

UN ALGORITMO HÍBRIDO BASADO EN COLONIAS DE HORMIGAS PARA LA RESOLUCIÓN DE PROBLEMAS DE DISTRIBUCIÓN EN PLANTA ORIENTADOS A PROCESOS

Ángel Cobo Ortega, Ana María Serrano Bedia
Universidad de Cantabria

RESUMEN

En este trabajo se presenta un algoritmo híbrido para resolver problemas de distribución en planta, considerando la posibilidad de incluir restricciones de tipo espacial o prioridades de cercanía. El algoritmo integra dos técnicas metaheurísticas: por un lado los algoritmos basados en colonias de hormigas (ACO) y por otro lado un algoritmo genético (AG) que permite mejorar el conjunto de soluciones obtenidas por las hormigas artificiales. En el ACO, un conjunto de hormigas artificiales construyen, de forma concurrente, un conjunto de posibles soluciones del problema por medio de asignaciones parciales de secciones o talleres a áreas de la planta. En ese proceso de asignación utilizan información heurística obtenida a partir de los datos del problema (distancias entre áreas, flujos entre secciones, capacidades de las áreas,...) y también la información que las hormigas precedentes aportan sobre el grado de eficiencia de las soluciones previamente obtenidas. En el trabajo se muestra la eficacia del algoritmo a partir de una implementación usando el software *Mathematica*.

1. DISTRIBUCIÓN EN PLANTA ORIENTADA A PROCESOS

La distribución en planta, vista como el proceso de determinar la organización física de un sistema productivo, es una decisión de carácter estratégico en las empresas. Una buena distribución de la planta productiva permite alcanzar ventajas competitivas en cuanto a costes y mejorar la eficiencia del sistema productivo. En el caso de la distribución en planta orientada a procesos, el objetivo es determinar la forma más efectiva de localizar un conjunto de secciones o talleres en una serie de áreas en la planta. En este caso la eficiencia se mide a partir del coste que supone el traslado de materiales o personas entre las diferentes secciones. Este coste puede expresarse como

$$Coste = \sum_{i=1}^n \sum_{j=1}^n t_{ij} d_{ij} c_{ij} \quad (1)$$

donde n es el número de secciones; t_{ij} representa el número de cargas o personas a trasladar, durante un periodo de tiempo, desde la sección i a la sección j ; d_{ij} es la distancia entre dichas secciones; y c_{ij} representa el coste unitario del traslado que, sin pérdida de generalidad, se puede suponer constante. El objetivo del problema es minimizar la función (1) sujeta a diferentes restricciones de carácter espacial. La existencia de restricciones, unido al extraordinariamente elevado número de posibles combinaciones, hace que este tipo de problemas sean muchas veces inabordables mediante algoritmos óptimos; en tales caso se suele recurrir a técnicas heurísticas que permitan obtener "buenas" soluciones en tiempos de respuesta razonables.

2. METAHEURÍSTICA

En las últimas décadas, muchos problemas de optimización combinatoria han sido abordados mediante un nuevo tipo de algoritmos llamados *metaheurísticos*. Aunque no es fácil encontrar una definición comúnmente aceptada (Blum and Roli, 2003), se podría decir que una *metaheurística es un proceso iterativo de generación que guía a una heurística subordinada combinando de forma inteligente diferentes conceptos para la exploración del espacio de búsqueda y usando estrategias de aprendizaje para estructurar la información con objeto de encontrar eficientemente soluciones cercanas al óptimo* (Osman and Laporte, 1996).

De acuerdo a una clasificación de Blum y Roli (2003), dentro de los algoritmos metaheurísticos se distinguen los denominados métodos de trayectoria (búsqueda local,

templado simulado, búsqueda tabú,...) y los métodos basados en poblaciones. En esta última categoría se enmarcan las dos técnicas utilizadas en este trabajo: los algoritmos genéticos y los basados en colonias de hormigas.

