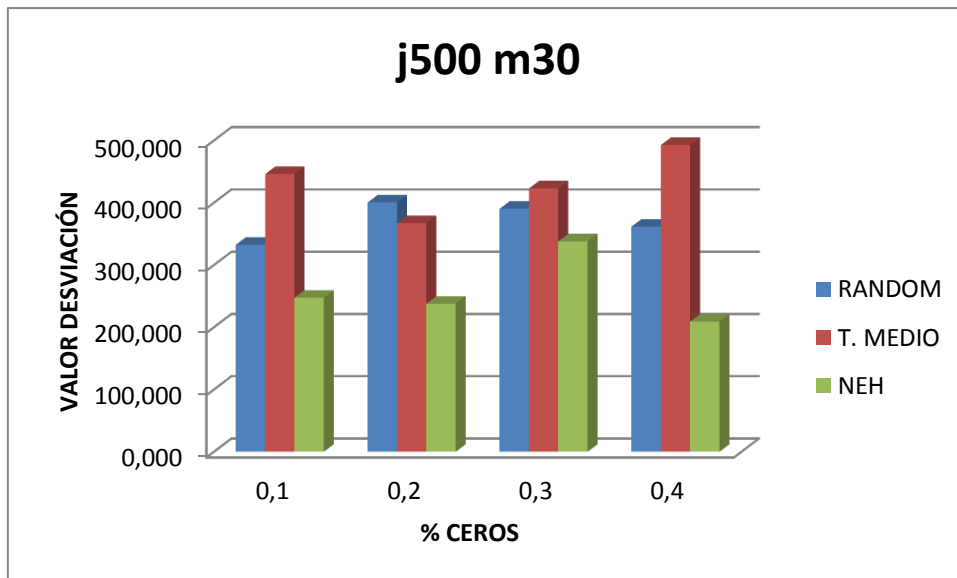


Figura 91: Valor desviación típica frente a porcentaje de cero para cada algoritmo del experimento de 500 trabajos y 30 máquinas.

En la figura 91 no hay alternancia de cara a alcanzar la mayor desviación. Se observa que para los casos 10%, 30% y 40% la desviación es mayor en el algoritmo de los

tiempos medios. Solo para el caso del 20% la mayor desviación se produce en el algoritmo Random.

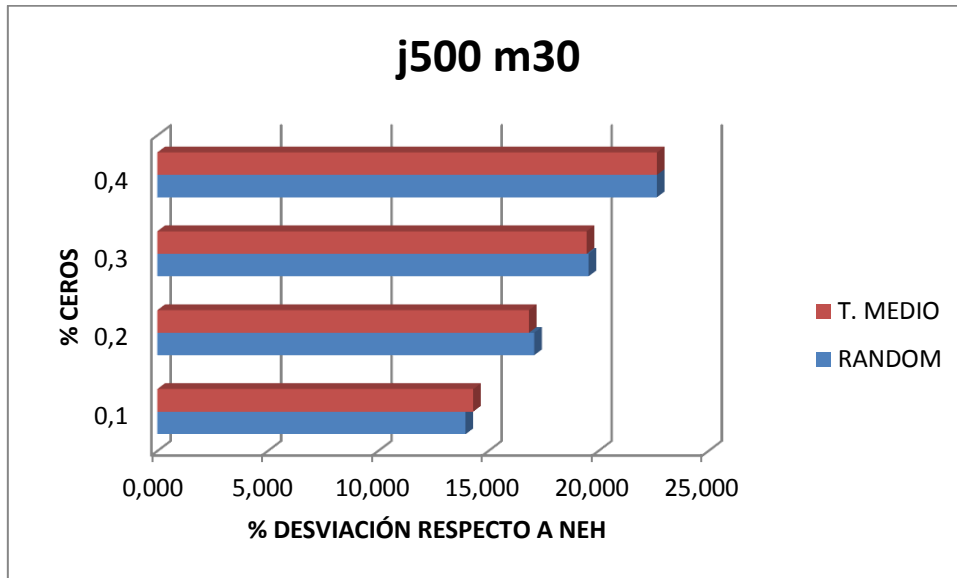


Figura 92: Porcentaje de mejora respecto a NEH de los algoritmos Random y T. Medio frente al porcentaje de ceros para el experimento de 500 trabajos y 30 máquinas.

La figura 92 muestra las mismas observaciones que se ha realizado para este gráfico en anteriores experimentos. Igualdad en el porcentaje de desviación respecto a NEH del resultado obtenido por estos dos algoritmos.

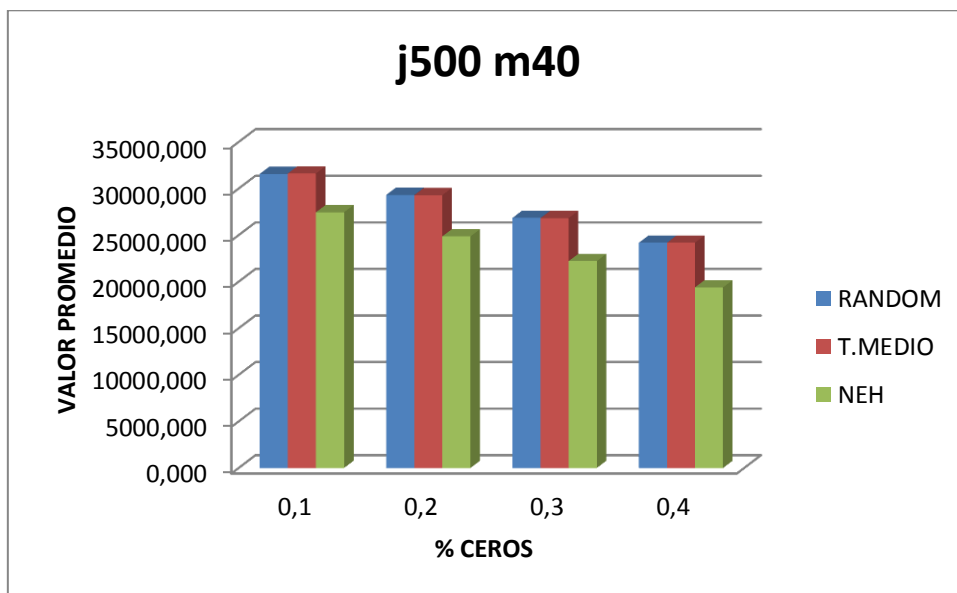


Figura 93: Valor promedio frente a porcentaje de cero para cada algoritmo del experimento de 500 trabajos y 40 máquinas.

