

Reasignación de Tareas a Operarios en una Estación de Trabajo: un Enfoque Evolutivo

Sergio Manuel Ramírez Campos
sramirez@its.mx

*Instituto Tecnológico de Saltillo
Departamento de Ingeniería Industrial
Saltillo, Coahuila, México*

Eduardo Marroquín Prado
emarro@its.mx

Juan de Dios Córdova García
jcordova@its.mx

Resumen

La problemática que se presenta se originó a partir de una situación real y sus características lo ubican dentro de los sistemas de líneas de ensamble con flujo unidireccional, estaciones de trabajo en serie y un mismo orden en la secuencia para todos los productos. Adicionalmente se consideran restricciones de precedencia, estaciones de trabajo diferentes, número de operarios y de tareas no limitado a uno por estación. Específicamente se considera la habilidad del operador y los tiempos de desplazamiento de una tarea a otra. Este escenario resulta altamente complejo en la medida que el número de tareas y de estaciones se incrementa. Se diseñó un algoritmo genético simple (AGS) para determinar el menor número de operarios en cada una de las estaciones de trabajo de una línea de ensamble sujeta a perturbaciones que obligan a reducir el tiempo de ciclo para mantener o incrementar la productividad. Bajo esta exigencia, se reasignan las tareas buscando el menor número de operarios en cada estación de trabajo sin exceder el tiempo de ciclo máximo permitido.

Palabras clave: perturbaciones, algoritmo genético, línea de ensamble, AGS.

Introducción

Una línea de ensamble típicamente consta de varias estaciones de trabajo cada de las cuales debe llevar a cabo un conjunto específico de tareas. El producto permanece en cada estación de trabajo lo que se conoce como tiempo de ciclo, el cual corresponde al intervalo de tiempo entre partes sucesivamente terminadas¹.

El problema de balanceo de líneas de ensamble cuyas siglas en inglés son SALBP-1 trata el caso de minimizar el número de estaciones de trabajo, en donde cada estación debe realizar un cierto número de operaciones antes de continuar en la siguiente estación de la línea y donde además pueden procesarse modelos únicos o una mezcla de ellos². En este artículo se asume que ya está decidido el número de estaciones de trabajo y que ahora se desea reducir el número de operarios asignado a cada una de las estaciones de trabajo.

Este trabajo es una continuación del estudio realizado por Ramírez et al.³, quienes plantean un algoritmo genético simple (AGS) para determinar el menor número de estaciones sujeto a un tiempo de ciclo máximo con restricciones de zona.

Información relevante del problema

En los sistemas de producción tales como los de industria automotriz, se tienen restricciones de espacio y de tiempo. La longitud física de la estación de trabajo está limitada y cuando la unidad se mueve a través de la estación a una velocidad constante, los operarios deben realizar su trabajo dentro de dicho espacio. En este tiempo, el operario toma las herramientas y materiales a ser ensamblados en la unidad. Por otro lado, cuando el modelo de la unidad cambia, las operaciones de cada estación de trabajo sufren alteraciones en cuanto a las tareas a realizar.

Para registrar la información relevante, la técnica de la ruta crítica (CPM, por sus siglas en inglés) proporciona una forma sencilla cuando hay de por medio una gran cantidad de tareas y recursos además de la interacción entre las actividades⁴. La información mínima necesaria consiste en la descripción de cada actividad, su tiempo estándar y la secuencia de las mismas. Con esta información disponible, se diseña la red correspondiente del contenido de trabajo de una estación. Así, un ejemplo del uso de esta técnica en el caso de un proceso de flujo en línea está dado por Ramírez y Salais⁵.

La red representa un conjunto de restricciones de secuencia, lo cual permite identificar soluciones factibles, entendidas como aquellas que satisfacen este conjunto de restricciones. La figura 1 muestra la red que se analizó, la cual consta de 38 tareas y 6 tareas ficticias con tiempos de cero. Las demás actividades tienen tiempos en milésimas de minuto. Por ejemplo, la tarea 10 que va del nodo 8 al nodo 11 posee un tiempo de 100 milésimas de minuto.

