

SISTEMA DE PLANEACIÓN AVANZADA Y SIMULACIÓN DE PROCESOS PRODUCTIVOS

Carlos Paredes Ortiz¹, Fabricio Echeverría².

¹Ingeniero Electrónico en Computación 2006

²Director de Tópico. Ingeniero Electrónico de Computación, Escuela Superior Politécnica 1996

RESUMEN

La simulación y optimización aplicada en la secuencia de procesos productivos, es el resultado, que en la actualidad no contamos con métodos diseñados para este tipo de procesos, de los cuales optimizamos la función de utilidad neta, es decir hacerla independiente del comportamiento de las variables aleatorias propias del proceso.

La simulación de procesos, es el resultado de diseñar un modelo de un sistema real, para realizar posteriores experimentos a partir de hipótesis. Por lo anterior resulta más eficiente experimentar sobre modelos basados en sistemas reales que en los mismos sistemas.

Inicialmente detallaremos los aspectos más relevantes sobre la simulación de procesos a partir de métodos estadísticos, como el Método Monte Carlo, enunciaremos las bases para la resolución de nuestro problema, como son la generación de números pseudos aleatorios aplicada a la formación de diferentes distribuciones, siempre considerando que las distribuciones son independientes entre sí.

Analizaremos el por qué? de la optimización combinatorial, posteriormente se hará una reseña de los algoritmos meta heurísticos más conocidos, así como el análisis de convergencia de estos. En base a los resultados que vamos a encontrar y el procedimiento a simular seleccionaremos el algoritmo meta heurístico. Por último, analizaremos los resultados de la simulación, partiendo de los resúmenes estadísticos y principalmente de los gráficos de sus funciones acumuladas (ojivas).

SUMARY

The simulation and applied optimization in the sequence of productive processes, is the result of non designed methods for this type of processes whose functional utility is optimized. This means to make it independent from the behavior of random variables belonging to the process.

The simulation of processes is the result of designing a model of a real system in order to develop late experiments starting on hypothesis. Based on this, it is more efficient to experiment over models based on real systems.

Initially the most relevant aspects about simulation of processes will be detailed starting on statistics methods, like the MONTE CARLO METHOD, it will also be set out the basis for the solution of this problem, like the generation of pseudo random numbers applied to the formulation of different distributions, considering always that the distributions are independent beneath.

It will be analyzed the why of the combinatorial optimization, later it will be done a review of the meta algorithms heuristics more known, as well as the analysis of convergence of these ones. Based on the results that we are going to find and the procedure to simulate we will choose the meta algorithm heuristic. The algorithm Tabu Search helps us in the exploration of a neighborhood through the matrix of memories to short and long term; additionally we will analyze the variable of the general algorithm named "Tabu with outset". Finally, we will analyze the results of the simulation, beginning at the statistics briefings and principally the graphics of accumulated functions (pointed arch).

INTRODUCCIÓN

Conforme avanza la tecnología y el pensamiento científico retoma algoritmos, que inicialmente fueron diseñados para eventos específicos, dado su difícil manejo tecnológico para la época que aparecieron, observamos que éstos (algoritmos meta-heurísticos) tienen cada vez más relevancia en la resolución de eventos de diversa naturaleza esto debido que en la actualidad ha evolucionado en gran medida la tecnología en hardware como en software. Aunque, como se detalla posteriormente los algoritmos meta-heurísticos en algunos casos no proveen de la mejor solución, en la mayoría de ellos nos proporcionan soluciones que son óptimas aunque no necesariamente la más eficiente, por lo anterior es necesario validar la necesidad de obtener una solución en función de su tiempo y esfuerzo computacional.

La simulación de los sistemas tiene su base en la estadística, y de ella obtenemos ¿cuándo y por qué simular un proceso?, es decir nos describe el marco referencial y los principios (escenarios) de la simulación de un proceso, independiente de la naturaleza de este. Uno de los grandes problemas en la simulación de sistema consiste en encontrar un método a través del cual podamos generar variables aleatorias independientes entre sí, como detallaremos posteriormente; adicional a esto, determinar una función probabilística que modele el comportamiento de las variables que interviene en el sistema, a partir de sus datos históricos.