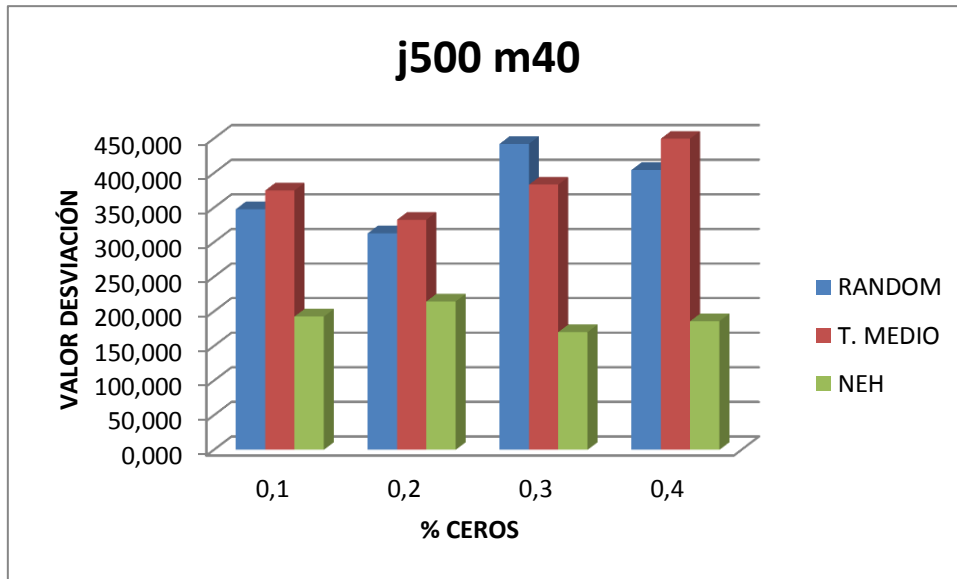


Figura 94: Valor desviación típica frente a porcentaje de cero para cada algoritmo del experimento de 500 trabajos y 40 máquinas.

Se puede observar en la figura 94 como los valores de desviación son elevados, debido al número de trabajos que presenta este experimento. En la mayoría de ellos se muestra el algoritmo de los tiempos medios como el de mayor desviación típica en sus resultados.

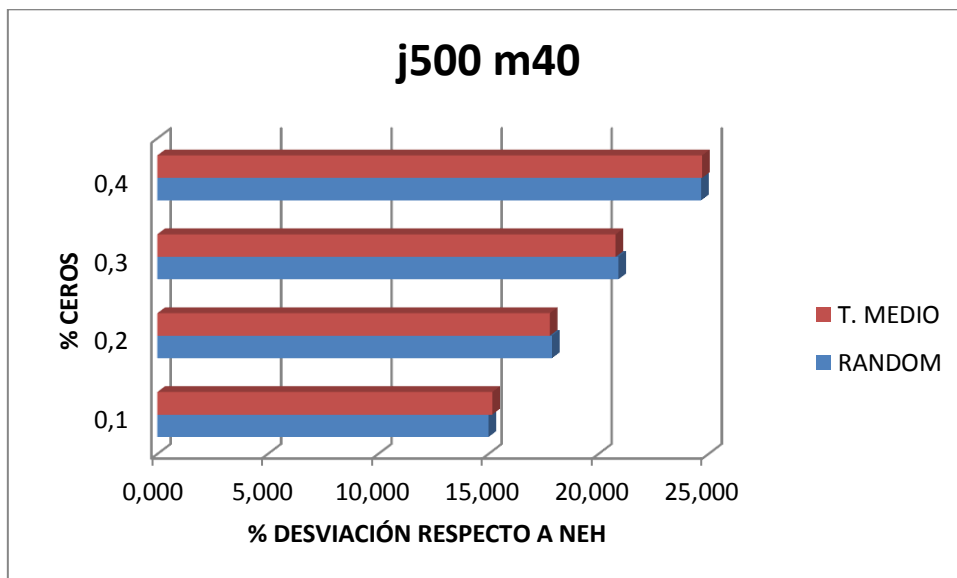


Figura 95: Porcentaje de mejora respecto a NEH de los algoritmos Random y T. Medio frente al porcentaje de ceros para el experimento de 500 trabajos y 40 máquinas.

Se puede observar en la figura 95, destacar que conforme se aumenta el porcentaje de tiempos de procesamiento que son ceros, el error o desviación respecto a la mejor solución aumenta, siendo el algoritmo Random el que presenta mayor desviación en todos sus resultados.

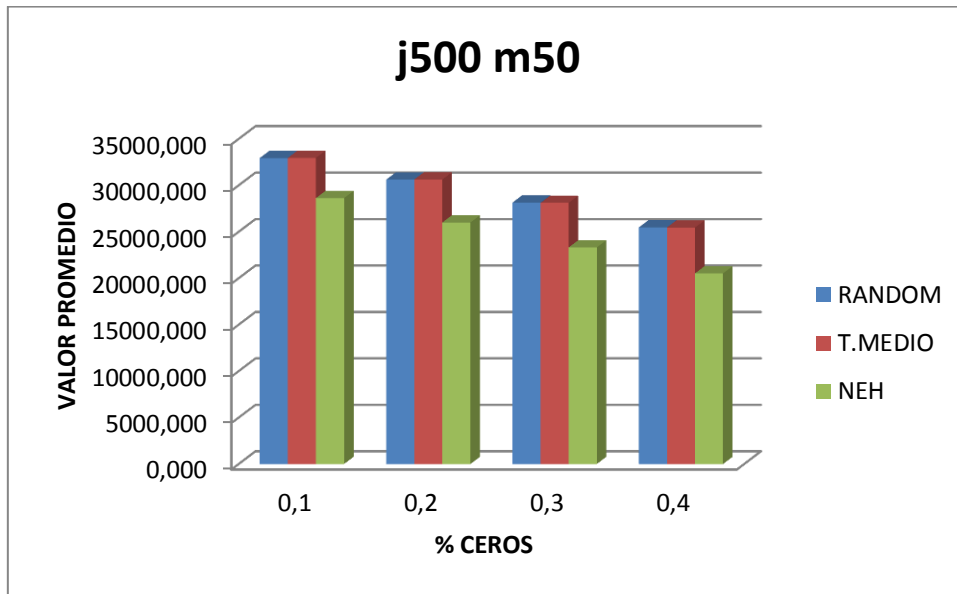


Figura 96: Valor promedio frente a porcentaje de cero para cada algoritmo del experimento de 500 trabajos y 50 máquinas.

