

Procedimientos de Búsqueda Local Aplicados a Problemas de Secuenciación de Máquinas

Julio Mario Daza-Escorcia¹, David Álvarez Martínez²

¹ Universidad de los Andes
Carrera 1 Este No. 19A – 40, Bogotá, Colombia
jm.dazae@uniandes.edu.co

² Universidad de los Andes
Carrera 1 Este No. 19A – 40, Bogotá, Colombia
d.alvarezm@uniandes.edu.co

Resumen

Este artículo presenta dos heurísticos de búsqueda local para el problema de secuenciación de máquinas, denominado *Flow Shop Scheduling Problem* (FSSP). Estos heurísticos establecen dos vecindarios para el FSSP que se combinan estratégicamente para lograr buenas soluciones. El primer heurístico de búsqueda local se basa en el conocido 2opt, mientras que el segundo se basa en un intercambio aleatorio. Los experimentos sobre las instancias de referencia de la literatura muestran que nuestros algoritmos se encuentran cercanos a los *Best Know Solutions* (BKS's) presentes en la literatura, obteniendo dos nuevos BKS's que mejoran el makespan en un 0.81% y 1.06% respectivamente.

1 Introducción

Los problemas de secuenciación de máquinas consisten en asignar recursos limitados a tareas a lo largo del tiempo y determinar la secuencia de operaciones para que se cumplan las restricciones del sistema y se optimicen los criterios de rendimiento (Akyol y Bayhan 2007). Las restricciones del sistema definen el problema particular a desarrollar. En nuestro caso, estudiaremos el FSSP que es uno de los problemas de secuenciación más estudiados.

El FSSP consiste en un conjunto de n trabajos independientes que tienen que ser procesados en un conjunto de m máquinas en el orden dado por la indexación de las máquinas, teniendo en cuenta que la secuencia de procesamiento de los trabajos es la misma para todas las máquinas. Cada trabajo cuenta con un tiempo de procesamiento en cada máquina y cada máquina puede procesar solo una operación a la vez. El objetivo es encontrar el orden de los trabajos que minimiza un criterio dado, en nuestro caso el tiempo máximo de terminación de los trabajos o *makespan*.

El FSSP es uno de los problemas más importantes y complicados en la programación de máquinas, y se conoce como un problema de optimización combinatoria NP-hard (Rinnooy Kan, 1976; Lenstra et al. 1977; Zhang et al. 2013). La alta complejidad del problema dificulta encontrar la solución óptima en un tiempo razonable en la mayoría de los casos (Asadzadeh y Zamanifar 2010) y solo pequeñas instancias pueden ser resueltas de forma exacta.

Como consecuencia, el FSSP presenta interés para profesionales e investigadores en el ámbito de problemas de programación o secuenciación de máquinas, donde podemos observar que los investigadores han centrado sus energías principalmente en enfoques heurísticos: de constructivos y mejora.

En el primer caso podemos resaltar los trabajos de (Campbell et al., 1970; Dannenbring, 1977) que proponen métodos constructivos que proporcionan una buena solución en poco tiempo. Además, el trabajo de (Nawaz et al., 1983) que presentan la heurística NEH, reconocida como la que genera los mejores resultados para este problema. En segundo lugar, se encuentran los métodos de mejora mediante un enfoque metaheurístico, donde resaltan los trabajos de (Osman y Potts, 1989; Ogbu y Smith, 1990) que presentan algoritmos basados en recocido simulado, (Widmer y Hertz, 1989; Taillard, 1990; Reeves, 1993; Nowicki y Smutnicki, 1996; Grabowski y Wodecki, 2004) que estudian métodos de solución mediante búsqueda tabú, (Stützle, 1998; Ruiz y Stützle, 2006) que resuelve el problema a través de una búsqueda local iterada. Finalmente, (Reeves, 1995; Reeves y Yamada, 1998) que utilizan algoritmos genéticos para dar solución a este problema.

