

Tarea 3: Aplicación del Algoritmo TABU SEARCH a un Problema de Flow-Shop

Juan Esteban Aguirre Olarte – 202210404 – je.aguirreo1

- **Introducción**

El problema de programación de taller de flujo (Flow Shop Scheduling Problem, FSSP) es un problema de optimización combinatoria ampliamente estudiado debido a su complejidad y relevancia en la industria manufacturera. El objetivo comúnmente es minimizar el makespan (C_{\max}), es decir, el tiempo total necesario para completar todos los trabajos.

Para abordar este problema, se ha seleccionado el algoritmo de búsqueda tabú (Tabu Search, TS) debido a su capacidad para evitar óptimos locales y mejorar soluciones iterativamente. La selección de este algoritmo se basa en diversas investigaciones que han demostrado su eficacia en la resolución del FSSP (Dodu, C., & Ancău, M., 2020) y (Umam, M. S., Mustafid, M., & Suryono, S., 2021).

- **Descripción del Problema**

Formulación Matemática:

Parámetros:

n : Número de trabajos.

m : Número de máquinas.

p_{ij} : Tiempo de procesamiento del trabajo j en la máquina i .

Variables:

σ : Permutación de los trabajos.

Función Objetivo:

-Minimizar makespan C_{\max}

$$\min C_{\max}(\sigma)$$

Donde C_{\max} es:

$$C_{\max}(\sigma) = C_{m\sigma_n}$$

Y $C_{m\sigma_n}$ está dado por:

$$C_{1\sigma_1} = p_{1\sigma_1}$$

$$C_{1\sigma_j} = C_{1\sigma_{j-1}} + p_{1\sigma_j}, \quad j = 2, \dots, n$$

$$C_{k\sigma_1} = C_{k-1\sigma_1} + p_{k\sigma_1}, \quad k = 2, \dots, m$$

$$C_{k\sigma_j} = \max\{C_{k\sigma_{j-1}}, C_{k-1\sigma_j} + p_{k\sigma_j}\}, \quad j = 2, \dots, n; k = 2, \dots, m$$

- **Metodología:**

El funcionamiento de la búsqueda tabú implementada sigue la siguiente lógica:

- Generar una solución inicial mediante una heurística constructiva en este caso se intentó hacer una especie de NEH, es decir, ordenar trabajos por tiempo total

Tarea 3: Aplicación del Algoritmo TABU SEARCH a un Problema de Flow-Shop

Juan Esteban Aguirre Olarte – 202210404 – je.aguirreo1

decreciente e insertarlos en la posición que minimiza el makespan (Vallada, E., Ruiz, R., & Framinan, J. M., 2014).

- Se comienza a iterar sobre las iteraciones máximas.
- Se exploran vecindades de soluciones, previamente creadas mediante intercambios de trabajos en la secuencia. Los vecindarios utilizados fueron:
 - Inserción de trabajos.
 - Swap(1,1) intercambia un trabajo con su vecino inmediato.
 - Swap de vecinos/trabajos no adyacentes.
 - K-shift mueve un trabajo hasta k posiciones hacia adelante o atrás.
- Mantener una lista tabú de movimientos recientemente realizados para evitar volver a soluciones previas.
- Sin embargo, se aplica un criterio de aspiración que permite aceptar soluciones tabú sí y solo sí mejoran el óptimo conocido.
- Se acaba la búsqueda tabú cuando se llega al límite de iteraciones.

Es necesario aclarar que el tamaño de la lista tabú (*tenure*) y las iteraciones máximas (*max_iter*) son calibradas de forma general según el tamaño de las instancias. Por otro lado, dentro del código se diseñó una **estrategia de exploración** que consiste en cambiar el tamaño del *K*-shift después del 70% de iteraciones. Dicho cambio consiste en reducir el número de vecino al cambiar el parámetro *k* de un valor de 2 a un valor de 5.

• Resultados y Análisis de Resultados:

Tras varias pruebas para los parámetros de *tenure* y *max_iter* se llegó a la conclusión que el valor del C_{max} llega a una aparente convergencia para una calibración de (41,211). Aunque ciertamente podrían ser menos, pero habría un costo/beneficio con respecto al gap y al tiempo de ejecución. Así pues, las siguientes gráficas tratan de dar prueba de ello, sin embargo, solo se realizaron para las instancias de 20x5 debido a la demanda de tiempo.