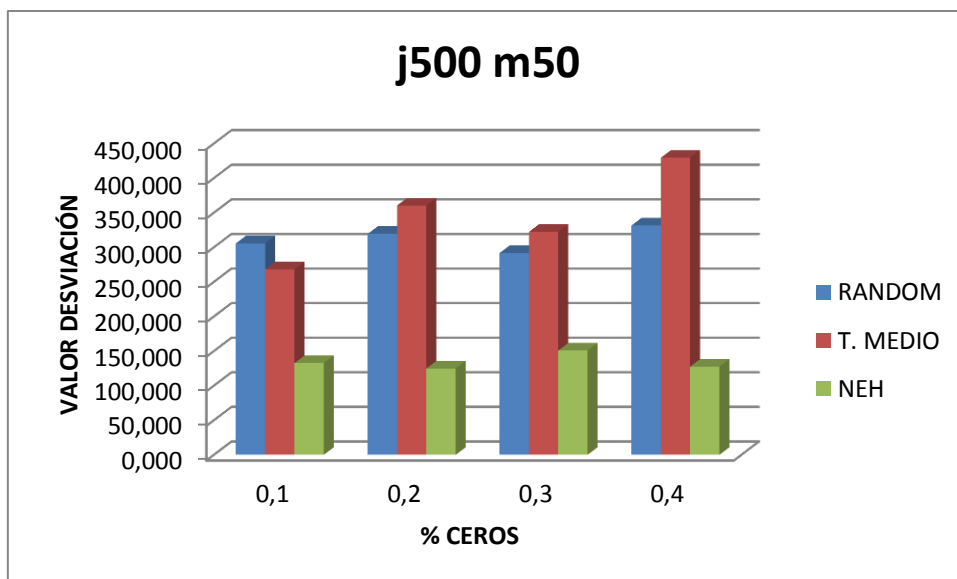


Figura 97: Valor desviación típica frente a porcentaje de cero para cada algoritmo del experimento de 500 trabajos y 50 máquinas.

En la figura 97 a excepción del experimento que concierne al 10% de los tiempos de proceso que son cero, las mayores desviaciones las presenta el algoritmo de los tiempos medios. En dicho caso citado, la mayor desviación la protagoniza el algoritmo Random ya que el algoritmo basado en la heurística de NEH obtiene unos valores muy inferiores a los de ellos dos.

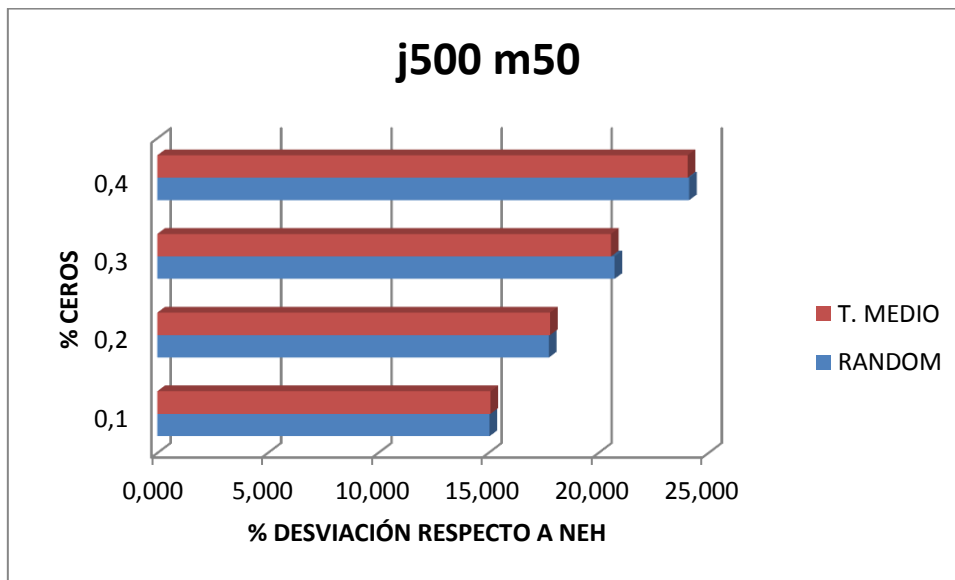


Figura 98: Porcentaje de mejora respecto a NEH de los algoritmos Random y T. Medio frente al porcentaje de ceros para el experimento de 500 trabajos y 50 máquinas.

Podemos observar en la figura 98 como permanece el mismo análisis comentado para otros experimentos similares, donde continúan los algoritmos Random y de los tiempos medios presentando resultados con una diferencia considerable respecto a la solución obtenida con NEH.

Se puede observar como el algoritmo de los tiempos medios obtiene una secuencia de trabajos similar a la del algoritmo que consigue la secuencia de forma aleatoria. Por lo que queda de manifiesto que este algoritmo tampoco alcanza una solución satisfactoria y adecuada para la programación de las operaciones y que tampoco se puede comparar con la secuencia alcanzada con NEH.

Incidir igualmente, que el algoritmo Random y el algoritmo de los tiempos medios han tardado un tiempo menor en encontrar la secuencia de trabajos respecto al algoritmo NEH aunque éste último ha logrado un mejor resultado en la misma proporción prácticamente.

Por lo que se puede afirmar la efectividad del algoritmo NEH frente a los otros dos algoritmos examinados, también, en este estudio, resultando ser la mejor elección para la búsqueda de la secuencia idónea de trabajos en un problema de Flow-Shop con tiempos de procesamientos de trabajos en máquinas iguales a ceros.

6. Conclusiones

La secuenciación adecuada de pedidos constituye un importante problema que se plantea dentro de la Dirección de Operaciones a corto plazo y el orden en que los pedidos serán atendidos o procesados, o en general el orden en el que cualquier tipo de tareas serán realizadas, no resulta indiferente, sino que determinará algún parámetro de interés cuyos valores convendrá optimizar en la medida de lo posible. Así podrá verse afectado el coste total de ejecución de los trabajos, el tiempo necesario para concluirlos o el stock de productos en curso que será generado.

Esto conduce de forma directa al problema de determinar cuál será el orden más adecuado para llevar a cabo las tareas con vistas a optimizar alguno de los anteriores parámetros o como es el caso que tratamos en este estudio, el de minimizar el tiempo de finalización de todos los trabajos, makespan.

Se trata de un problema de secuenciación o scheduling que se presenta de forma habitual en la programación de operaciones a corto plazo en entornos industriales o relativos a la producción y que puede adoptar una enorme variedad de formulaciones. La gran dificultad para resolver el problema determinando con una secuencia óptima, o al menos admisible, junto con la importancia de conseguirlo, han hecho proliferar reglas, más o menos complejas, muchas de ellas heurísticas, y algunas incluso empíricas, que proporcionan soluciones rápidas y fáciles de calcular destinadas a su uso en situaciones de trabajo reales.

Atendiendo a las premisas en las que está incluido nuestro problema se realiza el análisis y estudio del mismo. Para comprobar la aptitud de los algoritmos que he seleccionado, utilizo su aplicación en un banco de pruebas o experimentos, donde el número de trabajos, el número de máquinas y los tiempos de procesamiento de cada trabajo en cada máquina son valores arbitrarios, aunque dentro de unos rangos que se han considerado adecuados para realizar un estudio fiable y cercano a la realidad de la empresa.