El resto del documento está organizado como sigue. La sección 2 describe formalmente el FSSP. La sección 3 presenta los heurísticos de búsqueda local desarrollados. La sección 4 muestra el resultado de los experimentos computacionales y finalmente se presenta las conclusiones del trabajo.

2 Descripción del Problema

El *Flow Shop Scheduling Problem* (FSSP) es un problema general de programación o secuenciación de máquinas en el que:

- cada trabajo i consta de m operaciones O_{ij} con tiempos de procesamiento p_{ij} ($j=1, \dots, m$) donde O_{ij} debe procesarse en la máquina M_j , y
- existen restricciones de precedencia de la forma $O_{ij} \rightarrow O_{i,j+1}$ ($i=1, \dots, m-1$) para cada $i=1, \dots, n$, es decir, cada trabajo se procesa primero en la máquina 1, luego en la máquina 2, luego en la máquina 3, etc.

Así, el problema consiste en encontrar un orden de cada trabajo π_j para cada máquina j , con el objetivo de minimizar el tiempo máximo de terminación de los trabajos o *makespan* C_{max} . Formalmente, el problema estudiado se clasifica como $F || C_{max}$, acorde a la notación $\alpha|\beta|\gamma$ introducida por (Graham et al., 1979).

Para ilustrar lo expuesto anteriormente, se presenta en la figura 1 una solución para el FSSP. Los tiempos de procesamiento del FSSP con cinco trabajos (Job j) y cuatro máquinas (Machine i) se presentan en la tabla 1. La solución a este problema está dado por la secuencia $\pi_j = \{4, 2, 5, 1, 3\}$.

Machine (i)	Job (j)				
	1	2	3	4	5
1	31	19	23	13	33
2	41	55	42	22	5
3	25	3	27	14	57
4	30	34	6	13	19

Tabla 1. Ejemplo de un FSSP.

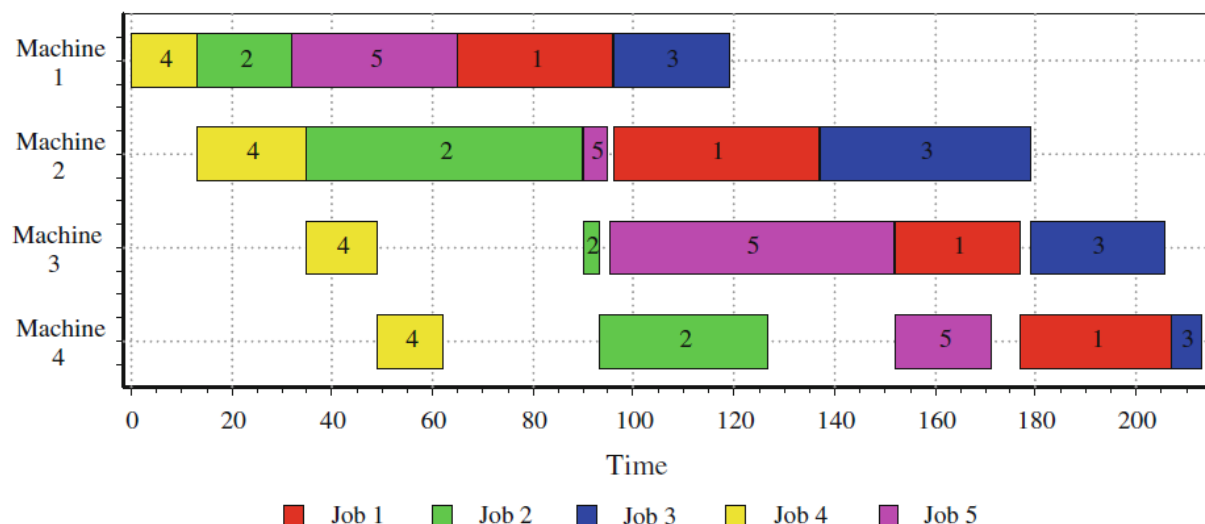


Figura 1. Diagrama de Gantt que ilustra una solución para el FSSP.

