V Congreso de Ingeniería de Organización
Valladolid-Burgos, 4-5 Septiembre 2003

Índice

Algoritmos de hormigas para un problema de equilibrado de líneas.

Joaquín Bautista Valhondo[1], Jordi Pereira Gude[2]
[1] Departamento de Organización de Empresas. UPC (Doctor Ingeniero Industrial, ETSEIB, joaquin.bautista@upc.es)
[2] Departamento de Organización de Empresas. UPC (Ingeniero en Organización Industrial, ETSEIB, jorge.pereira@upc.es)

RESUMEN

El problema de equilibrado de líneas de montaje con forma de U (UALBP) es una extensión del problema de equilibrado de líneas simple (SALBP) en que la configuración física de la línea de montaje tiene una forma de U en vez de rectilínea. En el presente trabajo se presenta el problema y sus diferencias con el problema clásico de líneas de montaje, y se adapta un algoritmo basado en la optimización por colonias de hormigas, [2], para su resolución. Finalmente se muestran los resultados de la implementación del algoritmo presentado con un conjunto de instancias de la bibliografía.

***Palabras clave**: Equilibrado de líneas de montaje, UALBP, algoritmos de hormigas.*

1. Introducción.

El problema de equilibrado de líneas de montaje ALBP (Assembly Line Balancing Problem) consiste en la asignación de un conjunto de tareas en que se descompone el ensamblado de un producto entre las diferentes estaciones de trabajo en que se divide su montaje en una línea de producción. Según una clasificación propuesta por Baybars [4], los problemas de equilibrado de líneas de montaje pueden agruparse en dos categorías: SALBP (Simple Assembly Line Balancing Problem) y GALBP (General Assembly Line Balancing Problem). El primer tipo de problema (SALB) consiste en diseñar un conjunto de estaciones de trabajo (cada una de ellas asignada a un operario, grupo de trabajo o robot), con idéntico tiempo de ciclo o tasa de producción, a partir de un conjunto de tareas elementales con duraciones preestablecidas. Cada tarea sólo es asignable a una estación y éstas pueden presentar entre sí relaciones de precedencia. El problema puede verse como una generalización del problema de Bin-Packing a cuya formulación se le añaden restricciones de precedencia. El objetivo del problema admite tres variantes: (1) minimizar el número de estaciones para una tasa de producción prefijada, problema conocido como SALBP-1; (2) minimizar el tiempo de ciclo asignado a cada operario para un número de estaciones establecido, denominado SALBP-2; y (3) minimizar el tiempo muerto asignado a cada operario, equivalente a maximizar la eficiencia de la línea, para un número de estaciones enmarcado entre un número mínimo y máximo: SALBP-E.

La segunda categoría, GALBP, incluye el resto de problemas. Dentro de esta categoría entran los problemas que tienen en cuenta otras restricciones adicionales, como estaciones en paralelo, [5], agrupaciones de tareas, [6], incompatibilidades entre tareas [1], así como otras funciones objetivo. Un caso específico de este problema es el problema de equilibrado de líneas de montaje en forma de U, conocido en la literatura como problemas UALBP, [14], en

que los operarios, grupos de trabajo o robots, pueden encargarse de tareas no consecutivas en la ruta natural que sigue el flujo productivo (véase figura 1).

Para la resolución de ambas clases de problemas, destaca el uso de procedimientos heurísticos *greedy* basados en reglas de prioridad que condicionan el orden de asignación de las tareas a las estaciones de trabajo, [10]. Las reglas se refieren o combinan aspectos tales como el tiempo de proceso de cada tarea, número de tareas siguientes a una concreta, cotas sobre el número mínimo de estaciones necesario para completar la asignación de las tareas, etc. Una segunda clase de heurísticas constructivas es la constituida por las metaheurísticas GRASP (Greedy Randomized Adaptative Search Procedure) y los algoritmos basados en hormigas, que permiten generar distintas soluciones gracias a la incorporación del azar al procedimiento constructivo mostrado anteriormente. En cada iteración del procedimiento Greedy se establece una selección de tareas entre el conjunto de tareas candidatas y las tareas seleccionadas se someten a un sorteo que puede depender o no de una regla de prioridad y de la información de soluciones obtenidas anteriormente. Finalmente, una tercera clase de heurísticas se conforma por métodos de búsqueda local o procedimientos de exploración de entornos; entre ellos se encuentran: los procesos de escalado (HC), el recocido simulado (SA), la búsqueda tabú (TS) y los algoritmos genéticos (GA), véase [3]. Esta clase de metaheurísticas proporciona vías alternativas para buscar soluciones en un espacio delimitado por la definición de un vecindario. Por otra parte, la resolución exacta de los problemas de equilibrado se ha abordado mediante diversos procedimientos, entre ellos los basados en branch and bound, [11] y [13].