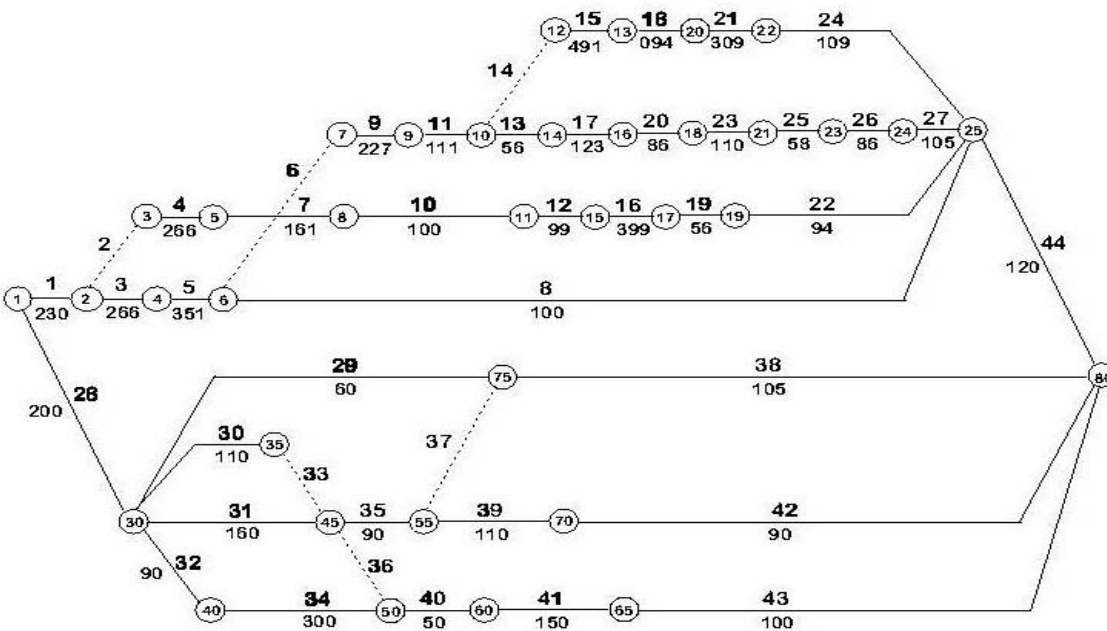


Figura 1. Red de tareas

Un aspecto más es el referente a las habilidades de los operarios. El desempeño de los operarios al realizar cada tarea es estimada y se muestra parcialmente en el cuadro 1.

operario	tarea												
	1	2	3	4	5	6	...	39	40	41	42	43	44
A	110	-	86	84	88	-	...	95	115	118	76	111	117
B	111	-	107	112	92	-	...	119	90	87	96	117	84
C	100	-	91	94	119	-	...	85	79	121	87	85	77
D	105	-	79	106	97	-	...	120	108	123	83	112	93
E	97	-	103	93	114	-	...	77	93	109	121	98	81
F	99	-	93	91	113	-	...	115	93	90	93	110	83

Cuadro 1. Rendimientos de los operarios (%)

Otro aspecto importante considerado se refiere a los tiempos de recorrido de una tarea a otra. En el cuadro 2 se muestran parcialmente los datos por cuestiones de espacio. Se asume que los tiempos de la tarea x a y son los mismos que de y a x .

de \ a	1	2	3	4	5	6	...	39	40	41	42	43	44
1	-	219	72	149	124	204	...	132	66	162	196	29	164
2		-	0	0	0	0	...	0	0	0	0	0	0
3			-	225	80	76	...	110	87	204	28	182	224
4				-	227	186	...	149	28	28	63	145	36
5					-	100	...	228	115	94	89	99	105
6						-	...	0	0	0	0	0	0
-							-						
-							-						
-							-						
39								-	50	27	100	168	207
40									-	146	57	89	72
41										-	26	106	158
42											-	42	182
43												-	223
44													-

Cuadro 2. Tiempos de recorrido de una tarea a otra (milésimas de minuto)

El Algoritmo Genético Simple

El algoritmo genético simple⁶ es un paradigma computacional perteneciente a la computación evolutiva que emula los procesos propuestos en la teoría de la selección natural de Darwin⁷. Los algoritmos genéticos han sido utilizados para resolver problemas de optimización en el cual se generan conjuntos o poblaciones de soluciones que iterativamente se les aplica los procesos de evaluación, selección y reproducción.