Es necesario recalcar que el presente trabajo resalta el hecho, que por tiempo y costo financiero resulta más conveniente y óptimo de escoger las soluciones que nos entregan estos algoritmos meta-heurístico, que definir complicadas estructura matemáticas las cuales son casi imposibles de interpretar dentro de un proceso de simulación, adicionalmente su implementación resulta altamente costoso en tiempo, dado que en ambos casos nos da como resultado soluciones con escaso margen de diferencia y en el mejor de los casos resulta la misma solución, por lo anterior se recomienda la utilización de este tipo de algoritmo meta-heurísticos en la resolución de problemas con complicada interpretación matemática.

El análisis no solo comprenderá la entrega de la solución óptima por parte del algoritmo, sino adicionalmente confirmaremos dicha solución a través de procedimientos estadísticos, como es la simulación de los procesos a intervenir en el sistema, es clave la idea de definir distribuciones estadísticas que encuadren el comportamiento de las variables de los procesos, así como la independencia de estas entre sí.

CONTENIDO

“Simulación es el proceso de diseñar un modelo de un sistema real y llevar a cabo experiencias con el mismo con la finalidad de comprender el comportamiento del sistema o de evaluar nuevas estrategias – dentro de los límites impuestos por un criterio o conjunto de ellos, para el funcionamiento del sistema.” R. S. Shannon. [54]

1. TIPOS DE SIMULACIÓN

Las técnicas de simulación se dividen en cuatro grupos:

- ✚ Simulación de sistemas continuos.
- ✚ Simulación de procesos por lotes (eventos discretos).
- ✚ Simulación combinada.
- ✚ Simulación de Monte Carlo.

Von Neuman y Ulam publicaron un artículo titulado “The Monte Carlo method” en 1949, los primeros ensayo de simulación que se realizaron en los Estados Unidos bajo el nombre de análisis Monte Carlo fueron en el año de 1940. El método ya era conocido en estadística de ahí su nombre de Simulación Estadística o Método Monte Carlo.

Todo experimento debe generar muestras independientes y plenamente identificables para aplicar el teorema del límite central, se garantiza la independencia aplicando semillas diferentes.

Detalle del algoritmo “Simulación Monte Carlo”

1. Diseñar el modelo lógico de decisión
2. Especificar distribuciones de probabilidad para las variables aleatorias relevantes.
3. Incluir posibles dependencias entre variables.
4. Muestrear valores de las variables aleatorias
5. Calcular el resultado del modelo según los valores del muestreo (iteración) y registrar el resultado

6. Repetir el proceso hasta tener una muestra estadísticamente representativa N veces (independientes entre si)
7. Obtener la distribución de frecuencias del resultado de las iteraciones
8. Calcular media, desvío y curva de percentil acumulado

2. Generación de números pseudos aleatorios.

La generación de números aleatorios tiene un rol importante en los procesos de simulación, para lo cual necesitamos números aleatorios como semillas para tomar las muestras de las variables aleatorias.

Características de un generador de números aleatorios son:

- ✚ Valores de distribución uniforme: Cualquier número dentro del rango deberá tener la misma probabilidad
- ✚ Asegura inexistencia de correlación serial: Un número en la secuencia, no afecta la probabilidad de seleccionar otro número o el mismo.

En la actualidad existen tres tipos de procedimientos:

- ✚ Utilización de tablas
- ✚ Dispositivos especiales

3. Procedimientos que generan números pseudos aleatorios

Son pseudos aleatorios porque se generan mediante funciones, y se usan como semillas para generar valores de variables aleatorias, entre sus ventajas son rápidas y de bajo costo, pero son de periodo finito.

En general, la secuencia de números generados debe contemplar:

- ✚ Distribución uniforme.
- ✚ Algoritmos rápidos.
- ✚ No correlación serial.
- ✚ Secuencias largas, sin saltos.

4. Algoritmos Meta-heurística.

En los problemas combinatorial, existe siempre el procedimiento elemental, para determinar la solución optima requerida. Entre los cuales tenemos el método de explosión exhaustiva (genera todas las soluciones posibles)

“Heurística son procedimientos simples, a menudo basados en el sentido común, que se supone ofrecerán una buena solución (aunque no sea la optima) a problemas difíciles, de un modo fácil y rápido...” [54]

A continuación enunciaremos los diferentes tipos de algoritmos heurísticas:

- ◆ **Métodos Constructivos:** Consiste en ir agregando componentes individuales a la solución hasta que se obtiene una solución viable.
- ◆ **Métodos de Descomposición:** Se trata de dividir el problema, para que la salida de una parte de solución sea el ingreso de la otra parte del problema
- ◆ **Métodos de Reducción:** Tratan de identificar algunas características que presumiblemente deba poseer la solución optima y de ese modo simplificar el problema.
- ◆ **Manipulación del Modelo:** Modifican las estructuras del modelo con el fin de hacerlo mas sencillo de resolver, deduciendo, a partir de su solución, la solución del problema original. Por ejemplo: Linealizar el modelo.
- ◆ **Métodos de Búsqueda de Entorno:** Parten de una solución inicial (obtenida mediante otra heurística) y mediante iteraciones de esta solución obtiene otras soluciones de forma iterativa, hasta que se obtenga el cumplimiento del criterio de parada. Por ejemplo: El método de aproximación de Newton

Entre los algoritmos meta-heurísticos más estudiados tenemos:

-
- ◆ Redes Neuronales
 - ◆ *Grasp*
 - ◆ *Ant Colony*
 - ◆ *Comportamiento Emergente*
 - ◆ *Scatter Search Estático (ES)*
 - ◆ *Scatter Search Dinámico (DI)*
 - ◆ Variante de *Scatter Search Dinámico (VD)*

- ♦ *Algoritmo Memético*.(ME)
- ♦ *Algoritmo de propagación* hacia atrás usado en *Neural Works Profession II/Plus*. Sexton y otros, (1998 BP)
- ♦ *Algoritmo Búsqueda Tabú* propuesto en Sexton y otros (1999 TS)
- ♦ *Algoritmo Temple Simulado* propuesto en Sexton y otros (1999 SA)
- ♦ *Algoritmo Genético* propuesto en Sexton y otros (1999 GA)
- ♦ Adaptación de *Scatter Search* propuesta por Laguna y Martí (2002 SS)

5. Selección de algoritmo meta-heurístico.

Entre las distintas técnicas meta-heurísticas de resolución de problemas de optimización combinatorial, surge la necesidad de innovar "inteligencia" en los algoritmos de búsqueda local [Glover et al., 1985]. Tabú Search en contraste, con otros algoritmos establecidos en técnicas aleatorias de búsqueda de soluciones en entorno, se precisa porque este utiliza un modelo basado en el uso de *estructuras de memoria dinámicas*, la cual nos asiste en abandonar los óptimos locales en los que puede converger al desplazarse de una solución a otra, dentro de la vecindad.

Por lo anterior, se almacenan los últimos movimientos realizados; en estructuras denominadas "memorias", y que son utilizadas en llevar presente aquellos movimientos que hacen al caer de nuevo en soluciones ya analizadas. Estas estructuras servirían para impedir el posterior desarrollo de esas soluciones previamente analizadas.

Memoria	Atributos	Estrategias	Ámbito
Corto Plazo	Reciente	Intensificar	Local
Largo Plazo	Frecuente	Diversificar	Global

Fig. 3.2 *ESTRATEGIAS DEL ALGORITMO TABÚ SEARCH

6. Tabú con Umbral

El algoritmo Tabú con Umbral [Glover, 1991] surge como una versión reducida del algoritmo general de búsqueda tabú, en el que se prescinde del uso de lista tabú. Este introduce criterios probabilísticas para la elección de movimientos sin hacer referencia explícita a ninguna de las memorias. De aquí se evita la exploración repetida de soluciones, por lo que, el algoritmo tabú con umbral pretende conseguir un procedimiento de búsqueda que permita una exploración de soluciones de manera que se evite caer en aquéllas que ya han sido analizadas con anterioridad.

El algoritmo tabú con umbral consiste en la ejecución de dos fases, intensificación y diversificación. La primera consiste en una exploración donde sólo se aceptan movimientos que mejoran el valor de la función a obtener o resultado, de manera que al final se obtiene un óptimo local, en la fase de mezcla o dispersión se aceptan movimientos tanto que mejoran como que empeoran el valor de la función resultado para, de esta forma escapar del óptimo local y diversificar la búsqueda hacia otras vecindades.

Una vez definido el conjunto de movimientos posibles, de forma que abarca todo el espacio de soluciones como una combinación de éstos, el algoritmo propone dividir en subconjuntos de movimientos que pueden ser utilizados tanto en la primera como en la segunda etapa. En cada una, el subconjunto de movimientos a realizar se selecciona de manera probabilística, sin embargo no es el mismo procedimiento de selección y asignación de probabilidades el que se utiliza en ambas etapas. Tabú con umbral sustituye la lista tabú por estrategias que combinan la aleatoriedad en la selección del movimiento con la que se usa para examinar los posibles movimientos a efectuar.