3 Metodología de Solución

Este artículo presenta dos heurísticos de búsqueda local para el FSSP. Estos heurísticos establecen dos vecindarios que se combinan estratégicamente para lograr buenas soluciones. El primer heurístico de búsqueda local se basa en el conocido 2opt, mientras que el segundo se basa en un intercambio aleatorio. Ambos parten de una solución inicial lexicográfica. A continuación, se presentan en detalle.

3.1 Heurísticos de Búsqueda Local

La búsqueda local es proceso iterativo que empieza en una solución y la mejora realizando modificaciones locales. Básicamente empieza con una solución inicial x_i y busca en su vecindad una mejor solución. Si la encuentra, reemplaza su solución actual por la nueva y continua con el proceso, hasta que no se pueda mejorar la solución actual. La vecindad son todas las soluciones que se consideran en cada paso (Ver Algoritmo 1).

Algoritmo 1 Búsqueda Local

Entrada: solución inicial construida con *orden lexicográfico* $s = x_i$

Salida: solución mejorada s .

Procedimiento Búsqueda Local

```

 $s$  = genera una solución inicial
while  $s$  no es óptimo local do
     $s' \in N(s)$  con  $f(s) < f(s')$ 
    (mejor solución  $s$  dentro de la vecindad)
     $s \leftarrow s'$ 
end
return  $s$ 

```

Algoritmo 1. Búsqueda Local

Los movimientos utilizados para llevar a cabo la búsqueda local son dos: *TwoOpt* y *Random Relocation*. El primero podríamos llamarle un movimiento de intensificación, en cambio el segundo podría ser considerado un movimiento de diversificación.

TwoOpt: se eligen dos trabajos para luego intercambiarlos. La estrategia utilizada en este vecindario es *best-improvement* (seleccionar el mejor vecino que mejora la solución). La figura 2 muestra el procedimiento de intercambio entre el Job 2 y el Job 3.

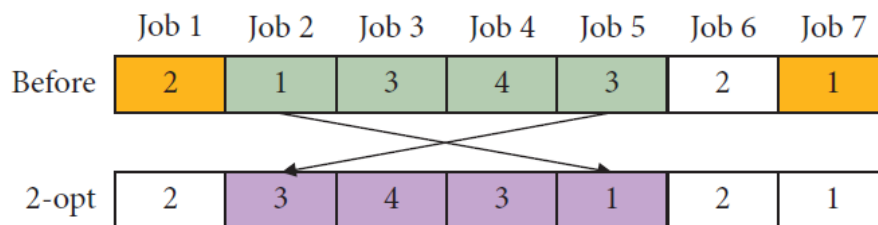


Figura 2. Ilustración del movimiento TwoOpt

Random Relocation: En este movimiento se selecciona y elimina aleatoriamente dos trabajos en cada iteración, luego se insertan aleatoriamente en otra posición. Este movimiento se efectúa durante 10.000 iteración (i.e., Número de trabajos X Número de máquinas, 500×20). Terminadas estas iteraciones se elige la mejor solución obtenida.

4 Resultados Computacionales

Las búsquedas locales se codificaron en C++ 2022 con un computador Intel(R) Core (TM) i5-4200 CPU, a 1.6 GHz y 4.00GB de RAM, y se implementaron en diez instancias de (Taillard, 1990), las cuales son accesibles desde el siguiente enlace <http://mistic.heig-vd.ch/taillard/problemes.dir/ordonnancement.dir/ordonnancement.html>. El conjunto de instancias elegidas fueron las de mayor tamaño (500 trabajos, 20 máquinas). Los resultados como el código fuente puede ser consultado en el siguiente enlace https://github.com/juliomariodaza/TAREA_2-METAHEURISTICAS.git

El análisis de los resultados consistió en tres partes: i) definir experimentos, ii) evaluación de rendimiento y iii) comparación con las bks's. La figura 3 muestra el esquema del análisis de resultados realizado. En las siguientes secciones se explicarán en detalle cada uno de los elementos que comprenden los resultados computacionales efectuados.