El algoritmo genético simple contempla mecanismos sencillos para realizar la selección y la reproducción. Sin embargo, lo más importante del cualquier algoritmo evolutivo es la representación de las posibles soluciones al problema y su correspondiente evaluación.

El diseño de la red es el primer paso y para ello se formula una red inicial la cual se revisa y modifica tantas veces como sea necesario a fin de que represente correctamente las tareas que se deben realizar en la estación de trabajo. Una vez que se obtiene la red definitiva, la cual tendrá m elementos de trabajo, el siguiente paso es representar cada solución factible por medio de una cadena o cromosoma. La cadena es el conjunto de elementos de trabajo asignados a un cierto operario y cada uno de ellos es un gen del cromosoma; cada operario tiene asignado un conjunto de elementos de trabajo a realizar.

Para generar una solución (o cromosoma) factible se fija la velocidad mínima deseada (v_m), por ejemplo 20 partes por hora (p/h), que sería la velocidad actual de la línea de ensamble. Dicha velocidad implica un tiempo de ciclo máximo (t_{cm}) de 3 minutos por parte (m/p), es decir, 60/20.

Para obtener una solución inicial factible se aplica el siguiente algoritmo:

1. $k = 0$ (Número de operario)
2. $k = k + 1$
3. Se genera un número aleatorio entero r_i tal que $1 \leq r_i \leq m$ (elemento i -ésimo de trabajo)
4. Si se satisfacen las restricciones de secuencia, continúa en el paso 5. De otra forma se continúa en el paso 3.
5. Si no se excede el t_{cm} se asigna el r_i elemento de trabajo al operario k y se continúa en el paso 6. De otra forma, se continúa en el paso 2.
6. Si ya se asignaron m elementos de trabajo, se detiene el algoritmo. De otra forma se continúa en el paso 3.

Una vez que se genera la cantidad de soluciones deseada, se dispone de una población inicial compuesta de p individuos o soluciones factibles.

Considerando que el valor de aptitud es el tiempo de ciclo resultante de la estación de trabajo el cual está determinado por la o las rutas críticas y cuyo valor está afectado por los operarios que participan en las tareas de la o las rutas críticas, la población se ordena del menor al mayor tiempo de ciclo, siendo identificada, por lo tanto, la mejor solución alcanzada hasta el momento.

El siguiente paso consiste en la reproducción para lo cual se fija una tasa t_R la cual se recomienda entre 0.5 y 0.8⁸. Si la fijamos en 0.7 implica que el 70% de la población va a cambiar de un valor de aptitud a otro. Para llevar a cabo la reproducción, se consideran dos métodos de selección a partir de la población ordenada:

- a. Selección de las parejas (1, 2), (3, 4), (5, 6), ...
- b. Selección aleatoria de las parejas

Después de seleccionar una pareja (s_i, s_j), se procede a su reproducción generando un número entero aleatorio R_k tal que $1 \leq R_k \leq m$. R_k representa un elemento de trabajo a partir del cual se lleva a cabo el intercambio entre la solución s_i y s_j para obtener dos nuevas soluciones.

Es importante verificar si son factibles las nuevas soluciones h_1 y h_2 , de lo contrario, se desechan una o ambas y se continúa la reproducción con otra pareja hasta completar la tasa t_R . Una vez que se ha completado dicha tasa, se procede a la inserción para lo cual se establece una tasa de reemplazo t_r cuyo valor puede ser hasta el 100%⁹. Si $t_r = 0.5$ implica que la segunda mitad de la población actual ordenada (la peor) se reemplazará por la primer mitad de la nueva generación ordenada (la mejor).

La proporción de la nueva población a utilizar para reemplazar a la anterior está compuesta: (a) exclusivamente por nuevas soluciones o (b) por nuevas soluciones, soluciones de la población actual que no se reprodujeron y soluciones producto de una mutación⁹.