El resto del presente trabajo se estructura así: en el apartado 2 se presentará el problema de equilibrado de líneas de montaje con forma de U; el apartado 3 se centrará en los procedimientos heurísticos constructivos para su resolución; el apartado 4 versa sobre la adaptación de la metaheurística ACO, Ant Colony Optimization, [7], al problema de diseño de líneas de montaje con forma de U; en el apartado 5 se mostrará los resultados obtenidos por una implementación de la heurística presentada para finalmente mostrar las conclusiones del presente trabajo.

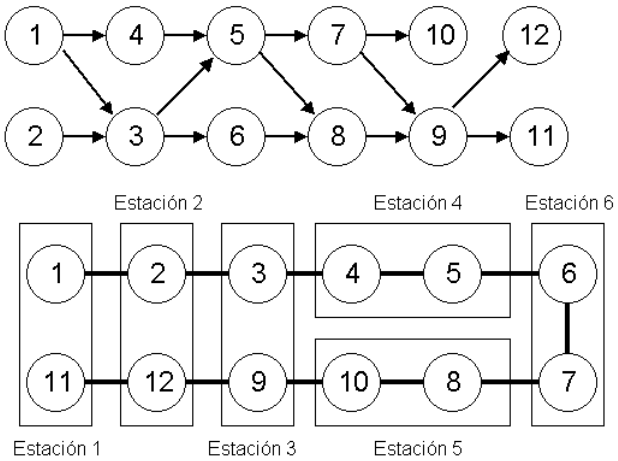

Figura 1: Ejemplo de línea con forma de U.

2. El problema de equilibrado de líneas de montaje con forma de U.

La diferencia entre los problemas de equilibrado en líneas con forma de U y los problemas tradicionales se basa en las relaciones de precedencia. En un problema en que la línea de

ensamblado tiene forma de U las tareas pueden asignarse si todas sus sucesoras y/o predecesoras han sido ya asignadas, respecto al caso simple en que la asignación de una tarea es posible sólo si todas las tareas predecesoras han sido asignadas. Debido a esta relajación de las relaciones de precedencia, la eficiencia de una línea con forma de U puede ser superior al formato de línea tradicional. Un ejemplo de línea con forma de U, y una asignación de tareas a sus estaciones (una solución) puede verse en la figura 1.

Formalmente, el problema de equilibrado de líneas de montaje en forma de U cuyo objetivo es la minimización del número de estaciones, UALBP-1 consiste en, dado un conjunto de tareas con duraciones deterministas y relaciones de precedencia y sucesión entre las tareas, encontrar el número de estaciones necesarias, y una asignación de tareas a estas estaciones tales que la suma de la duración de las tareas asignadas a cada estación no supere un valor del tiempo de ciclo, y todas las tareas asignadas a una estación tenga todas las tareas siguientes o precedentes asignadas a estaciones anteriores.

3. Heurísticas constructivas para el problema UALBP-1.

En la figura 2 se presenta el esquema de un procedimiento heurístico constructivo o greedy para generar una solución al problema UALBP-1.

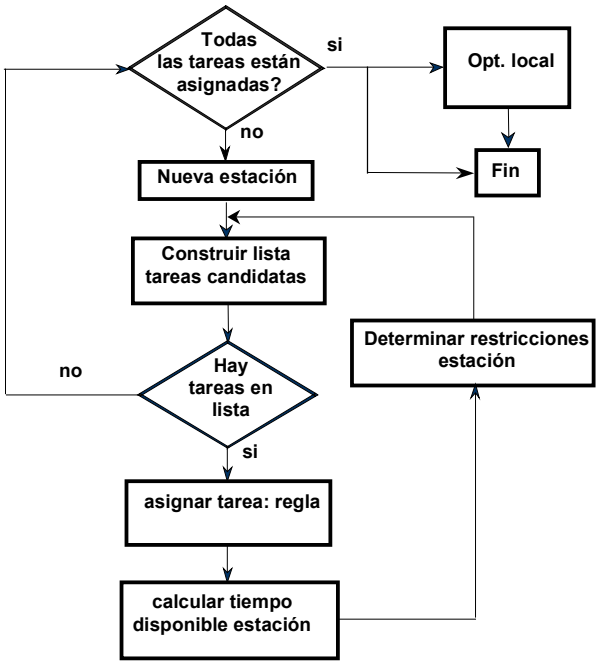

Figura 2:Esquema del procedimiento de obtención de una solución para un problema UALBP-1