Analizando los resultados obtenidos y la influencia de los valores arbitrarios, se puede concluir que la mejor alternativa es la heurística de NEH para la programación de operaciones en un taller de flujo con operaciones perdidas, frente a los algoritmos Random y Tiempo medio que también hemos examinado y comparados en la sección anterior.

Los resultados computacionales demuestran convincentemente la efectividad del algoritmo NEH. Ello confirma que este algoritmo es una eficaz herramienta para resolver problemas con un espacio de soluciones amplio y una función de optimización compleja.

NEH ha demostrado ser un buen método de resolución para una gran variedad de problemas difíciles. En esta ocasión, se ha adoptado a un problema de secuenciación marcado por los encuadres a los que está sometido nuestro problema y que se ha mencionado en apartados anteriores.

Utilizar la heurística NEH en los casos de alta complejidad presenta grandes ventajas, ya que reduce el costo computacional. Tratando como alta complejidad, los estudios realizados para valores elevados de trabajos y máquinas.

Destacar que el algoritmo Random no puede presentarse como una alternativa si se desea encontrar una secuencia de trabajos satisfactoria. Su sencillez a la hora de implementar el código se asemeja al algoritmo donde se calculan los tiempos medios, y este último presenta mejores resultados a la hora de minimizar el tiempo de finalización de todos los trabajos. Como ya se mencionó en el comienzo de este trabajo, el algoritmo Random es el equivalente a no programar la producción y no realizar ninguna acción referente a la planificación de los trabajos, y como tal se ha tratado en este estudio.

A medida que la capacidad de los ordenadores y programas ha ido creciendo, se ha incrementado la posibilidad de explorar áreas cada vez mayores en el espacio de soluciones, lo que a su vez garantiza mejores soluciones. Esto ha provocado que se hayan resuelto muchos problemas en el área de la programación de operaciones,

que en el pasado no han podido ser resueltos debido a la gran complejidad que encerraban.

En la actualidad, las organizaciones y principalmente las relacionadas con el área de producción buscan constantemente la mejora continua, razón por la cual día tras día deben ser más eficientes en la utilización de los recursos, a través de la optimización de sus procesos lo cual se logra en la medida en que se realicen una serie de estrategias entre ellas una adecuada programación de la producción.

7. Bibliografía

Martínez Sánchez, A.: "La estrategia de fabricación y la competitividad de la empresa", Editorial. Alta Dirección.

Boris, William: Control de Producción, Editorial Hispano-Europea.

Companys, R.: Planificación y Programación de la Producción. Ed. Marcombo, 1989.

Nawaz, M; Enscore, Jr. E.; Ham, I.: A heuristic algorithm for the machine, n-job flow shop sequencing problem, Omega, The International Journal of Management Science.

Wollmann, Thomas E.; Berry, Willian L.; Whybark, D. Clay: Gerencia Sistemas de Planificación y Control de la Producción. Ed. Irwin.

Johnson SM. Optimal two-and three-stage production schedules with setup times included. Naval Res Logist Q, 1954.

Framiñan Jose M; Fernandez-Viagas, Victor: On insertion tie-breaking rules in heuristics for the permutation flowshop scheduling problem.

Framiñan, Jose M; Gupta, J; Leisten, R.: A review and classification of heuristics for permutation flow-shop scheduling with makespan objective. J Oper Res Soc, 2004.

Framiñan, Jose M; Leisten, R.: An efficient constructive heuristic for flowtime minimisation in permutation flowshops. Omega, Int J Manag Sci, 2003.

Framiñan, Jose M; Leisten, R; Rajendran, C.: Different initial sequences for the heuristic of Nawaz, Enscore and Ham to minimize makespan, idletime or flowtime in the static permutation flowshop sequencing problem. Int J Prod Res, 2003.

Domínguez Machuca, J.A. Dirección de Operaciones Aspectos Tácticos y Operativos en la Producción y los Servicios Ed. Mc Graw-Hill, 1995.

Domínguez Machuca, José Antonio: Dirección de operaciones – Aspectos Estratégicos en la Producción y los Servicios Ed. Mc Graw-Hill, 1995.

Berry, Willian L. y Whybark, D. Clay.: Sistemas de Planificación y Control de la Fabricación Ed. McGraw-Hill.

Fernández E.; Avella L. y Fernández M.: Estrategia de Producción Ed. Mc Graw Hill.

Díaz, Adenso: "Producción: Gestión y Control." Ed. Ariel Economía, Barcelona.

Domínguez Machuca, J.A.; García, S.: "La gestión integrada de la empresa y el papel representado por el MRP." Ed. Alta Dirección.

García González, S.: "Dirección de Operaciones y Competitividad ", en "Orígenes y causas de la crisis de competitividad".

Heizer, J. y Render.B.: "Dirección de la Producción. Decisiones tácticas" Ed. Prentice-Hall.

De la Fuente García, David: Ingeniería de organización en la empresa: Dirección de operaciones, Ediciones de la Universidad de Oviedo, 2008.

<http://www.codeblocks.org/>

Code::Blocks IDE User Guide JN-UG-3028

http://www.jennic.com/support/user_guides/jn-ug-3028_codeblocks_ide

García Sánchez, José Manuel: Apuntes asignatura Programación de Operaciones,2014.

Velasco Sánchez, Juan: Gestión de la producción en la empresa: planificación, programación y control, Ed. Pirámide, 2013.

Chapman, Stephen N.: Planificación y control de la producción, Ed. Prentice Hall, 2006.

Crespo Franco, Antonio: Producción: planificación, programación e control, Ed. Servicio de Publicacións da Universidade de Vigo, 2003.

Cruelles Ruiz, José Agustín: Productividad industrial: métodos de trabajo, tiempos y su aplicación a la planificación y mejora continua, Ed. Marcombo, 2013.