Ejemplos de aplicación de diferentes metaheurísticas a problemas de distribución pueden ser encontrados en Tate y Smith (1995), Meller y Bozer (1996), Tavakkoli-Moghaddain y Shayan (1998), Kochhar, Foster y Heragu (1998), Kim y Kim (1998) o Solimanpur, Vrat y Shankar (2004). Diferentes estudios comparativos vienen a demostrar la importancia de desarrollar algoritmos híbridos que incorporen las ventajas de diferentes técnicas, como puede ser, por ejemplo, el uso de operadores genéticos para mejorar las soluciones obtenidas por las colonias de hormigas. Ejemplos de este nuevo enfoque son los trabajos de Lee y Lee (2002) y de McKendall y Shang (2004).

2.1. Algoritmos genéticos

Los algoritmos genéticos (Goldberg, 1989) son procedimientos de búsqueda basados en los principios de la evolución y la selección natural, y que pueden ser usados para resolver problemas de optimización en espacios de soluciones factibles que puedan ser codificadas mediante cadenas finitas en un alfabeto finito. Este tipo de algoritmos trabajan con poblaciones de potenciales soluciones, generando en cada iteración nuevas poblaciones mediante los conocidos como operadores genéticos. Estos operadores son tres: un operador de selección de las mejores soluciones, un operador de cruce que genera nuevas soluciones a partir de dos soluciones previas, y un operador de mutación que realiza una pequeña transformación sobre una solución previa. A pesar de su simplicidad, los algoritmos genéticos han demostrado ser una herramienta muy útil en diferentes problemas de optimización. Un ejemplo de aplicación a problemas de distribución en planta puede encontrarse en (Cobo y Serrano, 2001).

2.2. Optimización basada en colonias de hormigas (ACO)

En los últimos años se ha venido desarrollando una nueva técnica metaheurística que se inspira en el comportamiento real de las hormigas. Los algoritmos de optimización con colonias de hormigas (Ant Colony Optimization, ACO) han sido aplicados con éxito a una amplia variedad de problemas de optimización combinatoria; el primero de ellos fue el conocido *problema del viajante* (Dorigo, 1992). ACO es una técnica en la que un conjunto de hormigas artificiales construyen de forma incremental,

seleccionado paso a paso diferentes componentes, un conjunto de soluciones del problema. En este proceso, las hormigas hacen uso de información a priori sobre el problema (información heurística) e información proporcionada por las hormigas precedentes sobre la calidad de las soluciones por ellas construidas (feromona).

3. IMPLEMENTACIÓN DEL ALGORITMO

3.1. Función objetivo

El primer problema que se plantea a la hora de diseñar el algoritmo ACO para el problema de distribución en planta es el tratamiento de las restricciones. El algoritmo no garantiza la factibilidad de todas las soluciones obtenidas, pero haciendo uso de una estrategia de penalización se podría conseguir reducir notablemente la presencia de soluciones infactibles en la población final. De esta forma el coste total de una posible solución del problema sería:

$$C(P) = \sum_{i=1}^n \sum_{j=1}^n t_{ij} d_{ij} + p_c VC(P) + p_p VP(P) \quad (2)$$

donde $VC(P)$ y $VP(P)$ representan el número de secciones asignadas a áreas de menor capacidad de la prevista y una medida de las violaciones de proximidad producidas.

3.2. Representación de las soluciones

Cada posible solución del problema puede entenderse como una permutación del número de secciones a ubicar, y como tal, representada como un vector de n componentes enteras, donde $P(i)=j$ significa que la sección S_j será situada en el área A_i .

3.3. Información heurística y rastros de feromona

Cada una de las hormigas artificiales del algoritmo construirá una posible solución del problema realizando asignaciones parciales de secciones a áreas de la planta. La información que usará para decidir si debe realizar o no una asignación parcial son los datos sobre distancias entre áreas y flujos entre secciones (información heurística) e información sobre la calidad de las soluciones previamente obtenidas en las que se realizó esa asignación (rastro de feromona). Esta información será almacenada en sendas matrices cuadradas de dimensión n .