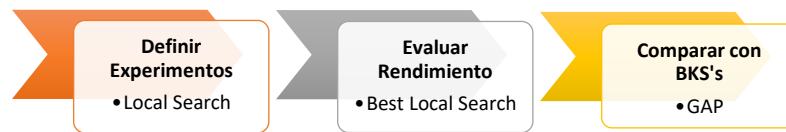


Figura 3. Esquema general de los resultados computacionales desarrollados.

4.1 Definir Experimentos

Partiendo de los dos métodos de búsqueda local diseñados, se crearon cuatro experimentos que consisten en evaluar las combinaciones que podría mejorar el makespan para cada una de las instancias. Las combinaciones analizadas para cada experimento son:

- Experimento 01: TwoOpt – Random
- Experimento 02: Random – TwoOpt
- Experimento 03: TwoOpt - Random – TwoOpt
- Experimento 04: Random - TwoOpt – Random

4.2 Evaluar Rendimientos

Una vez definidos los experimentos se procede a evaluar el rendimiento para cada caso. Esta evaluación se hace evaluando los makespan obtenidos de cada experimento con respecto al makespan de la solución inicial lexicográfica. Los resultados de la evaluación de desempeño se presentan en la Tabla 2, donde para cada instancia, se muestra el valor de la solución inicial en la columna 1, la evaluación de desempeños para los cuatro experimentos, donde cada experimento presenta su makespan, tiempo de corrida en segundos y el GAP con respecto al makespan de la solución inicial.

Instancia	Solución Inicial	Mejoramiento de la Solución Inicial con Búsqueda Local											
	Lexicográfica	Experimento 01			Experimento 02			Experimento 03			Experimento 04		
		TwoOpt - Random			Random - TwoOpt			TwoOpt - Random - TwoOpt			Random - TwoOpt - Random		
	Makespan	Makespan	Run Time	GAP	Makespan	Run Time	GAP	Makespan	Run Time	GAP	Makespan	Run Time	GAP
T500_20.1	30121	27425	12.8464	-9.83%	27473	12.7821	-9.64%	27405	23.4135	-9.91%	27107	13.8678	-11.12%
T500_20.2	31202	27928	13.3696	-11.72%	28022	12.8063	-11.35%	27778	23.8334	-12.33%	27816	14.1276	-12.17%
T500_20.3	29406	26299	12.6546	-11.81%	26160	12.6869	-12.41%	26314	23.5405	-11.75%	26254	13.8339	-12.01%
T500_20.4	30355	27649	12.4614	-9.79%	27611	12.6452	-9.94%	27565	23.9788	-10.12%	27379	13.9416	-10.87%
T500_20.5	30099	27530	12.4974	-9.33%	27542	13.0094	-9.28%	27554	23.8773	-9.24%	27246	13.8988	-10.47%
T500_20.6	30946	27730	12.4339	-11.60%	27598	12.6832	-12.13%	27845	23.7051	-11.14%	27493	13.8908	-12.56%
T500_20.7	30792	27474	12.5828	-12.08%	27531	12.7226	-11.84%	27384	23.8493	-12.45%	27220	14.251	-13.12%
T500_20.8	31034	27807	12.9563	-11.60%	27875	13.5101	-11.33%	27886	23.4638	-11.29%	27657	14.156	-12.21%
T500_20.9	30634	27489	12.4833	-11.44%	27281	12.8867	-12.29%	27567	23.8237	-11.13%	26997	13.9379	-13.47%
T500_20.10	29092	26272	12.4682	-10.73%	26484	13.1443	-9.85%	26447	23.8475	-10.00%	26180	13.8883	-11.12%
Promedios	30368.10	27360.30	12.67539	-10.99%	27357.70	12.88768	-11.90%	27374.50	23.73329	-10.94%	27134.90	13.97937	-11.92%

Tabla 2. Evaluación de desempeño de los experimentos versus la solución inicial.