Procedimiento para umbral tabú:

- 1.- En la etapa de intensificación se explora el conjunto de movimientos según la estrategia de selección aleatoria:

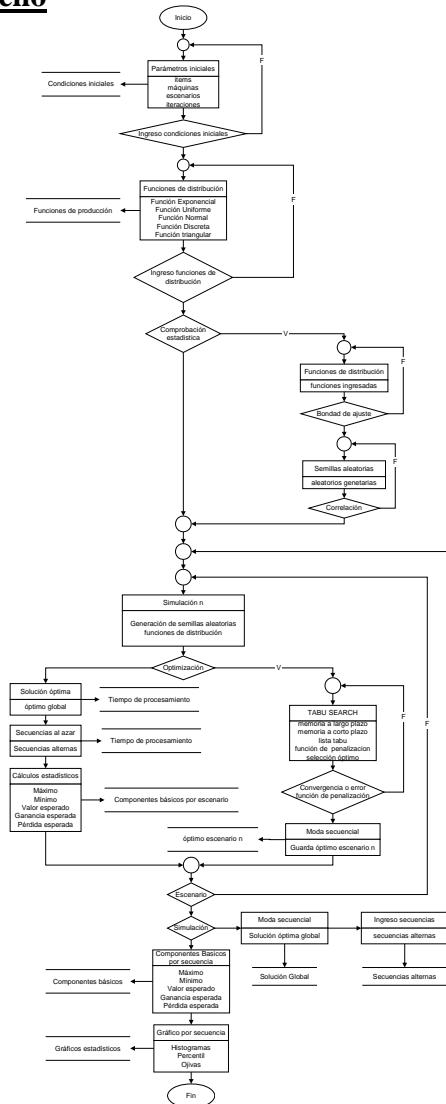
“Se divide el conjunto en bloques. En cada iteración de esta fase se realizan movimientos aleatorios dentro de cada bloque. Se elige el mejor movimiento si mejora la función objetivo. La división del conjunto en bloques y la elección aleatoria de los elementos en dichos bloques emulan el comportamiento de una lista tabú.”

2.- En la siguiente se explora el conjunto M mediante el procedimiento de selección aleatoria completa:

“Se toma un subconjunto de forma aleatoria. Se selecciona el mejor movimiento de dicho subconjunto y se acepta, tanto si mejora como si empeora la función objetivo. Este subconjunto no vuelve a ser examinado en esta fase.”

Tanto en una etapa como en otra, cada vez que un elemento es seleccionado, se busca el movimiento que conduzca a la solución óptima. La búsqueda se realiza mediante el uso de listas de candidatas, que constituyen un procedimiento de búsqueda que permite reducir el número de exploraciones.

7. Visión General del diseño



8. Planteamiento del problema.

Lo anterior, da la base para el estudio del siguiente sistema, sea una secuencia de procesos productivos la cual está gobernada por entidades exógenas y endógenas, y estas a su vez están generadas por funciones de distribución probabilística independientes (de acuerdo a lo revisado en la sección anterior). Definido nuestro entorno, procedemos a analizar las entidades que deberán ser optimizadas, primero mediante un muestreo de su comportamiento (con los datos históricos obtendremos su histograma y por consiguiente la forma de la función de distribución), definiendo de esta forma el tipo de distribución (distribución exponencial, triangular, uniforme, discreta y normal), ahora nos concentraremos básicamente en la secuencia del proceso ítems por equipo producido y los tiempos de funcionamiento y mantenimiento de cada uno, que se requieren en el proceso de producción, con el cual obtendremos una secuencia que nos garantice una utilidad óptima en cualquier escenario y a la vez se proteja contra el peor

de ellos, como se evidencia nuestra función de penalización será el tiempo de recurso en cada equipo (la utilidad máxima será inversamente proporcional, a la función de penalización “minimizar el tiempo de recurso, empleado por todos los equipos en la producción total”, esto será aplicado en el algoritmo meta-heurístico)

9. Representación Meta-Heurística de la solución.

Los requerimientos en la representación de un problema de búsqueda (Tabú Search) son:

- 1.- “Estrategia de búsqueda o de control que decide que operación aplicar.”
- 2.- “Estructura simbólica que contiene conjuntos de soluciones potenciales.”
- 3.- “Reglas que nos producen entornos depurados con posibles soluciones.”