La matriz con la información heurística del problema es:

$$\eta = (\eta_{ij}), \quad (3)$$

donde η_{ij} puede ser interpretado como una medida de la calidad de la asignación parcial de la sección S_i al área A_j . Una buena estrategia para situar las secciones es colocar en las áreas centrales de la planta aquellas secciones que tengan un alto grado de interacción con otras. Como medida de la centralidad de las áreas puede usarse la desviación típica del vector de distancias:

$$C(A_j) = r - SD(d_{1j}, d_{2j}, \dots, d_{nj}), \quad (4)$$

En cuanto a la interacción de las secciones, una sección tendrá una interacción alta cuando tenga flujos de mercancías con un gran número de secciones, o cuando a pesar de tener interacciones con pocas secciones, los flujos correspondientes sean altos. El grado de interacción de las secciones se definirá con la siguiente expresión:

$$I(S_i) = \frac{\lambda_1}{M} \frac{\sum_{k=1}^n (t_{ik} + t_{ki})}{n} + \frac{\lambda_2}{N} \text{Card}\{S_k \mid t_{ik} \neq 0 \text{ or } t_{ki} \neq 0\} \quad (5)$$

donde M es el flujo medio máximo de las secciones y N el número máximo de interacciones que una sección tiene con otras. Si se toman λ_1 y λ_2 dos números no negativos con $\lambda_1 + \lambda_2 = 1$, el nivel de interacción será una combinación lineal convexa de ambas medidas de interacción. Con objeto de obtener valores comparables, ambas medidas serán normalizadas. Usando (4) y (5) la matriz (3) es definida por:

$$\eta_{ij} = \frac{1}{\left| 1 - \frac{\bar{I}(S_i)}{C(A_j)} \right|} \quad i, j = 1, \dots, n \quad (6)$$

De acuerdo a esta expresión, cuando los niveles de interacción de la sección y de centralidad del área son similares, el valor del correspondiente elemento de la matriz será alto y habrá más posibilidades de asignar la sección i -ésima al área j -ésimo.

En el proceso de construcción, las hormigas artificiales usarán otra información: los rastros de feromona dejados por las hormigas precedentes. Esto será almacenado en una nueva matriz $\tau = (\tau_{ij})$. Para inicializar esa matriz se generará una solución aleatoria y a partir de ella se asignan los valores:

$$\tau_{ij} = \tau_0 = \frac{m}{C(P_0)} \quad i, j = 1, \dots, n \quad (7)$$

donde m es el número de hormigas que trabajarán concurrentemente en cada iteración, P_0 es la solución generada aleatoriamente y $C(P_0)$ su coste asociado.

3.4. Construcción de una solución por una hormiga

En el algoritmo m hormigas artificiales construirán sendas soluciones realizando asignaciones parciales de secciones a áreas. Para comenzar, cada hormiga elige de forma aleatoria el orden en el que las n secciones serán asignadas. A continuación se usa una regla probabilística para decidir el área a asignar a cada sección. La probabilidad con la que la hormiga k decidirá localizar la sección S_i en el área A_j es:

$$p_{ij}^k = \begin{cases} \frac{(\tau_{ij})^\alpha (\eta_{ij})^\beta}{\sum_{q \in N_k} (\tau_{iq})^\alpha (\eta_{iq})^\beta} & \text{if } j \in N_k \\ 0 & \text{if } j \notin N_k \end{cases} \quad (8)$$

donde N_k es el conjunto de índices de áreas que la hormiga k aún no ha utilizado, y α y β son dos parámetros que determinan la influencia relativa del rastro de feromona y la información heurística, respectivamente. De acuerdo a (8) la probabilidad de asignar S_i a A_j crece con el valor del rastro de feromona asociado τ_{ij} y el valor de la información heurística η_{ij} . La elección de los parámetros α y β debe realizarse con cuidado, ya que para valores de α superiores a 1 puede ocurrir que todas las hormigas construyan la misma solución, no explorando adecuadamente el espacio de soluciones.