En los resultados de la Tabla 2 se puede observar que en promedio los experimentos 1 y 2 son los que obtienen soluciones en menor periodo de tiempo, esto se debe a que solo usan dos vecindarios en

su búsqueda, mientras que los demás, usan tres. Además, podemos notar que los experimentos 4 y 2 presentan en promedio los mejores valores de makespan respectivamente. Esto puede deberse, quizás, a que el diversificar con Random permite llevar las soluciones a nuevo espacio de búsqueda, el cual luego es intensificado con el movimiento TwoOpt.

Teniendo en cuenta que las combinaciones efectuadas para los experimentos 2 y 4 cuentan con el mejor desempeño, tomaremos estos para compararlos con los BKS's presentes en la literatura. Esta comparación se presenta en la siguiente sección.

4.3 Comparación de las Best Know Solutions

La comparación para las mejores combinaciones (Experimentos 2 y 4) se evalúan con respecto a los BKS's en la Tabla 3. Esta muestra para cada instancia los BKS's presentes en la literatura en la columna 2 y los resultados para cada experimento evaluado, esto es, el valor de su makespan, tiempo de corrida en segundos y el GAP con respecto al BKS.

Instancias	BKS	Random - TwoOpt			Random - TwoOpt - Random		
		Makespan	Run Time	GAP	Makespan	Run Time	GAP
T500_20.1	26040	27473	12.7821	5.22%	27107	13.8678	3.94%
T500_20.2	26520	28022	12.8063	5.36%	27816	14.1276	4.66%
T500_20.3	26371	26160	12.6869	-0.81%	26254	13.8339	-0.45%
T500_20.4	26454	27611	12.6452	4.19%	27379	13.9416	3.38%
T500_20.5	26334	27542	13.0094	4.39%	27246	13.8988	3.35%
T500_20.6	26477	27598	12.6832	4.06%	27493	13.8908	3.70%
T500_20.7	26389	27531	12.7226	4.15%	27220	14.251	3.05%
T500_20.8	26560	27875	13.5101	4.72%	27657	14.156	3.97%
T500_20.9	26005	27281	12.8867	4.68%	26997	13.9379	3.67%
T500_20.10	26457	26484	13.1443	0.10%	26180	13.8883	-1.06%
Promedio	26360.70	27357.70	12.88768	3.64%	27134.90	13.97937	2.85%

Tabla 3. Evaluación de desempeño de los experimentos 2 y 4 versus los *Best Know Solutions*.

Los resultados de la Tabla 3 muestran que el experimento 4, cuya combinación corresponde a Random - TwoOpt – Random, permite alcanzar en promedios los mejores valores de makespan para el FSSP, con un 2.85% de GAP, lo cual demuestra que diversificar, intensificar y luego diversificar la búsqueda nuevamente, puede ayudar al algoritmo a encontrar buenas soluciones.

Además, logramos encontrar dos nuevos BKS para las instancias T500_20.3 y T500_20.10, mejorando el makespan en un 0.81% y 1.06% respectivamente. Estos nuevos BKS se presentan en la sección de Apéndices.

Conclusiones

En este artículo presentamos dos métodos de búsqueda local para el *Flow Shop Scheduling Problem* (FSSP). Estos heurísticos establecen dos vecindarios para el FSSP que se combinan estratégicamente para lograr buenas soluciones. El primer heurístico de búsqueda local se basa en el conocido 2opt, mientras que el segundo se basa en un intercambio aleatorio.

Los experimentos sobre las instancias de referencia de la literatura muestran que nuestros algoritmos se encuentran cercanos a los *Best Know Solutions* (BKS's) presentes en la literatura, obteniendo dos nuevos BKS's que mejoran el makespan en un 0.81% y 1.06% respectivamente.

Los algoritmos de búsqueda local desarrollados nos permiten encontrar diferentes soluciones posibles, que podrían ser usadas como puntos de partida para un algoritmo metaheurístico. Finalmente, nuestro procedimiento puede extenderse fácilmente para hacer frente a nuevas instancias del FSSP presentes en la literatura.