8. Anexos

8.1 Anexo 1

8.1.1 Código Algoritmo Random

```
#include <stdio.h>
#include <stdlib.h>
#include <schedule.h>
int main(int argc, char *argv[]) {
// 1. Variables
    int maquinas = 0;        // máquinas
    int trabajos = 0;        // trabajos
    MAT_INT pt;              // tiempos de proceso
    VECTOR_INT secuencia_trabajos; // secuencia
    long int makespan;       // makespan
// 2. Check arguments
    if(argc!=2) error(4, STOP, "Usage: load_flowshop
[datos.txt]\n");
// 3. Carga los datos
    pt = loadPTimes_nrows(argv[1], &trabajos, &maquinas,
NO);
// 4. Genera secuencia
    secuencia_trabajos = DIM_VECTOR_INT(trabajos);
    randSequence(secuencia_trabajos, trabajos);
// 5. Calcula makespan
    makespan = Cmax(secuencia_trabajos, trabajos, maquinas,
pt);
// 6. Muestra de resultados
    printf("\nEl makespan de la secuencia de trabajos
aleatoria es %ld", makespan);
    printf("\nLa secuencia de proceso seguida es: ");
```

```
    print_int_vector(secuencia_trabajos, trabajos);  
// 7. Liberar espacio  
    free_mat_int(pt, trabajos);  
    free(secuencia_trabajos);  
    return 1;  
}  
// Fin
```

8.1.2 Código Algoritmo Tiempos Medios

```
#include <stdio.h>  
#include <stdlib.h>  
#include <schedule.h>  
int main(int argc, char *argv[]) {  
// 1. Variables  
    int i; // índice para las filas de  
matriz de tiempos  
    int j; // índice para las columnas de  
matriz de tiempos  
    int maquinas = 0; // número de maquinas  
    int trabajos = 0; // número de trabajos  
    int c; // número de etapas o columnas  
    int suma; // tiempo por trabajo  
    MAT_INT pt; // matriz de tiempos de trabajos  
    VECTOR_INT secuencia; // secuencia  
    VECTOR_INT vtiempos; // vector de tiempos medios  
    long int makespan; // makespan  
// 2. Check datos  
    if(argc!=2) error(4, STOP, "Usage: makespan_flowshop  
[datos.txt]\n");  
// 3. Carga datos  
    pt = loadPTimes_nrows(argv[1], &trabajos, &maquinas,  
NO);
```

```
// 4. Calculo tiempos medios de los trabajos
vtiempos = DIM_VECTOR_INT(trabajos);
for(i=0;i<trabajos;i++)
{
    c=0;
    suma=0;
    for (j=0;j<maquinas;j++)
    {
        suma= suma + pt[i][j];
        c=c+1;
    }
    vtiempos[i]=suma/c;
}

printf("\nLos tiempos medio para cada trabajo
respectivamente son: ");

print_int_vector(vtiempos,trabajos);

// 5. Creacion de un vector auxiliar igual que vtiempos
secuencia = DIM_VECTOR_INT(trabajos);
copyIVector(vtiempos,secuencia,trabajos);

// 6. Ordeno el vector auxiliar de forma creciente respecto a
sus tiempos de proceso
sortIVector(secuencia,trabajos,'A');

// 7. Calculo makespan
makespan = Cmax(secuencia,trabajos, maquinas,pt);

// 8. Muestro los resultados
printf("\nLa secuencia obtenida es: ");
print_int_vector(secuencia,trabajos);
printf("\nCmax = %ld\n", makespan);

// 9. Libero memoria
free_mat_int(pt,trabajos);
free(secuencia);
free(vtiempos);
```

```
    return 1;
}
// Fin main()
```

8.1.3 Código Algoritmo NEH

```
#include <stdio.h>
#include <stdlib.h>
#include <schedule.h>
int main(int argc, char *argv[]) {
// 1. Variables
    int a; // índice para dimensionar el
vector de combinaciones
    int g; // índice para recorrer el
vector vaux
    int h; // índice para recorrer el
vector vtiempos
    int i; // índice para las filas de
matriz de tiempos
    int j; // índice para las columnas de
matriz de tiempos
    int q; // índice para recorrer el
vector secuencia
    int w; // índice para realizar el
numero correcto de combinaciones
    int u; // índice para recorrer vaux
para ordenarlo de menor a mayor
    int p; // índice para recorrer vaux
para ordenarlo de menor a mayor
    int aux; // entero auxiliar
    int maquinas = 0; // número de maquinas
    int trabajos = 0; // número de trabajos
    int suma; // tiempo por trabajo
    long int makespanaux; // makespan auxiliar
    long int makmax; // makespan menor (maximo en
terminos de optimalidad)
    MAT_INT pt; // matriz de tiempos de trabajos
```



```

VECTOR_INT secuencia;    // secuencia

VECTOR_FLOAT vtiempos;  // vector de tiempos medios

VECTOR_FLOAT vaux;      // vector auxiliar para hallar
la secuencia ordenada por tiempo de proceso

    VECTOR_INT vini;     // vector auxiliar para hallar
la secuencia mediante makespan

    VECTOR_INT secaux;   // vector auxiliar para hallar
la secuencia mediante makespan

    VECTOR_INT control; // vector auxiliar auxiliar para
hallar la secuencia mediante makespan

    VECTOR_INT NEH;     // vector secuencia optima del
algoritmo NEH

// 2. Check datos

    if(argc!=2) error(4, STOP, "Usage: makespan_flowshop
[datos.txt]\n");

// 3. Carga datos

    pt = loadPTimes_nrows(argv[1], &trabajos, &maquinas,
NO);

// 4. Calculo los tiempos de los trabajos

    vtiempos = DIM_VECTOR_FLOAT(trabajos);
    for(i=0;i<trabajos;i++)
    {
        suma=0;
        for (j=0;j<maquinas;j++)
            {
                Suma+=pt[i][j];
            }
        vtiempos[i]=suma;
    }

// 5. Creacion de un vector auxiliar igual que vtiempos

    vaux = DIM_VECTOR_FLOAT(trabajos);
    copyFVector(vtiempos,vaux,trabajos);

// 6. Ordeno el vector auxiliar de forma creciente respecto a
sus tiempos de proceso

    for (u=0; u<trabajos; u++)

```

```

{
    for (p=u+1; p<trabajos; p++)
        {
            if (vaux[p]<vaux[u])
                {
                    aux=vaux[u];
                    vaux[u]=vaux[p];
                    vaux[p]=aux;
                }
        }
}

// 7. Comparo los valores de cada uno de ellos para conocer la
// secuencia de trabajos ordenada por menor tiempo de proceso

secuencia = DIM_VECTOR_INT(trabajos);
for (g=0; g<trabajos; g++)
{
    for (h=0; h<trabajos; h++)
        {
            if (vaux[g]==vtiempos[h])
                {
                    secuencia[g]=h;
                }
        }
}

// 8. Creo el nuevo vector que voy a utilizar para hallar la
// secuencia con menor makespan

vini = DIM_VECTOR_INT(1);
vini[0]=secuencia[0];

// 9. Seccion para calcular la secuencia

for (q=1; q<trabajos; q++)
{
    a=q+1;

```

```
    makmax=1000000000000;
control = DIM_VECTOR_INT(a);
    secaux = DIM_VECTOR_INT(a);
    for (w=0; w<a; w++)
        {
            copyIVector(vini,secaux,q);
            insertIVector(secaux,a,secuencia[q],w);
            makespanaux = Cmax(secaux,a,maquinas,pt);
            if (makespanaux < makmax)
                {
                    makmax = makespanaux;
                    copyIVector(secaux,control,a);
                }
        }
free(secaux);
    free(vini);
    vini = DIM_VECTOR_INT(a);
    copyIVector(control,vini,a);
if (a==trabajos)
    {
        NEH = DIM_VECTOR_INT(trabajos);
        copyIVector(control,NEH,trabajos);
    }
    free(control);
}
// 10. Muestro los resultados
    printf("\nEl makespan de la secuencia de trabajos es
%ld", makmax);
    printf("\nLa secuencia solucion del algoritmo NEH es:
");
    print_int_vector(NEH,trabajos);
// 11. Libero memoria
```

```
free_mat_int(pt, trabajos);  
free(sequencia);  
free(vaux);  
free(vtiempos);  
free(vini);  
free(secaux);  
free(NEH);  
return 1;  
} // Fin main()
```

8.2 Anexo 2

➤ `copyFVector(src, destn., length)`

Copia un vector en otro, ambos declarados como vectores float.

