Métodos de Aproximación para Problemas de Planificación.

M. Nussbaum, E. Laval, y M. Sepúlveda
Departamento de Ciencia de la Computación
Escuela de Ingeniería, P. Universidad Católica
Casilla 306, Santiago 22, CHILE
FAX (56 2) 552 4054

Keywords: Inteligencia Artificial, Simmulated Annealing, Tabu Search.

Abstract.

Los problemas de planificación generalmente son problemas de optimización del tipo combinatorio. Estos se caracterizan por un espacio de búsqueda donde las posibles soluciones son finitas, o infinitas pero numerables. Aún cuando las posibles soluciones sean finitas, su número puede crecer en forma exponencial con el tamaño del problema. Los métodos tradicionales de optimización, basados en modelos matemáticos, no han podido solucionar un porcentaje importante de los problemas de planificación. En la búsqueda de mejores metodologías han surgido métodos híbridos de la Inteligencia Artificial y la Investigación Operativa los cuales son investigados en el presente trabajo.

1. Introducción.

En la mayoría de las empresas existen problemas de planificación en los cuales se deben tomar decisiones sobre que recursos asignar y como asignarlos. Estos problemas generalmente son problemas de optimización del tipo combinatorio, en los cuales el número de soluciones crece en forma exponencial con el tamaño del problema.
Los enfoques con que se han enfrentado los problemas de planificación tales como los modelos matemáticos, expertos humanos, sistemas expertos [1], no han sido suficientes para resolver este tipo de problemas. Por lo general, las soluciones no son óptimas y tienen tiempos de respuesta elevados.
En la búsqueda de mejores metodologías han surgido métodos híbridos de la Inteligencia Artificial y la Investigación Operativa. En estos métodos se integran heurísticas de búsqueda con algoritmos de optimización.Una línea de investigación que sigue esta tendencia es la de los métodos de Aproximación. Estos se han utilizado con éxito para enfrentar problemas de optimización combinatorios, como se verá en este trabajo.

2. Métodos de Aproximación.

2.1. Descripción.

Los problemas de optimización combinatorios son aquellos que poseen variables discretas y se caracterizan por la búsqueda de un elemento óptimo en un conjunto finito, o infinito pero contable de posibles elementos. Los elementos típicos del espacio de búsqueda son números enteros, conjuntos, permutaciones o grafos [2]. A cada elemento se le asigna un valor real mediante una función de evaluación.
Los Métodos de Aproximación se caracterizan por ir transitando por distintos elementos del espacio de búsqueda, tratando de aproximarse progresivamente al óptimo, para lo cual se escoge uno de los vecinos del elemento actual (generado al realizar un cambio pequeño sobre el elemento actual). De esta definición podemos destacar algunos elementos claves que caracterizarán los distintos métodos de aproximación:
Solución inicial: Los métodos de aproximación comienzan el proceso de aproximación partiendo de un elemento, o solución, inicial.
Vecinos de un elemento: Para transitar de un elemento a otro, se debe definir cuál es la vecindad de un elemento.

Este Trabjo fue finaciado parcialmente por el Fondo Nacional de Ciencia y Tecnología de Chile, proyecto 92842

Elección de la próxima solución: La siguiente solución debe escogerse de entre los vecinos, pero no siempre se deben evaluar todos ellos para escogerla. Es posible focalizarse en un grupo de vecinos mediante algún criterio de muestreo estadístico, o alguna heurística de selección.
Criterio de detención: Los métodos de aproximación continuarán transitando de una solución a otra hasta que se cumpla un cierto criterio de detención.

2.2. Métodos existentes.

En la literatura existen diversos métodos, ya tradicionales, que entran en la categoría de métodos de aproximación. Un método muy simple, es el método de Hill Climbing el cual se caracteriza por escoger al mejor de los vecinos mientras se mejore la evaluación con respecto al elemento actual. Este método tiene la gran desventaja de quedar atrapado en óptimos locales [3]. Entre los diversos métodos de aproximación algunos tienen la cualidad de evitar los óptimos locales, entre elos cabe destacar Simulated Annealing y Tabu Search.

3.3.1. Simulated Annealing.

El método de Simulated Annealing [4] asimila el proceso de búsqueda de una solución óptima con el fenómeno físico de annealing. Este fenómeno se refiere al proceso en el cual se busca el estado de mínima energía de un sólido, disminuyendo lentamente su temperatura desde un estado líquido hasta que se encuentre cristalizado [5].
En el campo de problemas combinatorios el método es el siguiente:

- Se tiene un espacio de búsqueda con posibles soluciones del problema, una de las cuales es la óptima.
- Se escoge una solución inicial y se mantiene un parámetro (la temperatura en un valor alto)
- Se escoge un vecino de la solución actual en forma aleatoria. Este vecino será aceptado como una nueva solución de acuerdo con el siguiente criterio:
  - Si el vecino tiene una mejor evaluación que la solución actual (se acerca al óptimo), se acepta con una probabilidad 1.
  - Si el vecino tiene una peor evaluación que la solución actual (se aleja del óptimo), será aceptado con una probabilidad que depende de la temperatura y de cuánto empeore la evaluación.