La representación debe de ser capaz de representar los objetos. Es preferible poder representar también subconjuntos de objetos o posibles potenciales, ya que nos permite eliminar grandes espacios de búsqueda.

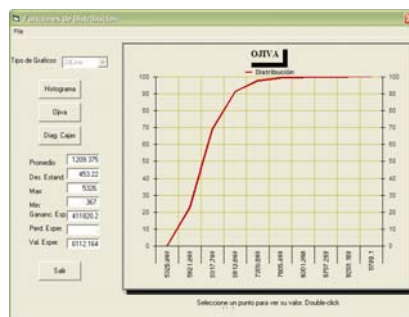
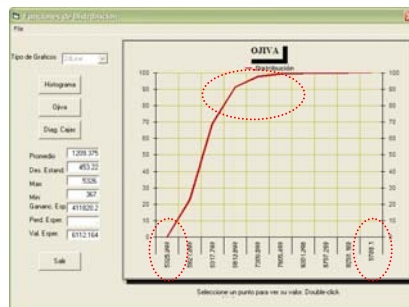
A continuación observamos un estado de la matriz a corto plazo, lista tabú, y valor de penalización o tiempo que se demora en un escenario particular la producción de la secuencia de ítems antes señaladas.

Solución actual	Estructura tabú (Reciente)							Primeros 5 candidatos
	1	2	3	4	5	6	7	Valor
1								1, 4 3 3 T
3	1			3				2, 4 -1 -6
6	2							3, 7 -3 -3 *
2	3	3			2			1, 6 -5 -5
7	4	1	5			1		6, 5 -4 -6
5	5		4	4				
4	6			1	2			Valor Penalizado
7	2			3				

Fig. 5.5 ESTRUCTURA TABÚ

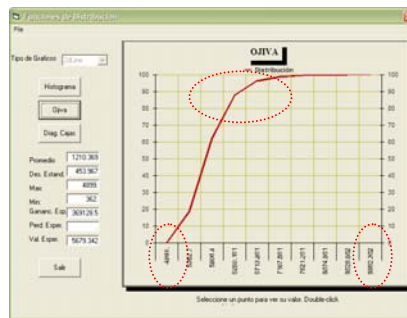
10. Análisis Función Acumulada.

En esta sección analizaremos, la función acumulada de cada secuencia escogida a la azar, incluida la secuencia óptima, para lo cual deberemos tener en cuenta que nuestra Ojiva, deberá converger en forma desacelerada en relación a las demás, debido que como indicamos el objetivo es encontrar una solución cuyo tiempo se el menor posible, es decir al inicio del periodo de tiempo (tiempo mínimo y máximo), la mayor concentración de eventos para la secuencia óptima deberá ocurrir en el margen izquierdo como lo comprobamos en los histogramas de frecuencias, con lo cual inferimos que la curva de la función acumulada inicialmente será mas levantada que las otras secuencia y al finalizar la curva deberá crecer en forma menos desacelerada que las otras curvas, otra característica importante es que el punto de intersección de la función óptima acumulada, con respecto a las demás ojivas deberá ser siempre mayor al 50%, es decir su ganancia deberá ser siempre mayor con relación a los demás.



Como observamos se cumple que los gráficos de las funciones acumuladas de las secuencias elegidas al azar, cumplen:

- a.- La convergencia del gráfico, no es rápida, en cambio en la secuencia óptima su convergencia es “acelerada”
- b.- La media de los escenarios es mayor que la media de la secuencia óptima
- c.- El punto de intersección de todas las funciones acumuladas con respecto a la secuencia óptima está por encima del 50%, esto es otra característica que identifica cual secuencia tiene una mejor ganancia o tiene el menor tiempo en procesar todos los ítems requeridos.



CONCLUSIONES

Con el presente trabajo, se ha valorado el efecto del uso de la meta-heurística: *Tabú Search* en los procedimientos de optimización con métodos estocásticos proporcionando técnicas para escaparse de los óptimos locales de mala calidad, dado que el valor de tales óptimos locales frecuentemente difiere notablemente del valor del óptimo global.