3.5. Uso de operadores genéticos

Una vez que las m hormigas artificiales han construido sus respectivas soluciones se utilizarán los tres operadores genéticos básicos (selección, reproducción y mutación) para tratar de mejorar la calidad de la población de soluciones. Mediante el operador de selección la probabilidad de que una solución se considere es inversamente proporcional al coste total asociado. Después de la selección algunas de las posibles soluciones de la población se combinarán para construir soluciones diferentes. Para la combinación se utilizará el conocido operador PMX (Partial Matched Crossover), propuesto por Goldberg y Lingle. Este operador construye dos soluciones seleccionando de forma aleatoria dos puntos de cruce; estos puntos definen una sección de intercambio que permite realizar el cruce mediante sencillos cambios. Por ejemplo, si se tienen dos permutaciones $P_1=(6,3,2,7,8,1,5,10,9,4)$ y $P_2=(1,6,9,3,4,5,2,7,10,8)$ y los dos puntos de cruce son $n_1=5$ y $n_2=8$, en primer lugar deberían intercambiarse los segmentos entre esos dos puntos. Este intercambio define también unas transformaciones ($8 \leftrightarrow 4$, $1 \leftrightarrow 5$,

$5 \leftrightarrow 2$, y $10 \leftrightarrow 7$) que deberían ser aplicadas sobre P_1 y P_2 . Como resultado se obtendrían dos nuevas soluciones: $Q_1=(6,3,5,10,4,5,2,7,9,8)$ y $Q_2=(2,6,9,3,8,1,5,10,7,4)$.

En todo algoritmo genético el operador de cruce debe ser aplicado con una probabilidad prefijada; una buena efectividad del algoritmo suele exigir la elección de altas probabilidades de cruce.

Tras la aplicación del operador de cruce, algunas de las soluciones sufrirán pequeñas transformaciones (mutaciones) que las convertirán en soluciones diferentes. El operador de mutación que se utilizará en el algoritmo consistirá en el intercambio de los elementos de dos posiciones elegidas al azar sobre el vector. En este caso, una buena efectividad del algoritmo suele exigir bajas probabilidades de mutación.

El principal problema que tiene el uso de los operadores genéticos es que con frecuencia rompen la factibilidad de las soluciones y por tanto se exige de nuevo penalizar las soluciones obtenidas. De esta forma, tras cada iteración del algoritmo se elegirán las m mejores soluciones obtenidas por las hormigas y por las transformaciones genéticas sobre éstas. Con esta nueva población de soluciones se pasará a la actualización de los rastros de feromona.

3.6. Actualización de la feromona

Tras cada iteración, los valores de la matriz de feromona deben ser actualizados para que aporten información nueva a las hormigas de la siguiente iteración. Lo primero que debe hacerse es la reducción del rastro de feromona en todos los elementos por un factor constante ρ , conocido como tasa de evaporación:

$$\tau_{ij} = (1 - \rho)\tau_{ij} \quad i, j = 1, \dots, n \quad (9)$$

Esta evaporación permite al algoritmo "olvidar" aquellas decisiones previamente tomadas y que condujeron a soluciones con altos costes asociados. Si las hormigas no hacen uso de una asignación particular, el correspondiente valor en la matriz decrecerá de forma exponencial con el número de iteraciones.

Después de la evaporación, todas las hormigas incrementarán el valor asociado a las asignaciones que han realizado de forma proporcional a la calidad de la solución obtenida:

$$\tau_{ij} = \begin{cases} \tau_{ij} + \frac{1}{C(P^k)} & \text{if } P^k(j) = i \\ \tau_{ij} & \text{if } P^k(j) \neq i \end{cases} \quad i, j = 1, \dots, n; \quad k = 1, \dots, m \quad (10)$$

La conocida como *estrategia elitista*, propuesta por Dorigo (1992), puede ser usada para mejorar el algoritmo; la idea es que la mejor solución encontrada desde el inicio del algoritmo realice una aportación extra de feromona:

$$\tau_{ij} = \begin{cases} \tau_{ij} + \frac{e}{C(P^{best})} & \text{if } P^{best}(j) = i \\ \tau_{ij} & \text{if } P^{best}(j) \neq i \end{cases} \quad i, j = 1, \dots, n; \quad (11)$$

donde e es un nuevo parámetro que define el peso que se dará a esa mejor solución.