Parámetros:

- Src: Vector origen
- Destn: Vector destino
- Length: Longitud de ambos vectores

Definida de la siguiente forma:

```
void copyFVector(float *source, float *destination, int len) {  
  register int i;  
  for(i=0;i<len;i++) destination[i] = source[i];  
} // end of copyFVector()
```

Ejemplo de utilización:

```
copyFVector(vtiempos,vaux,trabajos);
```

➤ `copyIVector(src, destn., length)`

Copia un vector en otro, ambos declarados como vectores de enteros.

Parámetros:

- Src: Vector origen
- Destn: Vector destino
- Length: Longitud de ambos vectores

Definida de la siguiente forma:

```
void copyIVector(int *source, int *destination, int len) {  
  register int i;
```

```
for(i=0;i<len;i++) *(destination+i)=*(source+i);  
} // end of copyIVector()
```

Ejemplo de utilización:

```
copyIVector(vtiempos,secuencia,trabajos);
```

➤ `randSequence(vector, jobs)`

Genera una secuencia aleatoria de trabajos de longitud len. La primera posición es cero, vector es el vector trabajos de longitud trabajos.

Parámetros:

- Vector: Vector de secuencia.
- Jobs: Longitud del vector que coincide con el número de trabajos.

Definida de la siguiente forma:

```
void randomVector( int *vector, int length) {  
register int i,j; // counters  
// For every position, seeks for a feasible number  
for(j=0;j<length;j++) {  
do {  
i = (rand()%length)+1;  
} while(searchInIVector(vector,j,i)!=-1);  
*(vector+j) = i;  
}  
// end of randomVector()
```

Ejemplo de utilización:

```
randSequence(secuencia_trabajos,trabajos);
```

➤ `sortLVector(vector,rws,crit)`

Ordenar un vector de `rws` filas utilizando un criterio específico, de forma ascendente o descendente.

Parámetros:

- Vector: Vector de enteros.
- Rws: Filas del vector.
- Crit: Criterio específico de orden (ascendente o descendente).

Definida de la siguiente forma:

```
void sortLVector(long int *vector, int *index, int rows, char
criterion) {
// 0. Variables
register int i;
// 1. Sample order
for(i=0;i<rows;i++) index[i] = i;
// 2. Call the branch, depending on the criterion
switch(criterion) {
case 'A':
brqsortLVectorA(vector, index, 0, rows-1);
break;
case 'D':
brqsortLVectorD(vector, index, 0, rows-1);
break;
default:
error(999, STOP, "Undefined sorting criterion\n");
}
} // end of sortLVector()
```

Ejemplo de utilización:

```
sortIVector(secuencia,trabajos,'A');
```

➤ Cmax(seq, Jobs, machs, pt)

Calcula la duración total (makespan) de una secuencia dada en un problema de flowshop. La secuencia comienza en cero.

Parámetros:

- Seq: Secuencia de trabajos.
- Jobs: Número de trabajos en la secuencia.
- Machs: Número de máquinas en flowshop.
- Pt: Matriz de tiempos de procesos. Ejemplo: pt[i][j] es el tiempo de proceso del trabajo i en la máquina j.

Definida de la siguiente forma:

```
// 0. Variables
MAT LONG ct; // completion times
register int i,j; // counters
long int makespan;

// 1. Initialize proc. times
ct = DIM MAT LONG(jobs, machs);

// 2. Calculate completion times
// 2.1. First job in first machine
ct[0][0] = (long int) pt[seq[0]][0];

// 2.2. First job in the rest of machines
for(j=1;j<machs;j++) ct[0][j] = ct[0][j-1] + ( (long int)
pt[seq[0]][j] );

// 2.3. Other jobs in the first machine
for(i=1;i<jobs;i++) ct[i][0] = ct[i-1][0] + ( (long int)
pt[seq[i]][0] );
```



```
// 2.4. Other jobs in other machines

for(i=1;i<jobs;i++) {

for(j=1;j<machs;j++)

ct[i][j] = (lmax( ct[i-1][j], ct[i][j-1]) + ((long int)
pt[seq[i]][j]) ) ;

}

// 3. Saving makespan

makespan = ct[jobs-1][machs-1];

// 4. Freeing completion times

FREE MAT LONG(ct,jobs);

return makespan;

} // end of Cmax()
```

Ejemplo de utilización:

Cmax(secuencia_trabajos, trabajos, maquinas, pt);

➤ loadPTimes_nrows(file,n,m,print_flag)

Carga los tiempos de procesos desde un archivo, almacenándolos en variables de memoria. El archivo debe tener la siguiente estructura (para un flowshop de n trabajos y m máquinas).

```
m
n
p11 p12 ...      p1m
.....
Pn1 pn2 ...      pnm
```

Parámetros:

- Filename: nombre del archivo.
- Machs: Puntero en el número de máquinas.

- Jobs: Puntero en el número de trabajos.
- Print_flag: Si los datos se deben mostrar o no (YES o NOT).

Definida de la siguiente forma:

```
// 0. Variables
FILE *data; // file pointer
int temp_data; // to store processing times
register int i,j; // counters
MAT_INT pt; // processing times matrix
// 1. Checks that file exists
if (!(data = fopen(filename,"rt"))) error(2,STOP, filename);
// 2. Reading data...
// 2.1. Headers
fscanf(data,"%d\n", &temp_data);
*(machs) = temp_data;
fscanf(data,"%d\n", &temp_data);
*(jobs) = temp_data;
// 2.2. Ask for space for matrix pt
pt = DIM MAT_INT((*jobs), (*machs));
// 2.3. Processing times
for(i=0;i<(*jobs);i++) {
for(j=0;j<*(machs);j++) {
fscanf(data,"%10d", &temp_data );
// storing processing times
pt[i][j] = temp_data;
}
fscanf(data,"\n");
} // end of 2.2.
// 3. Close file
```

```
fclose(data);

// 4. In case print_flag is set to yes

printf("Instance %s - n: %d m: %d\n",filename, *(jobs),
*(machs));

if(print_flag == YES) {
print_int_matrix(pt, *(jobs), *(machs) );
} // end of 4.

// 5. Returning processing times matrix
return pt;
} // end of loadPTimes_nrows()
```

Ejemplo de utilización:

```
pt = loadPTimes_nrows(argv[1], &trabajos, &maquinas, NO);
```

➤ print_int_vector(vector,len)

Imprime un vector de enteros por pantalla.