La heurística efectúa la asignación, una a una, de la tarea que presenta mejor valor del índice asociado a la regla (o reglas) de prioridad que caracteriza al algoritmo; la selección se realiza entre un conjunto de tareas compatibles (con la parte de la solución construida) que satisfacen las restricciones de precedencia y/o sucesión y tiempo, hasta que todas las tareas han sido asignadas a alguna estación. Posteriormente, se puede aplicar o no un procedimiento de optimización local a la solución hallada. Según los resultados obtenidos para el problema SALBP-1 y una clase de problemas GALBP-1, véase [2], parece no aconsejable el uso de un

procedimiento de mejora local para los problemas de equilibrado de líneas con minimización del número de estaciones, por ello no se utilizará durante el presente trabajo.

Evidentemente, para definir una heurística es necesario fijar al menos una regla de prioridad en el proceso de selección; en el Apéndice I (tabla 2) se presenta una muestra compuesta por 13 reglas habituales en la literatura sobre el tema, a las que puede añadirse el sorteo aleatorio.

4. Algoritmos de colonia de hormigas para el UALBP-1.

A lo largo de su desarrollo, la metaheurística ACO, desde sus inicios [7] hasta las últimas implementaciones [15], ha tenido diversas modificaciones. El presente trabajo parte de la heurística AS presentada en [8] a las que se añaden algunas modificaciones mostradas en [2]. Básicamente la metaheurística (véase los trabajos anteriores para una descripción más detallada) construye en cada iteración un conjunto de soluciones utilizando el procedimiento mostrado en la figura 2, posteriormente, la mejor de ellas es seleccionada para depositar rastro que será utilizado en las siguientes iteraciones para modificar las probabilidades de selección encadenada de las tareas, iterando estos pasos hasta que se alcance una condición de final.

A lo largo del proceso constructivo, en cada iteración, m hormigas, que denominaremos subcolonia, construyen m soluciones al problema. Las hormigas escogen las tareas a través del procedimiento constructivo mostrado en la figura 2, donde se utiliza una regla de selección probabilística basada en la información heurística aportada por la regla de prioridad utilizada y el rastro depositado por hormigas anteriores que han encontrado buenas soluciones. La información heurística, denotada por η_i donde i es la tarea, y la información de rastro, denotada por τ, se comportan como indicadores de la calidad asociada a seleccionar una tarea en el paso constructivo en curso. Como se cuenta con más de una regla heurística en la literatura para el problema, se opta por que cada una de las m hormigas que forman una iteración del algoritmo utilice una heurística propia para la resolución del problema, y aprovechando la inversibilidad del problema, unas hormigas resuelvan el problema directo y otras el inverso. Como se cuenta con 13 reglas heurísticas de la literatura, y se utiliza dos hormigas por heurística, una para resolver el ejemplar directo y otra para resolver el ejemplar inverso, cada iteración constará de 26 hormigas.

Se han probado tres políticas para gestionar la información de rastro utilizada y en qué características de la solución se deposita la información, y dos formas de leer esta información. Las tres políticas de depósito de rastro son: (1) el rastro se deposita entre parejas de tareas correlativas en la asignación $(i,i+1)$ (τ_{ij} equivale al rastro existente entre la tarea i asignada en último lugar y la tarea candidata j), y al que denominaremos rastro entre parejas de tareas; (2) el rastro se deposita entre la tarea y la posición en que se ha escogido la tarea (τ_{ij} equivale al rastro existente entre la posición de asignación i y la tarea candidata j), que denominaremos rastro entre tarea y posición; y (3) el rastro se deposita entre la tarea y la estación a la que se ha asignado la tarea (τ_{ij} equivale al rastro existente entre la estación i y la tarea candidata j), que denominaremos rastro entre tarea y estación. Para compartir la información provista por el rastro entre un problema resuelto de forma directa y otro de forma inversa, el rastro depositado en el caso inverso sigue la siguiente política de depósito: (1) el rastro se deposita entre parejas escogidas correlativamente $(i+1,i)$; (2) el rastro se deposita

entre la tarea y el número de tareas menos la posición en que se ha asignado la tarea; y (3) el rastro se deposita entre la tarea y la cota del número de estaciones necesarias menos la estación en que se ha asignado la tarea.