References

- Akyol, D. E., & Bayhan, G. M. (2007). A review on evolution of production scheduling with neural networks. *Computers and Industrial Engineering*, 53, 95–122.
- Asadzadeh, L., & Zamanifar, K. (2010). An agent-based parallel approach for the job shop scheduling problem with genetic algorithms. *Mathematical and Computer Modelling*, 52, 1957–1965.
- Campbell, H.G., R.A. Dudek and M.L. Smith. (1970). A heuristic algorithm for the n job, m machine sequencing problem. *Management Science*, 16, 630-637.
- Dannenbring, D.G. (1977). An evaluation of flow shop sequencing heuristics. *Management Science*, 23, 1174-1182.
- Grabowski, J. and M. Wodecki. (2004). A very fast tabu search algorithm for the permutation flow shop problem with makespan criterion. *Computers & Operations Research*, 31, 1891-1909.
- Graham, R.L., E.L. Lawler, J.K. Lenstra and A.H.G. Rinnooy Kan. (1979). Optimisation and approximation in deterministic sequencing and scheduling: a survey. *Annals of Discrete Mathematics*, 5, 236-287.
- Lenstra, J. K., Kan, A. H. G. R., & Brucker, P. (1977). Complexity of machine scheduling problems. *Annals of Discrete Mathematics*, 1, 343–362.
- Nawaz, M., E.E. Ensore Jr and I. Ham. (1983). A heuristic algorithm for the m-machine, njob flow-shop sequencing problem. *OMEGA, The International Journal of Management Science*, 11, 91-95.
- Nowicki, E. and C. Smutnicki. (1996), A fast tabu search algorithm for the permutation flow shop problem. *European Journal of Operational Research*, 91, 160-175.
- Ogbu, F.A. and D.K. Smith. (1990). The application of the simulated annealing algorithm to the solution of n/m/Cmax flow-shop problem. *Computers & Operations Research*, 17, 243-253.
- Osman, I.H. and C.N. Potts. (1989). Simulated annealing for permutation flow shop scheduling. *OMEGA, The International Journal of Management Science*, 17, 551-557.
- Reeves, C.R. (1993). Improving the efficiency of tabu search for machine scheduling problems. *Journal of Operational Research Society*, 44, 375-382.
- Reeves, C.R. (1995). A genetic algorithm for flowshop sequencing. *Computers & Operations Research*, 22, 5-13.
- Reeves, C.R. and T. Yamada. (1998). Genetic Algorithms, path relinking, and the flowshop sequencing problem. *Evolutionary Computation*, 6, 45-60.
- Rinnooy Kan, A.H.G. (1976). *Machine Scheduling Problems: Classification, Complexity and Computations*. Martinus Nijhoff, The Hague, The Netherlands.
- Ruiz, R. and T. Stützle. (2006). A simple and effective iterated greedy algorithm for the permutation flowshop scheduling problem. *European Journal of Operational Research*, to appear.
- Stützle, T. (1998). Applying iterated local search to the permutation flow shop problem. Technical Report, AIDA-98-04, FG Intellektik, TU Darmstadt.
- Taillard, E. (1990). Some efficient heuristic methods for the flow shop sequencing problem. *European Journal of Operational Research*, 47, 67-74.
- Taillard, E. (1993). Benchmarks for basic scheduling problems. *European Journal of Operational Research*, 64, 278-285.
- Widmer, M. and A. Hertz (1989), A new heuristic method for the flow shop sequencing problem. *European Journal of Operational Research*, 41, 186-193.
- Zhang, R., Song, S., & Wu, C. (2013). A hybrid artificial bee colony algorithm for the job shop scheduling problem. *International Journal of Production Economics*, 141, 167–178.

Apéndice

En este apéndice se presentan las nuevas BKS encontradas con la metodología propuesta.