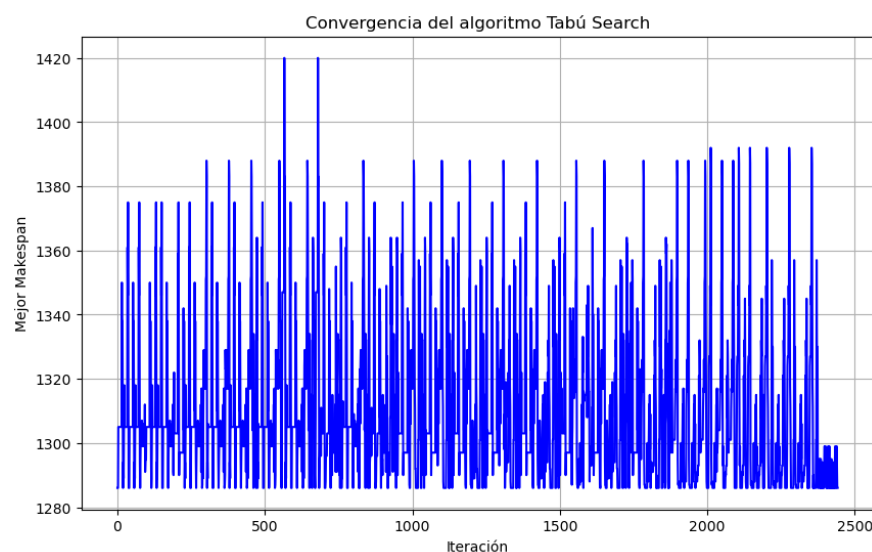


Figura 1: Convergencia del C_{max}

Tarea 3: Aplicación del Algoritmo TABU SEARCH a un Problema de Flow-Shop

Juan Esteban Aguirre Olarte – 202210404 – je.aguirreo1

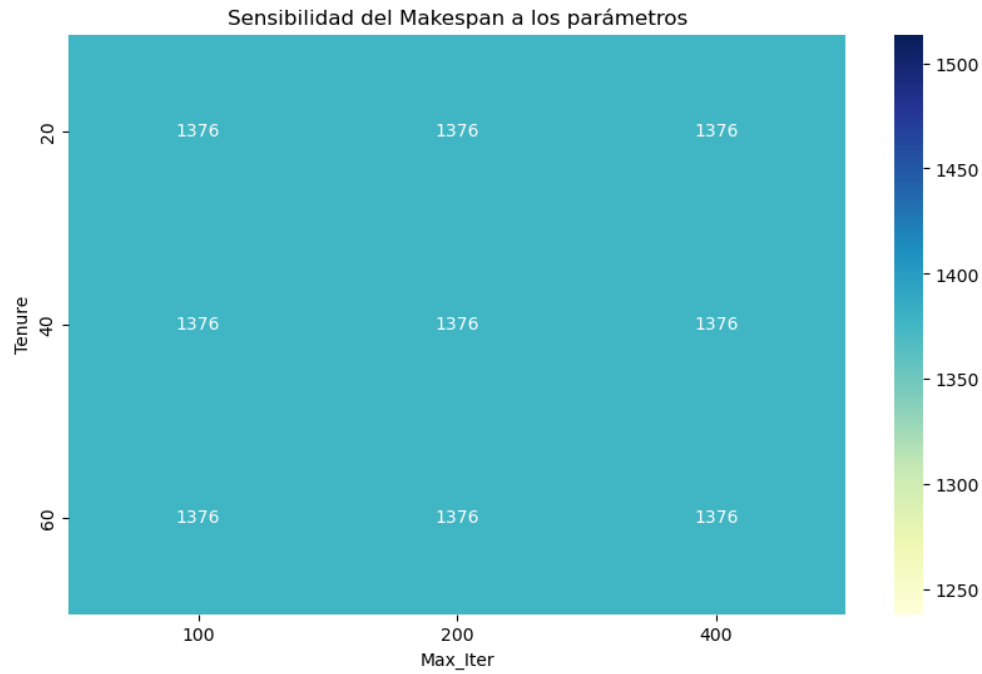


Figura 2: Sensibilidad del makespan ante los parámetros del tabú search

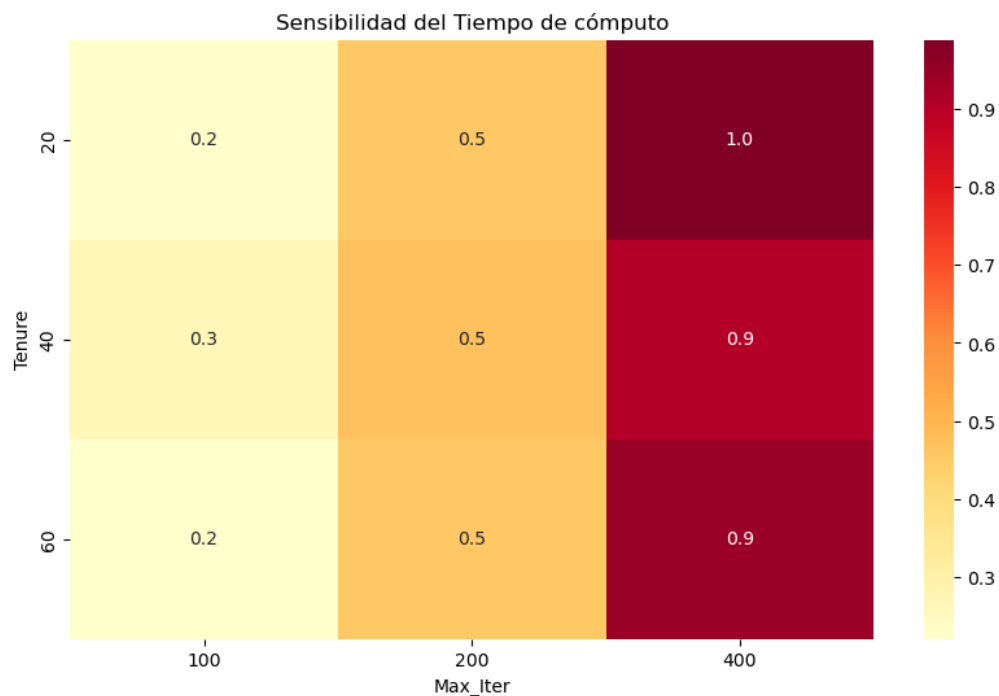


Figura 3: Sensibilidad del tiempo de cómputo ante los parámetros del tabú search

Tarea 3: Aplicación del Algoritmo TABU SEARCH a un Problema de Flow-Shop

Juan Esteban Aguirre Olarte – 202210404 – je.aguirreo1

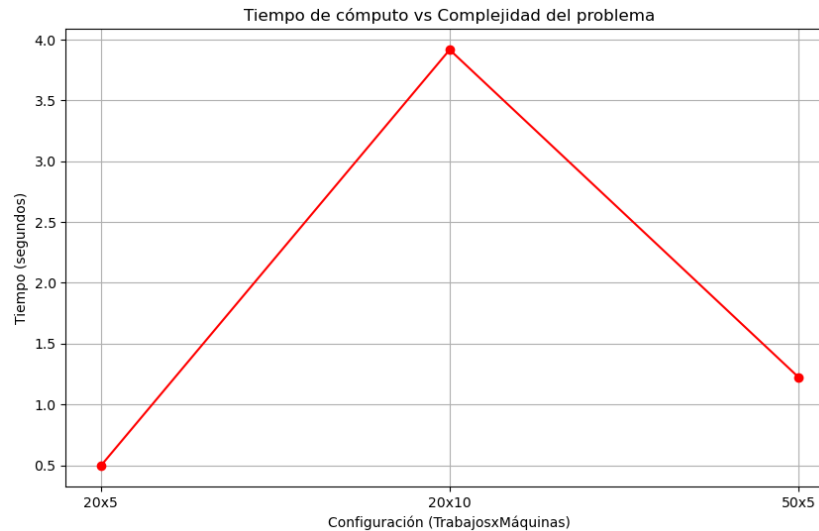


Figura 4: Tiempo de cómputo vs complejidad de la instancia

No obstante, para los valores registrados de tiempo de corrida para cada conjunto de semillas pertenecientes a un mismo número de máquinas y trabajos se concluyó que, ciertamente, el tiempo de corrida está determinado por el número de iteraciones. Así pues, la elección de cantidad y tipo de vecindarios pudo no haber sido la mejor, por lo que con solo 2 vecindarios véase de inserción y 2-opt se pudieron haber conseguido los mismos resultados con valores de *tenure* y *max_iter* más accesibles. De la misma manera, puede ser que la forma en la que se generan los números aleatorios y, posteriormente, se procesan en una lista de listas no sea la más eficaz; la razón detrás de este empaquetamiento fue que primero se pensó en el código, más no en la forma de obtención de los tiempos de procesamiento por números pseudoaleatorios.