Adicionalmente es necesario definir la forma en que se leerá el rastro anteriormente depositado. Se proponen dos formas de leer la información: (1) lectura directa sobre la matriz de rastro, que corresponde a la evaluación directa; y (2) lectura y cálculo acumulado sobre la matriz de rastro de la información, tal como en [12], que denominaremos evaluación acumulativa.

En estas condiciones, la probabilidad de que la tarea *j*, sea escogida entre un conjunto *D* de tareas candidatas se determina, en caso de evaluación directa, mediante:

$$p_{ij} = \frac{[\tau_{ij}]^{\alpha}[\eta_j]^{\beta}}{\sum_{h\in D}[\tau_{ih}]^{\alpha}[\eta_h]^{\beta}} \qquad (1)$$

Dónde τ_{ij} y τ_{ih} corresponderá a uno de los tres tipos de rastro y η_j, η_h al valor que ofrece la aplicación de la regla heurística para la tarea, considerando el mejor valor ofrecido al aplicarla al ejemplar directo e inverso. En el caso de evaluación acumulativa, la probabilidad se determina así:

$$p_{ij} = \frac{\left(\sum_{k=1}^{i}[\tau_{kj}]\right)^{\alpha}\cdot[\eta_j]^{\beta}}{\sum_{h\in D}\left(\sum_{k=1}^{i}[\tau_{kh}]\right)^{\alpha}\cdot[\eta_h]^{\beta}} \qquad (2)$$

Donde *k* representa, dependiendo de la política de gestión de rastro utilizada, las tareas previamente asignadas, las posiciones anteriores a la actual o las estaciones ya existentes en la solución.

En ambos casos, α y β son constantes que determinan la influencia relativa de la información de rastro y los valores heurísticos, respectivamente, en la decisión de las hormigas.

Por otra parte, la forma de utilizar el rastro tiene en cuenta la resolución de un ejemplar directo o inverso y la forma de almacenamiento del rastro en el caso inverso; por ejemplo para el caso de rastro entre tarea y estación acumulativo en el ejemplar inverso, la matriz de rastros se lee de forma que el rastro existente para asignar la tarea *j* a la estación i, τ_{ij} corresponde a la suma del rastro existente para la tarea *j* entre la estación (*cota_superior-i*) y la estación (*cota_superior*).

La actualización de los rastros se realiza después de que cada subcolonia de hormigas haya construido y evaluado todas las soluciones construidas por sus hormigas, depositando rastro solamente la mejor solución encontrada en la subcolonia para el ejemplar directo y el inverso, siguiendo el esquema de hormiga elitista expuesto en [9] para el algoritmo ACS. Para cada tarea *j*, la cantidad de feromona depositada se determina según la expresión (3), que depende de la solución encontrada y de la mejor solución conocida hasta el momento para el ejemplar.

$$\tau_{ij} \leftarrow \tau_{ij} + \frac{c\_s\_fo}{fo\_solucion} \cdot \rho \qquad (3)$$

En la expresión (3), *c_s_fo* representa el valor de función obtenido por la hormiga, *fo_solucion* el valor de la mejor solución obtenida hasta el momento y el parámetro ρ determina el porcentaje de evaporación utilizado para evitar la convergencia prematura (4). El rastro se deposita entre la tarea *j* e *i*, donde *i* es la tarea asignada previamente en caso que el rastro se deposite entre tareas, la posición en que se ha asignado la tarea en caso que el rastro se deposite entre una tarea y su posición, o la estación a la que se ha asignado en caso que el rastro se deposite entre tarea y estación. Para evitar problemas de convergencia, antes de la deposición de feromona, se aplica una evaporación de rastro según la expresión (4).

$$\tau_{ij} \leftarrow (1-\rho) \cdot \tau_{ij} \qquad (4)$$

El algoritmo se ejecuta hasta que se alcanza una condición de final, por ejemplo, número de iteraciones o tiempo de cálculo transcurrido.