4. EJEMPLO ILUSTRATIVO

Para mostrar el funcionamiento del algoritmo se considerará un ejemplo de asignación de 18 secciones a 18 áreas. Los flujos entre las diferentes secciones y las distancias entre las áreas se muestran en las Tablas 1 y 2, respectivamente.

Considérese igualmente que cada área tiene diferente tamaño y que cada sección requiere un espacio mínimo. Los tamaños de las áreas serán 4, 4, 2, 1, 1, 2, 3, 3, 3, 4, 2, 3, 2, 4, 2, 4, 2 y 2. Por otro lado se considerará que las 7 primeras secciones (S_1 a S_7) requieren un espacio mínimo de 1 unidad, las secciones S_8 a S_{13} un espacio de al menos 2 unidades, las secciones S_{14} a S_{16} , 3 unidades, y, finalmente, las dos últimas secciones (S_{17} y S_{18}) un espacio mínimo de 4 unidades.

	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9	S_{10}	S_{11}	S_{12}	S_{13}	S_{14}	S_{15}	S_{16}	S_{17}	S_{18}
S_1	0	45	32	30	0	0	0	12	23	31	11	0	40	0	12	15	31	0
S_2	12	0	27	0	45	0	0	0	0	0	0	0	0	0	0	0	0	0
S_3	14	0	0	0	0	24	0	0	0	0	0	0	0	11	7	34	23	0
S_4	22	12	0	0	32	0	0	0	0	60	45	0	0	0	0	0	0	0
S_5	0	0	0	0	0	43	32	16	48	0	0	0	0	0	0	0	21	12
S_6	0	12	0	0	0	0	0	8	4	8	11	0	20	7	9	3	8	12
S_7	0	0	0	0	0	0	0	0	0	0	0	0	21	80	97	56	76	0
S_8	0	0	0	45	23	12	0	0	0	0	0	0	0	0	0	0	0	0
S_9	0	0	0	0	0	11	32	14	0	0	0	0	0	0	0	0	0	0
S_{10}	0	25	0	0	0	0	0	0	0	0	0	33	0	0	0	0	0	0
S_{11}	0	0	0	0	0	0	0	0	0	0	0	45	0	0	0	0	5	23
S_{12}	0	0	0	0	0	0	14	28	38	0	0	0	0	0	0	0	0	0
S_{13}	0	0	0	0	43	24	0	0	0	0	0	0	0	0	0	0	0	0
S_{14}	11	0	15	23	0	0	0	0	0	0	0	0	0	0	0	0	0	0
S_{15}	0	0	0	0	0	0	23	11	12	0	0	0	0	0	0	0	0	0
S_{16}	0	0	0	0	0	0	0	0	0	0	0	56	24	0	18	0	45	39
S_{17}	0	0	0	0	0	0	0	0	0	0	0	17	0	22	0	45	0	11
S_{18}	0	0	0	0	12	15	18	0	0	0	0	0	36	49	67	23	11	0

Tabla 1: Flujos de mercancías entre las secciones.

Otro tipo de restricción que será considerada es la de exigir, por razones de seguridad, una distancia de al menos 4 unidades de separación entre las secciones segunda y octava. La penalización que se aplicará a una solución por no cumplir esta

restricción será proporcional a la distancia necesaria para alcanzar esa distancia de seguridad.

A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9	A_{10}	A_{11}	A_{12}	A_{13}	A_{14}	A_{15}	A_{16}	A_{17}	A_{18}	Areas
0	1	3	5	6	3	5	6	1	3	5	2	3	3	5	6	8	9	A_1
	0	1	3	4	1	3	4	1	1	3	2	5	3	4	5	7	8	A_2
		0	1	2	1	2	3	4	3	2	5	8	6	5	4	5	6	A_3
			0	1	2	1	2	6	4	3	7	10	8	7	5	4	5	A_4
				0	3	2	1	7	5	4	8	11	9	8	6	5	4	A_5
					0	1	2	3	1	1	4	7	5	4	3	4	5	A_6
						0	1	4	2	1	4	7	5	4	2	1	2	A_7
							0	5	3	2	5	8	6	5	3	2	1	A_8
								0	1	3	1	2	2	3	4	6	7	A_9
									0	1	1	4	2	1	1	3	4	A_{10}
										0	3	6	4	3	1	2	3	A_{11}
											0	1	1	2	3	5	6	A_{12}
												0	1	3	4	6	7	A_{13}
													0	1	2	4	5	A_{14}
														0	1	3	4	A_{15}
															0	1	2	A_{16}
																0	1	A_{17}
																	0	A_{18}