Inst.	Cmax Tabú Search	BS	Gap	Tiempo (s)
Ta001	1286	1278	0.626%	6.849
Ta002	1365	1359	0.442%	6.714
Ta003	1098	1081	1.573%	6.509
Ta004	1299	1293	0.464%	6.345
Ta005	1258	1235	1.862%	6.535
Ta006	1210	1195	1.255%	6.303
Ta007	1251	1239	0.969%	6.273
Ta008	1206	1206	0.000%	6.341
Ta009	1253	1230	1.870%	6.324
Ta010	1108	1108	0.000%	6.917
Ta011	1583	1582	0.063%	13.395
Ta012	1676	1659	1.025%	13.383
Ta013	1519	1496	1.537%	13.312
Ta014	1391	1377	1.017%	12.323
Ta015	1444	1419	1.762%	12.399
Ta016	1401	1397	0.286%	12.876
Ta017	1526	1484	2.830%	12.657

Tarea 3: Aplicación del Algoritmo TABU SEARCH a un Problema de Flow-Shop

Juan Esteban Aguirre Olarte – 202210404 – je.aguirreo1

Ta018	1572	1538	2.211%	12.372
Ta019	1611	1593	1.130%	12.272
Ta020	1603	1591	0.754%	12.472
Ta021	2330	2297	1.437%	25.217
Ta022	2113	2099	0.667%	26.038
Ta023	2345	2326	0.817%	24.45
Ta024	2239	2223	0.720%	24.4
Ta025	2358	2291	2.924%	23.908
Ta026	2261	2226	1.572%	25.232
Ta027	2303	2273	1.320%	25.729
Ta028	2222	2200	1.000%	25.785
Ta029	2242	2237	0.224%	24.691
Ta030	2219	2178	1.882%	24.413
Ta031	2724	2724	0.000%	88.818
Ta032	2838	2834	0.141%	91.043
Ta033	2621	2621	0.000%	90.182
Ta034	2768	2751	0.618%	86.55
Ta035	2864	2863	0.035%	88.501
Ta036	2836	2829	0.247%	86.929
Ta037	2732	2725	0.257%	86.813
Ta038	2686	2683	0.112%	87.266
Ta039	2561	2552	0.353%	86.776
Ta040	2783	2782	0.036%	87.601
Ta041	3037	3021	0.530%	174.597
Ta042	2937	2902	1.206%	171.872
Ta043	2905	2871	1.184%	171.637
Ta044	3094	3070	0.782%	172.129
Ta045	3044	2998	1.534%	173.038
Ta046	3112	3024	2.910%	171.394
Ta047	3156	3122	1.089%	174.263
Ta048	3089	3063	0.849%	254.493
Ta049	2958	2914	1.510%	292.42
Ta050	3195	3076	3.869%	257.558
Ta051	3966	3874	2.375%	351.103
Ta052	3815	3734	2.169%	346.237
Ta053	3756	3688	1.844%	342.434
Ta054	3832	3759	1.942%	345.888
Ta055	3706	3644	1.701%	345.914
Ta056	3792	3717	2.018%	353.232
Ta057	3832	3728	2.790%	354.336
Ta058	3840	3730	2.949%	352.046
Ta059	3855	3779	2.011%	349.904
Ta060	3836	3801	0.921%	349.453