Para la mutación se fija una tasa t_m cuyo valor se recomienda⁸ alrededor de 0.2. Si $t_m = 0.2$ implica que el 20% de la población actual se mutará. Una mutación se obtiene de acuerdo con el siguiente algoritmo:

1. Se genera un número aleatorio entero r_i tal que $0 < r_i < 1$
2. Se calcula la probabilidad de mutación de acuerdo con la ecuación 1.

$$P_m = \frac{1}{2} \left(\frac{1}{p} + \frac{1}{m} \right) \quad (1)$$

3. Si $r_i < P_m$ se lleva a cabo la mutación, de lo contrario no se efectúa y termina el algoritmo.
4. Si procede la mutación, se genera otro número aleatorio tal que $0 < r_j < p$ para seleccionar la solución a mutar. Se genera una nueva solución factible la cual reemplazará a la solución r_j a partir del gen r_k .
5. Se genera otro número aleatorio tal que $0 < r_k < m$ para definir a partir de qué elemento de trabajo (o gen) se mutará.
6. Finalmente se lleva a cabo la mutación y se verifica si es factible. En caso de que no lo sea, se repite el paso 5 hasta que se logre una solución factible.

El proceso descrito en cada iteración va creando nuevas generaciones y se detiene cuando alcanza una meta establecida o satisface una regla de detención.

Experimentación y resultados

Se consideró la red de tareas de una estación de trabajo que actualmente consta de 44 tareas y corre a una velocidad de 19.1 partes por hora (p/h). En la figura 1 se muestra esta red la cual incluye 6 líneas discontinuas que representan tareas ficticias con tiempos de cero que son necesarias para fijar secuencias. El objetivo buscado era determinar el menor número de operarios para dicha estación alcanzando al menos una velocidad de 19.1 p/h. Enseguida se muestran los pasos seguidos y los resultados obtenidos.

Paso 1: Se actualizaron los tiempos estándar de cada elemento de trabajo y se diseñó la red correspondiente, la cual se revisó a fin de que su estructura se apegara al proceso. Una vez que se finalizó la red, se capturó en la interfase. En la figura 2 se muestra parte de la información de esta red donde la información más relevante de cada elemento de trabajo es la siguiente: la columna 3 es un consecutivo, las columnas 4 y 5 son los nodos conectores (i,j), la columna 6 es el tiempo de ejecución en minutos, la columna 13 es el operario asignado inicialmente y la columna 15 es la descripción.

Paso 2: El tiempo de ciclo inicial fue de 3.141 minutos el cual está determinado por la asignación arbitraria inicial de seis operarios, esto es, no se sabe si es factible dicha asignación inicial. Este tiempo de ciclo equivale a una velocidad inicial de 19.1 partes/hora. Además, dicho tiempo de ciclo está compuesto por los elementos de trabajo que constituyen la o las rutas críticas resultantes. Sin embargo, estos datos sirven como punto de partida para llevar a cabo las corridas.

Paso 3: Se efectuaron diez corridas considerando 6 operarios, un tiempo de ciclo máximo (t_{cm}) de 3.141 minutos y 500 iteraciones, además de los siguientes parámetros:

$$m = 44 \quad p = 30 \quad t_R = 0.7 \quad t_m = 0.4 \quad t_r = 0.5$$

La función de aptitud quedó definida como:

Nombre de la red:										No. de operarios:	No. de actividades:	No. de tiempos:		
S3										6	44	1		
										13	15			
Mod. eti.	act.	i	j	Tiempos						unico operario	tipo	descripcion		
00	001	001	0001	0002	0.230	0.000	0.000	0.000	0.000	0.000	0	D	NVA	Toma polipasto y posiciona gancho
					Operarios no permitidos: 0									
00	002	002	0002	0003	0.000	0.000	0.000	0.000	0.000	0.000	0	N/R	NVA	Ficticia
					Operarios no permitidos: 0									
00	003	003	0002	0004	0.266	0.000	0.000	0.000	0.000	0.000	0	D	NVA	Engancha cabina lado derecho
					Operarios no permitidos: 0									
00	004	004	0003	0005	0.266	0.000	0.000	0.000	0.000	0.000	0	C	NVA	Engancha cabina lado izquierdo
					Operarios no permitidos: 0									
00	005	005	0004	0006	0.351	0.000	0.000	0.000	0.000	0.000	0	D	NVA	Traslada cabina a punto de transferencia
					Operarios no permitidos: 0									
00	006	006	0006	0007	0.000	0.000	0.000	0.000	0.000	0.000	0	N/R	NVA	Ficticia
					Operarios no permitidos: 0									
00	007	007	0005	0008	0.161	0.000	0.000	0.000	0.000	0.000	0	C	NVA	Anota y registra unidad
					Operarios no permitidos: 0									
00	008	008	0006	0025	0.100	0.000	0.000	0.000	0.000	0.000	0	D	NVA	Regresa a punto de inicio
					Operarios no permitidos: 0									
00	009	009	0007	0009	0.227	0.000	0.000	0.000	0.000	0.000	0	A	NVA	Toma polipasto y traslada cabina a estación 1
					Operarios no permitidos: 0									
00	010	010	0008	0011	0.100	0.000	0.000	0.000	0.000	0.000	0	C	NVA	Regresa a interior de rack
					Operarios no permitidos: 0									

Figura 2. Registro de la información de la red en la interfase.

$$F = \text{menor (tiempo de ciclo)} \times (\text{número de operarios}) \text{ minutos-operario}$$

Adicionalmente, la regla de detención se activa cuando se acumulan 250 intentos consecutivos y en cada caso la diferencia entre los valores de la función de aptitud de una solución a otra es menor a 0.0001.

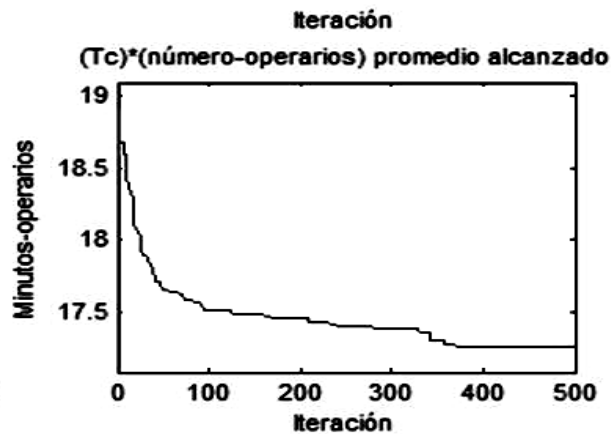


Figura 3. Gráfica del desempeño promedio.

En la figura 3 se muestra el resultado de dichas corridas. Se puede apreciar que el comportamiento es consistente en la reducción de la función de aptitud F además de que se encontraron soluciones factibles con 6 operarios y un tiempo de ciclo menor al t_{cm} .

Paso 4: A raíz del resultado del paso anterior se hizo una sola corrida considerando los mismos parámetros anteriores para evaluar un resultado típico el cual se muestra en el cuadro 3 observando que con 25 iteraciones es suficiente para el resultado deseado.