Por último y como se ha concluido a lo largo de la presente investigación:

- Resulta económicamente rentable realizar sistemas de optimización y simulación en procesos de productivos.
- No solo por la característica de no intervenir en los sistemas reales, sino por que además su tiempo de implementación así como su costo de prueba resulta efectivo
- Desde el punto de vista del tomador de decisión, sistemas como los desarrollados en la presente tesis, ayudan a probar distintos métodos de producción. Verificando su resultado final sin la necesidad de detener los procesos o generar pérdidas económicas.
- Entre los métodos meta-heurísticos más utilizados son los algoritmos de búsqueda por que los mismos están diseñados en base de procedimientos sistemáticos y no en métodos aleatorios.
- Se recalca que el hecho de disminuir el número de iteraciones, (convergencia del método *Tabú Search*) degrada significativamente la solución, para contrarrestar este efecto se debe comenzar analizar al sistema posterior al transiente de inicio de los procesos, es decir afinar la función de penalización.
- Si disminuimos el número de escenarios, el tiempo de ejecución se reduce drásticamente y la solución se deteriora levemente, por lo que dependiendo de la cantidad de procesos y la urgente necesidad de una respuesta aceptable “no óptima”, se puede disminuir el número de escenarios, pero la solución arrojada con estos parámetros jamás deberá ser considerada como una solución definitiva.
- De lo anterior se desprende, que aunque la Meta-Heurística: *Tabú Search* tenga un eficiente método de análisis, a través de su procedimiento de memorias: corto y largo plazo; su mayor inconveniente para analizar sistemas de procesos, estará ligada a las características del equipo en el cual se desee ejecutar el sistema de simulación y optimización, ya que la mayoría de los procesos que estos ejecutan se los realizan en la memoria del equipo.
- El sistema actual debido a uso es académico tiene la limitante de 50 variable aleatoria independientes de entrada (funciones de distribución de las variables de entrada), esto es fácilmente de mejorar, si aumentamos la cantidad de semillas independientes. (para confirmar que son semillas independiente verificar la correlación de estas con respecto a cada una de las ingresadas)