Tabla 2: Distancias entre las áreas (se asume la simetría en las distancias).

4.1. Resultados experimentales

Para comprobar la efectividad del algoritmo, éste ha sido implementado en *Mathematica* y probado sobre el ejemplo propuesto. El escenario sobre el que se ha realizado la simulación es el determinado por los siguientes valores de los parámetros:

- Número de hormigas: $m=20$
- Coeficiente de penalización por violaciones de restricciones de capacidad: $p_c=10$
- Coeficientes de la definición del grado de interactividad: $\lambda_1=0.5$ $\lambda_2=0.5$
- Probabilidad de cruce: 0.6
- Tasa de evaporación: $\rho=0.5$
- Número de iteraciones: 40
- Coeficiente de penalización por violaciones de restricciones de proximidad: $p_p=10$
- Pesos de los rastros de feromona y de la información heurística: $\alpha=1$ $\beta=0.15$
- Probabilidad de mutación: 0.01
- Peso de la mejor solución: $e=2$

Después de las 40 iteraciones del algoritmo, la mejor solución encontrada correspondió a la permutación $(18,13,8,5,6,11,1,4,14,16,3,7,9,15,12,17,2,10)$ con un coste asociado de 6113. Además de esa solución, como resultado se obtuvo toda una población de 20 soluciones, de las cuales se tuvieron que descartar 2 por resultar no factibles. La Tabla 3 muestra esas soluciones obtenidas junto con sus costes.

Como prueba de la convergencia del algoritmo, la Figura 1 muestra la evolución del coste medio de las soluciones obtenidas por las hormigas en cada una de las 40 iteraciones, y un gráfico de densidad con las concentraciones finales de feromona.

Solución	Coste total	Solución	Coste total
16,11,8,5,6,1,15,3,14,17,4,7,9,18,12,10,2,13	7764	15,13,8,4,6,3,10,14,11,18,12,7,9,16,5,17,2,1	7486
16,13,8,5,6,11,10,14,4,17,3,7,9,15,12,18,2,1	8298	17,18,8,4,6,11,1,7,5,16,3,14,9,15,12,10,2,13	8323
15,13,8,5,6,11,1,12,14,16,3,7,9,18,4,17,2,10	7149	15,13,8,5,6,11,10,4,14,17,3,7,9,18,12,16,2,1	6893
16,13,8,5,6,11,10,4,14,17,3,7,9,18,12,15,2,1	7412	17,15,8,5,6,11,1,3,14,16,13,4,9,18,10,7,2,12	8669
1,5,8,3,6,11,14,7,9,16,12,15,13,18,4,17,2,10	8315	15,13,8,4,6,1,14,5,11,17,3,7,9,18,12,16,2,10	6871
17,13,8,3,6,11,16,4,14,7,10,12,9,18,5,15,2,1	7757	13,10,8,5,6,11,16,4,14,17,3,7,9,15,12,18,2,1	7738
16,7,8,5,6,13,10,3,14,15,12,11,9,18,4,17,2,1	8019	16,13,8,5,6,11,1,12,14,18,3,7,9,15,4,17,2,10	7472
15,13,8,3,6,11,10,5,14,16,12,7,9,18,4,17,2,1	7368	13,18,8,4,6,10,14,7,11,5,3,16,9,15,12,17,2,1	8124

Tabla 3: Soluciones obtenidas.