Operario	Elemento					tiempos/operario			
	k	nodos		Tiempo		reco-rrido	inicia	ter-mina	Total
		i	j	nor-mal	ajus-tado				
A	3*	2	4	0.266	0.333	0.000	0.235	0.567	
A	16	15	17	0.399	0.438	0.160	1.384	1.982	
A	19	17	19	0.056	0.070	0.102	1.982	2.154	
A	38	75	80	0.105	0.124	0.035	0.671	0.830	
A	44*	25	80	0.120	0.110	0.056	2.792	2.959	1.428
B	13*	10	14	0.056	0.066	0.034	1.446	1.546	
B	20*	16	18	0.086	0.097	0.051	1.821	1.969	
B	25*	21	23	0.058	0.098	0.183	2.143	2.424	
B	29	30	75	0.060	0.066	0.000	0.155	0.221	
B	34	40	50	0.300	0.494	0.054	0.250	0.798	
B	42	70	80	0.090	0.108	0.031	0.975	1.114	
B	43	65	80	0.100	0.078	0.042	1.211	1.330	1.400
C	9*	7	9	0.227	0.195	0.037	1.014	1.247	
C	11*	9	10	0.111	0.124	0.076	1.247	1.446	
C	22	19	25	0.094	0.078	0.218	2.154	2.450	
C	24	22	25	0.109	0.082	0.123	2.551	2.756	
C	28	1	30	0.200	0.155	0.000	0.000	0.155	
C	30	30	35	0.110	0.164	0.168	0.155	0.486	
C	35	45	55	0.090	0.097	0.088	0.486	0.671	
C	39	55	70	0.110	0.152	0.152	0.671	0.975	1.909
D	5*	4	6	0.351	0.373	0.074	0.567	1.014	
D	8	6	25	0.100	0.132	0.172	1.014	1.318	
D	17*	14	16	0.123	0.106	0.169	1.546	1.821	
D	18	13	20	0.112	0.108	0.199	1.903	2.209	
D	26*	23	24	0.086	0.114	0.093	2.424	2.631	
D	27*	24	25	0.105	0.090	0.072	2.631	2.792	
D	31	30	45	0.160	0.106	0.000	0.155	0.261	1.807
E	7	5	8	0.161	0.182	0.093	0.698	0.973	
E	10	8	11	0.100	0.066	0.138	0.973	1.177	
E	12	11	15	0.099	0.114	0.092	1.177	1.384	
E	21	20	22	0.309	0.251	0.091	2.209	2.551	
E	23*	18	21	0.110	0.133	0.042	1.969	2.143	
E	32	30	40	0.090	0.096	0.000	0.155	0.250	1.298
F	1*	1	2	0.230	0.235	0.000	0.000	0.235	
F	4	3	5	0.266	0.314	0.149	0.235	0.698	
F	15	12	13	0.491	0.338	0.118	1.446	1.903	
F	40	50	60	0.050	0.047	0.028	0.798	0.873	
F	41	60	65	0.150	0.192	0.146	0.873	1.211	1.567

Cuadro 3. Resultado típico de una corrida

En la columna 2 del cuadro 3 los elementos que conforman la ruta crítica son señalados con un asterisco. Cabe resaltar lo siguiente:

- El tiempo de ciclo de esta corrida es de 2.959 minutos, es decir, un 5.82% inferior al t_{cm} que es de 3.141 minutos.
- Puede verificarse que la solución es factible dado que el tiempo total de cada operario no excede el t_{mc} .
- Los tiempos de inicio y terminación de cada uno de los elementos de trabajo asignados a cada operario no se traslapan entre sí.
- Se puede verificar que está considerado la eficiencia de cada operario (columnas de "tiempo normal" y "tiempo ajustado").

- Se puede verificar que los tiempos de recorrido de un elemento de trabajo a otro también están considerados (columna de "recorrido").
- La figura 4 muestra la evolución del AG de dicha corrida en la que se aprecia claramente la convergencia deseada.

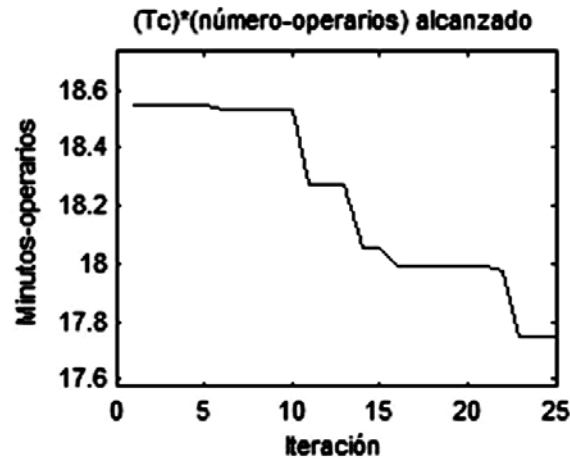


Figura 4. Evolución de una corrida típica.

Finalmente, esta corrida consumió 11.05 minutos en una computadora Pentium® D CPU 3.00GHz Intel®, con 0.99 GB de RAM.