REFERENCIAS

- [1.] SOBOL I., *El Método de Monte Carlo*, Editorial MIR
- [2.] J.M. HAMMERLEY & D.C. HANDSCOMBS, *Monte Carlo, Methods*
- [3.] GEORGE D. FISHMAN, *Conceptos y Métodos en la Simulación Digital de Eventos Discretos*
- [4.] GEOFFREY GORDON, *Simulación de Sistemas*.
- [5.] JAIME BARCELÓ, *Simulación de Sistemas Discretos*, Publicaciones de Ingeniería de Sistemas.
- [6.] RICARDO SANZ BRAVO, AGUSTÍN JIMÉNEZ AVELLO, MANUEL-ALONSO CASTRO, *Sistema de Control Inteligente, Modelo y Simulación*, Departamento de Automática, Ingeniería Electrónica e Informática industrial. Universidad Politécnica de Madrid. 1992.
- [7.] P. KOUVELIS Y G. YU. *Robust Discrete Optimization and Its Applications*. Kluwer Academic Publishers, Dordrecht, Holanda, 1997.
- [8.] ANTONIO CREUS SOLE MARCOMBO, *Simulación y Control de Procesos por Ordenador*
- [9.] ANGEL A. SARABIA, *La Teoría General de los Sistemas*, Publicaciones de Ingeniería de Sistemas.
- [10.] JAVIER ARACIL, *Dinámica de Sistemas*, Publicaciones de Ingeniería de Sistemas.
- [11.] DONAL R. DREW, *Dinámica de Sistemas Aplicada*, Publicaciones de Ingeniería de Sistemas.
- [12.] XAVIER PI JOSEP CASANOVAS, *Software de Simulación*, Facultad de Informática de la UPC, Barcelona Revista Automática e Instrumentación, Junio 1996.
- [13.] E. H. L. AARTS, J. H. M. KORST, Y P. J. M. VAN LAARHOVEN, *Simulated annealing*, E. H. L. Aarts y J.K. Lenstra, editores *Local Search in Combinatorial Optimization*, páginas 91-120. John Wiley & Sons, Chichester, 1997. 1
- [14.] E. H. L. AARTS Y J. K. LENSTRA, editors, *Local Search in Combinatorial Optimization*, John Wiley & Sons, Chichester, 1997. 2
- [15.] R. ALCALÁ, J. CASILLAS, O. CORDÓN, Y F. HERRERA, *Improvement to the cooperative rules methodology by using the Ant Colony System algorithm*, *Mathware & Soft Computing*, 8: 3, páginas 321-335, 2001. 3
- [16.] S. ALONSO, O. CORDÓN, I. FERNÁNDEZ DE VIANA, F. HERRERA, *Análisis de distintas vertientes para la paralelización de los algoritmos de Optimización basada en Colonias de Hormigas*. *Actas del Segundo Congreso Español de Meta heurísticas, Algoritmos Evolutivos y Bioinspirados (MAEB03)*, páginas 160-167, Gijón, 2003. 4
- [17.] T. BÄCK, D. FOGEL, Z. MICHALEWICZ, editors, *Handbook of Evolutionary Computation*, 1998 6
- [18.] S. BALUJA AND R. CARUANA, *Removing the genetics from the standard genetic algorithm*, A. Prieditis y S. Russel, editores, *Proceedings of the Twelfth International Conference on Machine Learning (ML-95)*, pages 38-46. Morgan Kaufmann Publishers, Palo Alto, CA, 1995. 7
- [19.] M. BIRATTARI, T. STÜTZLE, L. PAQUETE, Y K. VARRENTRAPP, *A racing algorithm for configuring metaheuristics*, W.B. Langdon y otros, editor, *GECCO 2002: Proceedings of the Genetic and Evolutionary Computation Conference*, páginas 11-18. Morgan Kaufmann Publishers, San Francisco, CA, 2002. 9
- [20.] O. CORDÓN, I. FERNÁNDEZ DE VIANA, F. HERRERA, Y L. MORENO, *A new ACO model integrating evolutionary computation concepts: The Best-Worst Ant System*. M. Dorigo, M. Middendorf, y T. Stützle, editores, *Abstract proceedings of ANTS2000 - From Ant Colonies to Artificial Ants: A series of International Workshops on Ant Algorithms*, páginas 22-29. IRIDIA, Université Libre de Bruxelles, Belgium, 2000. 23
- [21.] O. CORDÓN, F. HERRERA, L. MORENO, *Integración de Conceptos de Computación Evolutiva en un Nuevo Modelo de Colonias de Hormigas*. *VIII Conferencia de la Asociación Española para la Inteligencia Artificial, (Seminario Especializado en Computacion Evolutiva)*, Murcia (España), 1999, Vol. II, páginas 98-105. 24