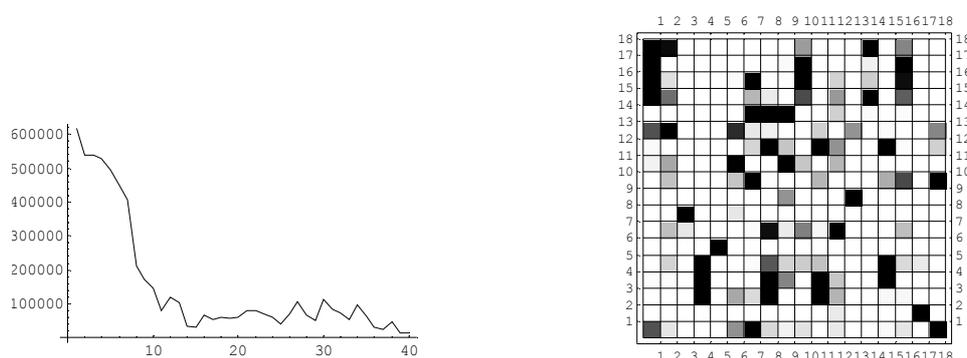


Figura 1: Evolución del coste medio (izquierda) y concentraciones finales de feromona (derecha) tras las 40 iteraciones.

5. CONCLUSIONES

Este trabajo ha pretendido mostrar como la utilización combinada de técnicas metaheurísticas puede resultar interesante en problemas de naturaleza combinatoria, y en particular en los problemas de distribución en planta. La principal diferencia del algoritmo propuesto respecto a otras alternativas es la utilización de reglas de transición probabilísticas, y el trabajar con poblaciones de potenciales soluciones. Esta última característica facilita notablemente la toma de decisión final. Un sencillo ejemplo ha sido utilizado para mostrar la validez del algoritmo, observándose la necesidad de una cuidadosa selección de los parámetros que intervienen. Desde un punto de vista computacional, el algoritmo propuesto resulta muy adecuado para una programación orientada a objeto y con computación paralela.

5. REFERENCIAS BIBLIOGRÁFICAS

- Blum, C. y Roli, A. (2003): Metaheuristics in combinatorial optimization: overview and conceptual comparison, *ACM Computing Surveys*, vol. 35(3), pp.268-308.
- Cobo, A. y Serrano, A. (2001). Algoritmos genéticos para la resolución de problemas de distribución en planta con restricciones espaciales. 5° Congreso CAIP. Campos do Jordao (Brasil).
- Dorigo, M. (1992). Optimization, Learning and Natural Algorithms (in italian). PhD Thesis. Dipartimento di Elettronica, Politecnico di Milano.
- Dorigo, M.; Stützle, T. (2004). *Ant Colony Optimization*. MIT Press.
- Goldberg, D.E. (1989): *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley.
- Kim, J. G. y Kim, Y. D. (1998): A space partitioning method for facility layout problems with shape constraints, *IIE Transactions*, vol. 30, pp. 947-957.
- Lee, Y.H., Lee, M.H. (2002): A shape-based block layout approach to facility layout problems using hybrid genetic algorithm, *Computers and Industrial Engineering*, 42, pp 237-248.
- McKendall, A. R. y Shang, J. (2004): Hybrid ant systems for the dynamic facility layout problem, *Computers and Operations Research*, article in press.
- Meller, R. D. y Gau, K-Y. (1996): The facility layout problem: Recent and emerging trends and perspectives, *Journal of Manufacturing Systems*, vol. 15 (5), pp. 351-366.
- Osman, I.H; Laporte, G. (1996). Metaheuristics: A bibliography. *Ann. Operations Research*. 63, 513–623.
- Solimanpur, M., Vrat, P. y Shankar, R. (2004): Ant colony optimization to the inter-cell layout problem in cellular manufacturing, *European Journal of Operational Research*, 157, pp. 592-606.
- Tate, D. M. y Smith, A. E. (1995): Unequal area facility layout using genetic search, *IIE Transactions*, vol 27 (4), pp. 465-472.
- Tavakkoli-Moghaddain, R. y Shayan, E. (1998): Facilities layout design by genetic algorithms. *Computers Industrial Engineering*, vol. 35 (3-4), pp. 527-530.