Paso 5: Se evaluaron otras opciones tales como utilizar cinco operarios. El AG no encontró soluciones factibles en tal caso. Así mismo, utilizar más de seis operarios reduce aún más el tiempo de ciclo como era de esperarse, pero el tiempo total por operario es menor. Es decir, el costo de reducir el tiempo de ciclo es un mayor número de operarios con menor carga de trabajo.

Conclusiones

Los resultados obtenidos revelan la conveniencia de utilizar un AGS en una problemática compleja como la descrita en este artículo. Cuanto mayor sea el número de elementos de trabajo a considerar, mayor será el beneficio al utilizar este algoritmo.

Por otro lado, a partir de los elementos de trabajo que forman parte de la o las rutas críticas, se sabe qué elementos impactan el tiempo de ciclo incurrido. Es decir, si tales elementos se ejecutan en mayor o menor tiempo, el tiempo de ciclo de la estación se verá afectado. Esta información permite también identificar factores que potencialmente puedan afectar la ejecución de tales elementos, ayudando a tomar medidas preventivas para evitar alguna contingencia indeseable o al contrario, reducir los tiempos actuales. Tales factores pueden estar relacionados con la distribución física de la estación de trabajo, con los equipos y herramientas que se requieren y naturalmente con la mano de obra asignada entre otros.

Finalmente, al aplicar el AG a las estaciones de trabajo más lentas, se logrará reducir el tiempo de ciclo de toda la línea de ensamble.

Trabajo futuro

Es necesario realizar una comparación exhaustiva para determinar la eficiencia de este algoritmo genético, para lo cual debe de indagarse en la literatura otras heurísticas que puedan aplicarse al escenario planteado. Aunque es importante hacer notar que en la

literatura abundan heurísticas en relación a cierto tipo de problemas no es fácil encontrar una heurística que se adapte al escenario específico aquí considerado. Sin embargo, sí se puede mejorar el algoritmo genético tomando en cuenta otros enfoques tales como: una función de aptitud multi-objetivo, una población inicial pequeña que puede aumentar de una generación a otra, considerar soluciones no factibles y diseñar una representación del cromosoma más eficiente para reducir la epistasis en la codificación.

Además es posible incorporar otra clase de restricciones de zona como es el hecho de que algún elemento de trabajo deba ser realizado solo por uno o varios operarios en particular excluyendo a otros.

Referencias

- ¹ Agpak, K. and Gokcen, H., "A chance-constrained approach to stochastic line balancing problem," *European Journal of Operational Research*, Vol. xx No. xx, 2006, article in press.
- ² Bautista, J. and Pereira, J., "Ant algorithms for a time space constrained assembly line balancing problem", *European Journal of Operational Research*, Vol. Xx No. xx, 2006, article in press.
- ³ Ramírez C., S., Torres Trevino L., Cedillo C., G. and Villegas L. Jorge. "A Genetic Algorithm for the Reassignment of Work in an Assembly Line: A Real Scenario," *Research in Computer Science*, Vol. 23 No.1, 2006.
- ⁴ Mathur, K. and Solow, D., 1996. "Investigación de Operaciones." *Prentice-Hall Hispanoamericana*, S. A. primera edición: 492-541.
- ⁵ Ramírez-Campos, S. and Saláis-Fierro, T., 2004. "Flexibility and Time Optimization in an Automotive Assembly Line: A Neural Network Approach." In proceedings of *Business & Industry Symposium: Advanced Simulation Technology Conference*, Arlington, VA, USA.
- ⁶ Goldberg, D.E., 1989. "Genetic algorithms in search, optimization, and machine learning", *Addison-Wesley Pub Co*".
- ⁷ Darwin, C., "El origen de las especies", 1997. *UNAM*.
- ⁸ Jahangirian, M. and Conroy, G. V., 2000. "Intelligent dynamic scheduling system." *Integrated Manufacturing Systems*, Vol. 11 No. 4: 247-257.
- ⁹ Badibaru, A. B. and Cheung, J. Y., 2002. "Fuzzy Engineering Expert Systems with Neural Network Applications." *John Wiley & Sons, Inc.*