5. Experiencia computacional.

Para analizar el comportamiento de los procedimientos propuestos, comparamos éstos con las soluciones presentes en la colección de problemas presentada por Scholl y disponible en la página web http://www.bwl.tu-darmstadt.de/bwl3/forsch/projekte/alb/index.htm del mismo autor y formada por un total de 282 ejemplares. A continuación se muestra una tabla resumen, tabla 1, con las diferencias respecto al mejor valor conocido de la solución y el número de óptimos obtenidos para cada tipo de algoritmo, fijando la condición de final de ejecución del algoritmo en un minuto y cinco minutos y tres combinaciones de parámetros (α=0.75, β=0.25 y ρ=0.1, denotada como 75-25 en la experiencia computacional, en el primer caso, α=0.50, β=0.50 y ρ=0.1, denotada como 50-50, en el segundo, y finalmente α=0.25, β=0.75 y ρ=0.1, denotada como 25-75). A continuación se analizarán las diferencias entre las diferentes políticas de gestión de la información de rastros, denotada por ES, estación, PO, posición, ó PR, pareja de tareas, la utilización directa o acumulativa, PD, directa, ó PA, acumulativa, de los rastros y el tiempo ejecución.

Como se desprende a partir los resultados que se muestran a continuación, los algoritmos mejoran levemente sus resultados al aumentar el tiempo de su ejecución, aunque no de forma significativa, siendo los algoritmos implementados válidos para la obtención de soluciones de calidad en tiempos de cálculo reducidos. Respecto a las diferentes alternativas expuestas, parece que aquellos algoritmos que hacen un mayor uso del rastro respecto a los valores heurísticos obtienen mejores resultados, así como que los procedimientos que utilizan la lectura de rastros directa proveen mejores soluciones para el algoritmo. Finalmente entre las diferentes maneras de depositar el rastro, el depósito en la posición de la secuencia en que se ha seleccionado una tarea es la que reporta mejores resultados, aunque el mayor número de óptimos se obtiene mediante el depósito de rastro entre parejas escogidas consecutivamente con lectura directa.

	Desv.mejor % (1 min)	Desv.max. Est. (1 min)	Óptimos	Desv.mejor % (5 min)	Desv.max. Est. (5 min)	Óptimos
ESPA-25-75	0.84	3	139	0.84	3	139
ESPA-50-50	0.80	3	154	0.80	3	154
ESPA-75-25	0.72	3	163	0.72	3	166
ESPD-25-75	0.72	2	156	0.69	2	159
ESPD-50-50	0.71	3	168	0.66	3	173
ESPD-75-25	0.63	3	174	0.60	3	180
POPA-25-75	0.83	3	141	0.83	3	141
POPA-50-50	0.77	3	159	0.75	3	160
POPA-75-25	0.73	3	165	0.71	3	165
POPD-25-75	0.67	3	164	0.66	3	166
POPD-50-50	0.67	3	167	0.62	2	169
POPD-75-25	0.65	3	178	0.62	2	180
PRPA-25-75	0.76	3	167	0.73	3	175
PRPA-50-50	0.67	3	175	0.65	3	183
PRPA-75-25	0.68	3	181	0.62	2	185
PRPD-25-75	0.75	3	167	0.69	3	167
PRPD-50-50	0.68	3	170	0.67	3	178
PRPD-75-25	0.63	2	181	0.62	2	185
MEJOR				0.56	2	194

Tabla 1:Resultados obtenidos para la colección de instancias UALBP-1 de la literatura

6. Conclusiones.

El presente trabajo propone una serie de procedimientos basados en la metaheurística ACO para los problemas de equilibrado de líneas de montaje en forma de U. Una vez analizadas las heurísticas constructivas para el problema, se comparan diferentes políticas de gestión de la información depositada por las hormigas, su lectura y el uso de diferentes reglas de prioridad en el procedimiento. Para verificar la validez de los algoritmos presentados, se ha resuelto una colección de ejemplares de referencia del problema con resultados satisfactorios.

Referencias

[1] Agnetis, A., A. Ciancimino, M. Lucertini y M. Pizzichella (1995) "Balancing Flexible Lines for Car Components Assembly". *International Journal of Production Research* 33, 333-350.

[2] Bautista J., J. Pereira, (2002) "Ant Algorithms for Assembly Line Balancing*", Lectures Notes of Computer Science*, 2463, p.65-72.