Instancia: ../Taillard/T500_20/t500_20.3.txt

Estrategia de Mejora: RandomMove & TwoOptMove

Tiempo de corrida en segundos: 12.6869

Nuevo BKS: 26160 – **BKS Anterior:** 26371 – **Mejora:** 0.81%

Secuencia de trabajos:

181, 311, 404, 1, 3, 7, 285, 114, 444, 398, 10, 245, 460, 13, 313, 390, 421, 16, 17, 20, 19, 243, 387, 21, 22, 452, 24, 25, 287, 30, 26, 27, 354, 28, 463, 31, 32, 162, 35, 36, 37, 456, 41, 415, 302, 43, 44, 449, 149, 46, 391, 53, 54, 247, 55, 56, 57, 58, 59, 60, 62, 63, 156, 154, 40, 14, 64, 65, 434, 286, 221, 470, 91, 424, 93, 72, 73, 350, 66, 74, 160, 383, 76, 50, 319, 288, 77, 78, 475, 79, 80, 81, 372, 131, 218, 458, 459, 82, 85, 71, 368, 110, 86, 87, 216, 238, 88, 89, 112, 90, 92, 96, 98, 99, 234, 220, 438, 325, 380, 341, 252, 51, 437, 103, 105, 151, 106, 107, 108, 109, 111, 264, 115, 116, 117, 118, 119, 120, 464, 413, 122, 100, 471, 123, 49, 407, 125, 323, 129, 121, 273, 130, 132, 293, 297, 164, 450, 136, 140, 141, 142, 34, 143, 146, 147, 148, 23, 152, 153, 432, 155, 159, 157, 158, 161, 33, 198, 163, 165, 166, 167, 168, 169, 75, 171, 201, 177, 178, 179, 182, 378, 294, 97, 170, 184, 48, 187, 188, 189, 454, 191, 192, 400, 193, 195, 196, 199, 200, 134, 261, 202, 203, 248, 204, 356, 206, 207, 208, 39, 209, 226, 210, 5, 126, 212, 214, 176, 468, 101, 215, 217, 12, 219, 222, 47, 327, 223, 284, 224, 225, 227, 83, 228, 253, 229, 230, 138, 232, 231, 139, 384, 2, 236, 240, 241, 239, 246, 249, 251, 42, 254, 213, 255, 257, 260, 174, 326, 173, 265, 267, 353, 269, 403, 270, 102, 150, 276, 175, 272, 274, 237, 275, 277, 320, 279, 281, 197, 283, 258, 474, 6, 301, 289, 290, 357, 268, 291, 292, 113, 211, 295, 296, 298, 446, 186, 303, 304, 386, 305, 307, 308, 309, 312, 314, 315, 271, 316, 317, 250, 348, 318, 321, 322, 29, 324, 339, 328, 329, 330, 331, 332, 345, 360, 4, 333, 334, 335, 369, 336, 128, 371, 337, 338, 104, 425, 392, 427, 340, 342, 343, 180, 282, 344, 346, 347, 61, 349, 351, 358, 359, 361, 362, 363, 190, 280, 364, 365, 443, 366, 367, 448, 370, 94, 373, 375, 376, 377, 379, 352, 11, 385, 45, 256, 69, 235, 389, 205, 183, 310, 393, 263, 394, 396, 382, 259, 397, 9, 262, 399, 233, 401, 402, 405, 194, 406, 68, 124, 409, 410, 411, 412, 414, 127, 455, 416, 436, 417, 145, 300, 418, 419, 244, 420, 422, 423, 144, 95, 426, 428, 429, 430, 431, 433, 306, 435, 135, 70, 440, 441, 266, 442, 445, 355, 408, 447, 451, 453, 172, 18, 388, 185, 38, 52, 439, 462, 465, 466, 467, 299, 469, 472, 473, 242, 476, 457, 477, 478, 479, 461, 395, 480, 481, 278, 15, 482, 483, 381, 484, 485, 486, 374, 8, 487, 488, 489, 490, 491, 492, 493, 494, 133, 495, 84, 496, 497, 498, 499, 500, 67, 137,