Tarea 3: Aplicación del Algoritmo TABU SEARCH a un Problema de Flow-Shop

Juan Esteban Aguirre Olarte – 202210404 – je.aguirreo1

Ta061	5495	5493	0.036%	666.85
Ta062	5284	5268	0.304%	666.942
Ta063	5213	5175	0.734%	664.927
Ta064	5021	5014	0.140%	655.599
Ta065	5255	5250	0.095%	884.456
Ta066	5135	5135	0.000%	672.317
Ta067	5252	5246	0.114%	668.025
Ta068	5098	5094	0.079%	667.153
Ta069	5448	5448	0.000%	666.373
Ta070	5322	5322	0.000%	667.94
Ta071	5805	5770	0.607%	1341.22
Ta072	5364	5358	0.112%	1341.255
Ta073	5763	5676	1.533%	1340.68
Ta074	5826	5792	0.587%	1337.063
Ta075	5537	5467	1.280%	1333.168
Ta076	5308	5311	-0.056%	1348.056
Ta077	5677	5605	1.285%	1376.791
Ta078	5694	5617	1.371%	1347.923
Ta079	5979	5877	1.736%	1354.225
Ta080	5903	5845	0.992%	1348.385
Ta081	6395	6303	1.460%	2779.723
Ta082	6446	6266	2.873%	2873.23
Ta083	6470	6351	1.874%	2965.176
Ta084	6500	6360	2.201%	2863.781
Ta085	6562	6408	2.403%	4066.198
Ta086	6575	6453	1.891%	781.511
Ta087	6505	6332	2.732%	776.401
Ta088	6687	6482	3.163%	778.626
Ta089	6569	6343	3.563%	784.889
Ta090	6603	6506	1.491%	778.818
Ta091	10942	10885	0.524%	3036.974
Ta092	10597	10495	0.972%	1527.319
Ta093	11025	10941	0.768%	1810.768
Ta094	10911	10889	0.202%	1518.135
Ta095	10537	10524	0.124%	12378.009
Ta096	10423	10346	0.744%	1548.666
Ta097	10939	10866	0.672%	1538.185
Ta098	10828	10741	0.810%	1553.751
Ta099	10505	10451	0.517%	1533.747
Ta100	10733	10684	0.459%	1556.192
Ta101	11471	11339	1.164%	1885.146
Ta102	11557	11344	1.878%	1907.651
Ta103	11724	11445	2.438%	1902.3

Tarea 3: Aplicación del Algoritmo TABU SEARCH a un Problema de Flow-Shop

Juan Esteban Aguirre Olarte – 202210404 – je.aguirreo1

Ta104	11685	11434	2.195%	1902.548
Ta105	11557	11369	1.654%	1905.724
Ta106	11528	11292	2.090%	1985.276
Ta107	11759	11481	2.421%	1977.102
Ta108	11780	11442	2.954%	1984.925
Ta109	11569	11313	2.263%	2272.289
Ta110	11778	11424	3.099%	1083.596
Ta111	27968	26228	6.634%	7141.64
Ta112	28650	26688	7.352%	7251.488
Ta113	28245	26522	6.496%	7294.304
Ta114	28338	26586	6.590%	7312.078
Ta115	27953	26541	5.320%	7326.298
Ta116	28326	26582	6.561%	7315.999
Ta117	28160	26660	5.626%	7391.894
Ta118	28431	26711	6.439%	7647.873
Ta119	28086	26148	7.412%	8065.818
Ta120	28427	26611	6.824%	8308.828

Por otro lado, la poca eficiencia en tiempo de computo se pudo deber al hecho que la solución inicial fue creada con una heurística constructiva como lo es el NEH, por lo que la mayor parte del tiempo está dada por ese proceso inicial.

Cabe aclarar que los valores para las instancias de **Ta086 a Ta120** se realizaron con la calibración (11,41) esto pues por cuestiones de tiempo computacional, y para enfatizar el hecho que la complejidad temporal la establece el número de iteraciones (generando así un trade-off). Por otro lado, valores atípicos en los tiempos de computo se pueden deber a que la maquina entraba en modo suspensión mientras el código corría.

- **Referencias:**

Dodu, C., & Ancău, M. (2020). "A Tabu Search Approach for Permutation Flow Shop Scheduling". *Studia Universitatis Babeş-Bolyai Informatica*, 65(1), 104-115. <https://doi.org/10.24193/subbi.2020.1.08>

Umam, M. S., Mustafid, M., & Suryono, S. (2021). A hybrid genetic algorithm and tabu search for minimizing makespan in flow shop scheduling problem. *Journal Of King Saud University - Computer And Information Sciences*, 34(9), 7459-7467. <https://doi.org/10.1016/j.jksuci.2021.08.025>

Vallada, E., Ruiz, R., & Framinan, J. M. (2014). New hard benchmark for flowshop scheduling problems minimising makespan. *European Journal Of Operational Research*, 240(3), 666-677. <https://doi.org/10.1016/j.ejor.2014.07.033>

Zobolas, G., Tarantilis, C., & Ioannou, G. (2008). Minimizing makespan in permutation flow shop scheduling problems using a hybrid metaheuristic algorithm. *Computers & Operations Research*, 36(4), 1249-1267. <https://doi.org/10.1016/j.cor.2008.01.007>