Parámetros:

- Vector: Vector de enteros.
- Len: Longitud del vector.

Definida de la siguiente forma:

```
register int i;
for(i=0;i<len;i++) printf("%5d", vector[i]);
printf("\n");
} // end of print_int_vector()
```

Ejemplo de utilización:

```
print_int_vector(NEH,trabajos);
```

➤ insertIVector(vector,len,val,pos)

Inserta valores en una posición determinada en un vector de enteros.

Parámetros:

- Vector: Vector de enteros.
- Len: Longitud del vector.
- Val: Valor que se desea insertar.
- Pos: Posición del vector en la que se quiera insertar dicho valor.

Definida de la siguiente forma:

```
register int i;
// shift forward those position downstream pos
for(i=(length-1);i>pos;i--) *(vector+i)=*(vector+i-1);
//set the value in position pos
*(vector+pos)=value;
} // end of insertIVector()
```

Ejemplo de utilización:

```
insertIVector(secaux,a,secuencia[q],w);
```

8.3 Anexo 3

INSTANCIAS	RANDOM	T. MEDIO	NEH
1	3619	3995	3082
2	3491	3415	2964
3	3850	3621	3093
4	3672	3623	3244
5	3702	3941	2987
6	3768	3701	3060
7	3773	3473	3129
8	3414	3497	3008
9	3609	3549	2895
10	3883	3303	3095
11	3783	3771	3003
12	3758	3414	3003
13	3463	3499	3030
14	3524	3412	2894
15	3918	3609	2989
16	3629	3555	3028
17	3662	3668	2917
18	3483	3578	3034
19	3646	3412	2865
20	3832	3889	3198
21	3692	3561	2979
22	3524	3799	3238
23	3659	3496	3097
24	3732	3554	3076
25	3587	3406	2797
26	3669	3450	2917
27	3649	3636	3019
28	3692	3506	3062
29	3599	3611	2979
30	3626	3676	3079
MEDIA	3663,600	3587,333	3025,367
DESVIACIÓN	124,838	165,687	102,925

8.4 Anexo 4

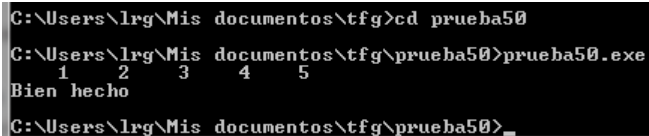
Ejemplos de iniciación con el programa Code::Blocks utilizando la librería Schedule.h facilitada por el Departamento de Organización y Gestión de Empresas I.

- 1) Crear un vector de trabajos e imprimirlo por pantalla.

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <schedule.h>
4  int main()
5  {
6      VECTOR_INT trabajos;
7      trabajos = DIM_VECTOR_INT(5);
8      trabajos [0]= 1;
9      trabajos [1]= 2;
10     trabajos [2]= 3;
11     trabajos [3]= 4;
12     trabajos [4]= 5;
13
14     print_int_vector(trabajos,5);
15     free(trabajos);
16     printf("Bien hecho\n");
17     return 0;
18 }

```



```

C:\Users\lrg\Mis documentos\tfg>cd prueba50
C:\Users\lrg\Mis documentos\tfg\prueba50>prueba50.exe
1 2 3 4 5
Bien hecho
C:\Users\lrg\Mis documentos\tfg\prueba50>_

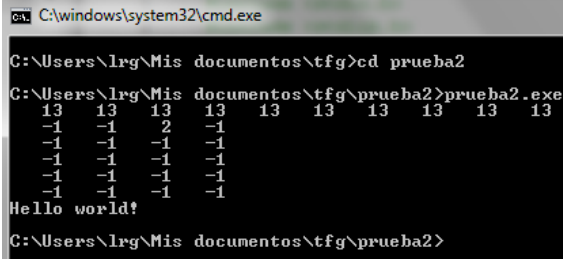
```

- 2) Introducir a un vector de dimensión 10 el valor 13 en todas sus posiciones. Obtener una matriz que tenga el valor -1 en todas sus posiciones, excepto en la fila 1-columna 3 que presentará el valor 2.

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <schedule.h>
4
5  int main()
6  {
7      VECTOR_INT mivector;
8      MAT_INT mimatriz;
9
10     mivector = DIM_VECTOR_INT(10);
11     mimatriz = DIM_MAT_INT(5,4);
12     setval_IVector(mivector,10,13);
13     setval_IMatrix(mimatriz,5,4,-1);
14     mimatriz[0][2] = 2;
15     print_int_vector(mivector,10);
16     print_int_matrix(mimatriz,5,4);
17     printf("Hello world!\n");
18     free(mivector);
19
20     FREE_MAT_INT(mimatriz,5);
21     return 0;

```



```

C:\windows\system32\cmd.exe
C:\Users\lrg\Mis documentos\tfg>cd prueba2
C:\Users\lrg\Mis documentos\tfg\prueba2>prueba2.exe
13 13 13 13 13 13 13 13 13 13
-1 -1 -1 -1
-1 -1 -1 -1
-1 -1 -1 -1
-1 -1 -1 -1
-1 -1 -1 -1
Hello world!
C:\Users\lrg\Mis documentos\tfg\prueba2>

```

- 3) Imaginar 5 trabajos cuyos tiempos de procesos en las 5 máquinas son iguales y son respectivamente 7, 10, 14, 22, 17. Obtener por pantalla la matriz de tiempos en la que la primera fila estará dedicada para especificar el número de la máquina a la que corresponde el trabajo.