Instancia: /Taillard/T500_20/ t500_20.10.txt

Estrategia de Mejora: RandomMove – TwoOptMove – RandomMove

Tiempo de corrida en segundos: 13.8883

Nuevo BKS: 26457 – **BKS Anterior:** 26180 – **Mejora:** 1.06%

Secuencia de trabajos:

313, 344, 4, 371, 7, 249, 12, 13, 14, 472, 40, 9, 15, 55, 18, 298, 19, 20, 21, 86, 5, 163, 24, 37, 27, 99, 246, 28, 30, 31, 417, 414, 338, 32, 85, 94, 139, 34, 35, 353, 146, 38, 39, 43, 45, 46, 48, 49, 426, 51, 52, 67, 57, 53, 54, 23, 56, 36, 145, 394, 58, 60, 62, 64, 65, 66, 70, 381, 194, 73, 231, 75, 76, 79, 81, 128, 82, 83, 347, 196, 84, 87, 88, 114, 90, 25, 261, 92, 322, 97, 98, 101, 434, 104, 106, 107, 47, 130, 345, 260, 77, 109, 211, 113, 116, 117, 118, 364, 200, 16, 154, 122, 447, 168, 445, 337, 123, 127, 259, 129, 132, 136, 137, 140, 156, 141, 294, 142, 143, 147, 149, 446, 197, 192, 151, 155, 157, 126, 158, 469, 159, 342, 91, 160, 161, 162, 164, 166, 50, 167, 169, 170, 221, 69, 178, 179, 180, 206, 181, 214, 244, 182, 216, 357, 183, 185, 416, 29, 189, 334, 190, 388, 191, 193, 198, 199, 336, 201, 202, 273, 316, 203, 204, 208, 96, 210, 212, 215, 144, 72, 297, 119, 3, 219, 220, 222, 442, 223, 451, 224, 8, 225, 375, 226, 133, 17, 227, 228, 233, 234, 209, 184, 330, 235, 237, 103, 100, 115, 217, 195, 238, 239, 240, 241, 443, 242, 243, 245, 205, 229, 247, 74, 248, 254, 280, 252, 279, 412, 78, 255, 256, 251, 262, 264, 265, 266, 267, 269, 270, 397, 272, 274, 275, 430, 278, 2, 304, 89, 283, 285, 277, 253, 271, 286, 287, 288, 289, 290, 291, 61, 293, 296, 365, 108, 300, 386, 301, 302, 303, 340, 305, 306, 309, 466, 310, 311, 312, 315, 125, 474, 188, 250, 318, 380, 230, 321, 258, 105, 323, 295, 281, 299, 324, 325, 326, 327, 329, 331, 332, 333, 335, 339, 341, 93, 343, 176, 346, 348, 349, 351, 352, 268, 354, 320, 410, 402, 422, 355, 356, 284, 428, 358, 314, 359, 360, 63, 361, 373, 362, 463, 363, 366, 367, 368, 369, 148, 370, 350, 372, 207, 26, 376, 152, 292, 377, 378, 165, 379, 382, 421, 383, 374, 385, 387, 389, 390, 404, 391, 6, 213, 68, 392, 393, 177, 395, 22, 398, 399, 405, 401, 406, 407, 400, 408, 409, 384, 475, 110, 411, 415, 396, 418, 102, 419, 131, 420, 135, 153, 59, 11, 423, 425, 308, 232, 427, 429, 431, 319, 432, 433, 435, 413, 33, 436, 307, 438, 439, 441, 444, 1, 448, 449, 450, 403, 452, 453, 42, 455, 456, 317, 276, 457, 41, 458, 459, 460, 461, 328, 111, 462, 440, 138, 186, 464, 465, 112, 454, 467, 171, 218, 468, 134, 470, 473, 263, 476, 10, 477, 478, 172, 150, 479, 480, 175, 121, 257, 481, 482, 424, 483, 484, 71, 485, 282, 486, 487, 174, 80, 488, 95, 44, 187, 489, 120, 490, 471, 491, 492, 493, 494, 124, 495, 437, 173, 496, 497, 498, 499, 500, 236,