[3] Bautista, J., R. Suárez, M. Mateo y R. Companys. (2000) "Local Search Heuristics for the Assembly Line Balancing Problem with Incompatibilities Between Tasks", *in Proceedings of the 2000 IEEE International Conference on Robotics and Automation, ICRA 2000*, 2404-2409.

[4] Baybars, I. (1986) "A survey of exact algorithms for the simple assembly line balancing problem". *Management Science*, 32 (8) 909-932.

[5] Daganzo, C.F y D.E. Blumfield Assembly System Design Principles and Tradeoffs, International Journal of Production Research (1994) 32, 669-681

[6] Deckro, R.F. Balancing Cycle Time and Workstations. IIE Transactions (1989) 21, 106-111

[7] Dorigo M., V. Maniezzo y A. Colorni "The Ant System: An autocatalytic optimizing process". *Thechnical Report 91-016 Revised, Dipartimento di Electronica*, Politecnico di Milano, Italy (1991).

[8] Dorigo M., V. Maniezzo y A. Colorni The Ant System: Optimization by a colony of cooperating agents. IEEE Transactions on Systems, Man., and Cybernetics – Part B, (1996) 26(1) 29-41

[9] Dorigo M. & L.M. Gambardella (1997). Ant Colonies for the Traveling Salesman Problem. BioSystems, 43:73-81

[10] Hackman, S.T. M.J. Magazine y T.S. Wee (1989) "Fast, Effective Algorithms for Simple Assembly Line Balancing Problems". *Operations Resarch 37*, 916-924.

[11] Hoffmann, T.R. Eureka. A hybrid system for assembly line Balancing. Management Science, (1992) 38 (1), 39-47.

[12] Merkle, D., M. Middedorf, H. Schmeck Ant Colony Optimization for Resource Constrained Project Scheduling. GECCO-2000. (2000)

[13] Scholl, A. (1999) "Balancing and sequencing of assembly lines". *Physica, Heidelberg. 2nd ed*.

[14] Scholl, A. Klein R. (1999) "ULINO: Optimally Balancing U-Shaped JIT Assembly Lines", *International Journal of Production Research*, 37, p.721-736.

[15] Taillard É. D., Ant Systems, in: P. Pardalos y M. G. C. Resende (eds.), Handbook of Applied Optimization, Oxford Univ. Press, (2002) 130—137.

Apéndice I: Reglas de prioridad

Nomenclatura

i, j	Índices de tarea
N	Número de tareas
C	Tiempo de ciclo
t_i	Duración de la tarea i
IS_i	Conjunto de tareas siguientes inmediatas a la tarea i
S_i	Conjunto de tareas siguientes a la tarea i
TP_i	Conjunto de tareas precedentes a la tarea i
L_i	Nivel de la tarea i en el grafo de precedencias
Z	Tareas pendientes de asignar.

Asignar la tarea z^*: $v(z^*)=max_{i \in Z}[v(i)]$

Nombre	*Regla*
1.*Longest Processing Time*	$v(i) = t_i$
2.*Greatest Number of Inmediate Successors*	$v(i) = \| IS_i \|$
3.*Greatest Number of Successors*	$v(i) = \| S_i \|$
4.*Greatest Ranked Positional Weight*	$v(i) = t_i + \sum t_j \ (j \in S_i)$
5.*Greatest Average Ranked Positional Weight*	$v(i) = (t_i + \sum t_j \ (j \in S_i)) \ / \ (\| S_i \| + 1)$
6.*Smallest Upper Bound*	$v(i) = -UB_i = -N - 1 + [(t_i + \sum t_j \ (j \in S_i))/C]^+$
7.*Smallest Upper Bound / Number of Successors*	$v(i) = -UB_i \ / \ (\| S_i \| + 1)$
8.*Greatest Processing Time / Upper Bound*	$v(i) = t_i / UB_i$
9.*Smallest Lower Bound*	$v(i) = -LB_i = -\ [\ (t_i + \sum t_j \ (j \in TP_i)) \ / \ C\ ]^+$
10. *Minimum Slack*	$v(i) = -\ (UB_i - LB_i)$
11.*Maximum Number Successors / Slack*	$v(i) = \| S_i \| \ / \ (\ UB_i - LB_i\ )$
12.*Bhattcharjee & Sahu*	$v(i) = t_i + \| S_i \|$
13. *Etiquetas Kilbridge & Wester*	$v(i) = -\ L_i$

Tabla 2. Tabla de reglas heurísticas utilizadas en los procedimientos de resolución