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <schedule.h>
4  int main()
5  {
6      VECTOR_INT trabajos;
7      VECTOR_INT trabajos1;
8      VECTOR_INT trabajos2;
9      VECTOR_INT trabajos3;
10     VECTOR_INT trabajos4;
11     VECTOR_INT trabajos5;
12     MAT_INT matriztiempos;
13
14     trabajos = DIM_VECTOR_INT(5);
15     trabajos1 = DIM_VECTOR_INT(5);
16     trabajos2 = DIM_VECTOR_INT(5);
17     trabajos3 = DIM_VECTOR_INT(5);
18     trabajos4 = DIM_VECTOR_INT(5);
19     trabajos5 = DIM_VECTOR_INT(5);
22     trabajos [0]= 1;
23     trabajos [1]= 2;
24     trabajos [2]= 3;
25     trabajos [3]= 4;
26     trabajos [4]= 5;
27
28     setval_IVector(trabajos1,5,7);
29     setval_IVector(trabajos2,5,10);
30     setval_IVector(trabajos3,5,14);
31     setval_IVector(trabajos4,5,22);
32     setval_IVector(trabajos5,5,17);
33
34     pasterowImatrix(matriztiempos,trabajos,0,5);
35     pasterowImatrix(matriztiempos,trabajos1,1,5);
36     pasterowImatrix(matriztiempos,trabajos2,2,5);
37     pasterowImatrix(matriztiempos,trabajos3,3,5);
38     pasterowImatrix(matriztiempos,trabajos4,4,5);
39     pasterowImatrix(matriztiempos,trabajos5,5,5);
40
41     print_int_matrix(matriztiempos,6,5);
42     printf("Bien hecho\n");
43
44     free(trabajos);
45     free(trabajos1);
46     free(trabajos2);
47     free(trabajos3);
48     free(trabajos4);
49     free(trabajos5);
50     FREE_MAT_INT(matriztiempos,5);
51
52     return 0;
53 }
54
```

```

C:\windows\system32\cmd.exe
Microsoft Windows [Versión 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. Reservados todos los derechos.

C:\Users\lrg>cd mis documentos
C:\Users\lrg\Mis documentos>cd tfg
C:\Users\lrg\Mis documentos\tfg>cd prueba3
C:\Users\lrg\Mis documentos\tfg\prueba3>prueba3.exe
 1  2  3  4  5
 7  7  7  7  7
10 10 10 10 10
14 14 14 14 14
22 22 22 22 22
17 17 17 17 17
Bien hecho
C:\Users\lrg\Mis documentos\tfg\prueba3>_
    
```

- 4) Crear un vector de dimensión 5 correspondiente al número de máquinas. Introducir dicho vector en la fila 1 de la matriz. El resto de filas y columnas deben tomar el valor 7. Dimensión de la matriz 7x5.

```

4 int main()
5 {
6     VECTOR_INT num_maq;
7     num_maq = DIM_VECTOR_INT(5);
8     num_maq [0]=1;
9     num_maq [1]=2;
10    num_maq [2]=3;
11    num_maq [3]=4;
12    num_maq [4]=5;
13
14    MAT_INT matriz;
15    matriz = DIM_MAT_INT(6,5);
16    setval_imatrix(matriz,6,5,7);
17
18    print_int_vector(num_maq,5);
19    print_int_matrix(matriz,6,5);
20    printf("Bien hecho\n");
21
22    free(num_maq);
23    free_mat_int(matriz,6);
24    return 0;
    
```

```

C:\Users\lrg\Mis documentos\tfg\prueba4>prueba4.exe
 1  2  3  4  5
 7  7  7  7  7
 7  7  7  7  7
 7  7  7  7  7
 7  7  7  7  7
 7  7  7  7  7
 7  7  7  7  7
Bien hecho
    
```


- 5) Crear un vector de tiempos de dimensión 3 y valores decimales. Copiar dicho vector en otro e imprimir por pantalla.

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <schedule.h>
4  int main()
5  {
6      VECTOR_DOUBLE tiempos1;
7      VECTOR_DOUBLE tiempos2;
8      MAT_DOUBLE maq_trab;
9
10     maq_trab = DIM_MAT_DOUBLE(3,3);
11     tiempos1 = DIM_VECTOR_DOUBLE(3);
12     tiempos2 = DIM_VECTOR_DOUBLE(3);
13
14     tiempos1 [0] = 7.5;
15     tiempos1 [1] = 10.5;
16     tiempos1 [2] = 2.5;
17     VECTOR_DOUBLE main::tiempos1
18     copyDVector(tiempos1,tiempos2,3);
19
20     print_double_vector(tiempos1,3);
21     print_double_vector(tiempos2,3);
22     printf("ok!\n");
23
24     free(tiempos1);
25     free(tiempos2);
26     free_mat_double(maq_trab,3);
27
28     return 0;
29 }
30
```

```
C:\Users\lrg\Mis documentos\tfg>cd prueba5
C:\Users\lrg\Mis documentos\tfg\prueba5>prueba5.exe
7.50000010.5000002.500000
7.50000010.5000002.500000
ok!
```

- 6) Crear un vector trabajos de dimensión 5 con valores de 1 a 5. Crear 5 vectores más cuyos tiempos en cada máquina son los mismos y corresponden respectivamente a 7, 10, 14, 22, 17. Introducir todos los vectores en la matriz de tiempos con el fin de mostrarla por pantalla.

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <schedule.h>
4
5  int main()
6  {
7      int k;
8
9      VECTOR_INT trabajos;
10     VECTOR_INT trabajos1;
11     VECTOR_INT trabajos2;
12     VECTOR_INT trabajos3;
13     VECTOR_INT trabajos4;
14     VECTOR_INT trabajos5;
15     MAT_INT matriztiempos;
16
17     trabajos = DIM_VECTOR_INT(5);
18     trabajos1 = DIM_VECTOR_INT(5);
19     trabajos2 = DIM_VECTOR_INT(5);
20     trabajos3 = DIM_VECTOR_INT(5);
21     trabajos4 = DIM_VECTOR_INT(5);
22     trabajos5 = DIM_VECTOR_INT(5);
23     matriztiempos = DIM_MAT_INT(6,5);
24
25     for (k=0;k<5;k++) trabajos[k] = k+1;
26
27     setval_IVector(trabajos1,5,7);
28     setval_IVector(trabajos2,5,10);
29     setval_IVector(trabajos3,5,14);
30     setval_IVector(trabajos4,5,22);
31     setval_IVector(trabajos5,5,17);
32
33     pasterowImatrix(matriztiempos,trabajos,0,5);
34     pasterowImatrix(matriztiempos,trabajos1,1,5);
35     pasterowImatrix(matriztiempos,trabajos2,2,5);
36     pasterowImatrix(matriztiempos,trabajos3,3,5);
37     pasterowImatrix(matriztiempos,trabajos4,4,5);
38     pasterowImatrix(matriztiempos,trabajos5,5,5);
39
40     print_int_matrix(matriztiempos,6,5);
41     printf("Buen trabajo\n");
42
43     free(trabajos);
44     free(trabajos1);
45     free(trabajos2);
46     free(trabajos3);
47     free(trabajos4);
48     free(trabajos5);
49
50     FREE_MAT_INT(matriztiempos,5);
51     return 0;
52 }
53

```

```

C:\Users\lrg\Mis documentos\tfq>cd prueba8
C:\Users\lrg\Mis documentos\tfq\prueba8>prueba8.exe
 1  2  3  4  5
 7  7  7  7  7
10 10 10 10 10
14 14 14 14 14
22 22 22 22 22
17 17 17 17 17
Buen trabajo
C:\Users\lrg\Mis documentos\tfq\prueba8>

```