- [22.] O. CORDÓN, F. HERRERA, T. STÜTZLE, *A Review on Ant Colony Optimization Metaheuristic: Basis, Models and New Trends*. Mathware & Soft Computing, 9: 2-3, páginas 141-175, 2002. 25
- [23.] D. COSTA AND A. HERTZ, *Ant can colour graphs*. *Journal of the Operational Research Society*, 48, páginas 295-305, 1997. 26
- [24.] G. DI CARO AND M. DORIGO. ANTNET, *Distributed stigmergetic control for communications networks*. *Journal of Artificial Intelligence Research*.
- [25.] M. DORIGO Y G. DI CARO, *The Ant Colony Optimization meta-heuristic*. En D. Corne, M. Dorigo, y F. Glover, editores, *New Ideas in Optimization*.
- [26.] M. DORIGO, G. DI CARO, Y L. M. GAMBARDELLA, *Ant algorithms for discrete optimization*. *Artificial Life*, 5: 2, páginas 137-172, 1999. 31
- [27.] M. DORIGO, V. MANIEZZO, Y A. COLORNI, *The Ant System: Optimization by a colony of cooperating agents*. *IEEE Transactions on Systems, Man, and Cybernetics - Part B*, 26: 1, páginas 29-41, 1996. 33
- [28.] M. DORIGO Y T. STÜTZLE, *Ant Colony Optimization*. MIT Press, Cambridge, MA. En prensa. 35
- [29.] M. DORIGO Y T. STÜTZLE, *The ant colony optimization metaheuristic: Algorithms, applications and advances*. En F. Glover and G. Kochenberger, editores, *Handbook of Metaheuristics*, páginas 251-285. Kluwer Academic Publishers, 2003. 36
- [30.] T. A. FEO Y M. G. C. RESENDE, *Greedy randomized adaptative search procedures*. *Journal of Global Optimization*, 6, páginas 109-133, 1995. 39
- [31.] P. FESTA Y M. G. C. RESENDE. GRASP, *An annotated bibliography*. En P. Hansen y C. C. Ribeiro, editores, *Essays and Surveys on Metaheuristics*, Kluwer Academic Publishers, 2001. 40
- [32.] R. A. FISHER, *Theory of Statistical Estimation*. *Proceedings of the Cambridge Philosophical Society*, 22, páginas 700-725, 1925. 41.
- [33.] R.A. FISHER, *The Comparison of Samples with Possibly Unequal Variances*, *Annals of Eugenics*, 9, páginas 174-180, 1936. 42
- [34.] L. M. GAMBARDELLA AND M. DORIGO. ANT-Q, A. Prieditis y S. Russell, editores, *Proceedings of the Twelfth International Conference on Machine Learning (ML-95)*.
- [35.] M. R. GAREY Y D. S. JONSON, *Computers and Intractability: A Guide to the Theory of NPCompleteness*, Freeman, San Francisco, CA, 1979. 47
- [36.] F. GLOVER AND G. KOCHENBERGER, editores. *Handbook of Metaheuristics*. Kluwer Academic Publishers, 2003. 48
- [37.] F. GLOVER Y M. LAGUNA, *Tabú Search*. Kluwer Academic Publishers, Boston, MA, 1997. 49
- [38.] W. J. GUTJAHR, *A Graph-based Ant System and its convergence*. *Future Generation Computer Systems*, 16: 8, páginas 873-888, 2000. 54
- [39.] W. J. GUTJAHR, *ACO* *Information Processing Letters*, 82: 3, páginas 145-153, 2002. 55
- [40.] P. HANSEN Y N. MLADENOVIC, S. Voss, S. Martello, I. H. Osman, y C. Roucairol, editores, *MetaHeuristics - Advances and Trends in Local Search Paradigms for Optimization*, páginas 433-458. Kluwer Academic Publishers, Dordrecht, Holanda, 1999. 57
- [41.] J. L. HENNESY, D. A. PATTERSON, *Computer Architecture, A Quantitative Approach*, páginas 40-42, Morgan Kaufmann Publishers, 2003. 58
- [42.] J. HOLLAND, *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor, 1975. 60
- [43.] NAIM A. KLAIR, *System Modeling and Simulation*, Electrical Engineering and Electronics Marcel Dekker, Inc.
- [44.] L. BLUM M., SHUB M., *A Simple unpredictable pseudorandom number generator*, *SIAM Journal on Computing* 15, pp 364-383, 1986 Blum.
- [45.] FIPS 140, *Security requirements for crytographic modules*, Federal Information Processing Standards Publication 140-I, U.S. Department of Commerce / N.I.S.T., National Technical Information Service, Springfield, Virginia, 1994.

- [46.] KALISKI JR. B. S., PhD Thesis, Elliptic curves and cryptography; a pseudorandom bit Generator and other tools, MIT Department of Electrical Engineering and Computer Science, 1988.
- [47.] KNUTH D.E., ADDISON QESLEU READING, The Art of Computer Programming – Seminumerical Algorithms, Vol 2, 1973.
- [48.] MAURER U.M., A Universal Statistical Test for Random Bit Generators, Advances in Cryptology Cripto '90, LNCS 537, pp 409-420, 1991.
- [49.] MENEZES A.J. VAN OORSCHOT P.C., HandBooj of Applied Cryptography, Vanstone S.A., CRC Press 1997.
- [50.] MICALI S., Efficient, perfect polynomial random number generators, Journal of Cryptology 3, Schnorr pp 157-172,1991
- [51.] SHAMIR A., On the generation of cryptographically strong pseudorandom sequences, ACM Transaction on Computer Systems, 1, 1983.
- [52.] ALBERTO ARAGON TORRES, Métodos Evolutivos para el aprendizaje de Redes Neuronales, Junio 2002 pag. 134 – 206.
- [53.] JOSÉ FLORENTINO AUGUSTO MEDINA JACOBO, Un enfoque robusto a un problema de diseño de red multiproducto con incertidumbre en parámetros de entrada, Mayo 2005 pag. 134 – 206, Tesis en opción al grado de maestro en ciencias en ingeniería de sistemas.
- [54.] Belén Melián, José A. Moreno Pérez, J. Marcos Moreno Vega, Metaheuristics: A global view Departamento de Estadística, I.O. y Computación, Centro Superior de Informática, Universidad de La Laguna