El valor del parámetro que representa la temperatura es inicialmente alto, de manera que se acepten vecinos malos con una probabilidad cercana a 1. A medida que evoluciona el proceso la temperatura disminuye lentamente hasta llegar a 0, caso en el que no se aceptan vecinos malos. De esta manera se realiza un recorrido por distintas zonas del espacio de búsqueda, sin quedar atrapado en óptimos locales.
A continuación se describe la manera en que Simulated Annealing responde a las interrogantes generales de los métodos de aproximación que se plantearon previamente:
¿ Cómo escoger la solución inicial ? Debido a la alta aleatoridad inicial del proceso, no tiene sentido escoger una "buena" solución inicial, o sea una solución que sea cercana al óptimo. Basta con una solución inicial aleatoria.
¿ Cómo se definen los vecinos de la solución actual ? El método no tiene asociada una definición de vecindad, está dependerá de cada problema en particular.
¿ Cómo escoger la próxima solución de entre esos vecinos ? Se escoge en forma aleatoria un vecino y es aceptado de acuerdo a la función probabilística descrita anteriormente.
¿ Cuándo detenerse ? La condición de termino depende del problema en particular, usualmente se refiere a un límite de iteraciones efectuadas o a iteraciones sin obtener mejoras en la solución.

3.3.2. Tabu Search.

El método de Tabú Search [6][7] también se basa en acercarse al óptimo mediante aproximaciones sucesivas. Una de las principales características de este método es el utilizar un mecanismo explícito para evitar quedar atrapado en óptimos locales: la lista tabu. Estas son soluciones que no pueden escogerse durante un cierto número de iteraciones. En cada iteración se genera el conjunto de vecinos de la solución actual, de este conjunto se eliminan aquellos que pertenezcan a la lista tabu, y se escoge el "mejor" de ellos. La nueva solución se acepta, aún cuando esta tenga una peor evaluación que la solución antigua. Cada nueva solución escogida se incorpora a la lista tabú, evitando así retroceder por caminos recientemente recorridos. En el contexto del Tabú Search, la lista tabú es conocida como un procedimiento de memoria de corto plazo. Otros métodos de memoria, de más largo plazo, también son utilizados para buscar la intensificación y diversificación de la búsqueda.

En base al método básico de Tabu Search se responden a continuación las interrogantes sobre los métodos de aproximación:
¿ Cómo escoger la solución inicial ? En Tabu Search una "buena" solución inicial permitirá obtener resultados en forma más rápida. Generalmente se utilizan métodos heurísticos relacionados con el problema para obtener soluciones más cercanas al óptimo que las determinadas aleatoriamente.
¿ Cómo se definen los vecinos de la solución actual ? Dependerá de cada problema en particular y de la manera en que se modelen las soluciones.
¿ Cómo escoger la próxima solución de entre esos vecinos ? Antes de escoger el vecino óptimo se descartan los vecinos pertenecientes a la lista tabu.
¿ Cuándo detenerse ? La condición de termino depende del problema en particular.

3. Problemas de planificación resuelta con un enfoque de aproximación.

3.1. Descripción.

Dada la complejidad de los problemas de planificación y los resultados poco satisfactorios obtenidos con las técnicas clásicas, se han buscado nuevos enfoques. El que se presenta a continuación corresponde a utilizar Métodos de Aproximación para resolver problemas de planificación, y cumple con los siguientes requisitos básicos:

- Un modelo para representar problemas de planificación en forma estándar.
- Un conjunto de estrategias concretas para realizar la aproximación.
- Un grado de flexibilidad para adaptar el sistema a cada problema específico.

3.2. Modelo Genérico: Secuencia de elementos.

Se desea utilizar una misma herramienta para distintos problemas de planificación. Para ello se deben representar los distintos problemas de una manera estándar, utilizando un modelo genérico para una gran cantidad de problemas de planificación. El modelo genérico que se utilizó fue el de una secuencia de elementos, a la cual se le busca una permutación óptima. Este modelo se aplica a distintos problemas de planificación ya sea en forma directa o indirecta utilizando un mapeo del problema a uno de secuencias.

3.3. Estrategias concretas de búsqueda.

Las estrategias de búsqueda corresponden a la manera en que se definen los elementos claves de los métodos de aproximación (las preguntas definidas anteriormente). El sistema planificador debería proveer alternativas concretas para responder estas preguntas, de tal manera que el usuario no se preocupe de cómo implementar la búsqueda para cada problema en particular. En los siguientes puntos se describen las estrategias de búsqueda que se han definido para el sistema planificador, sin perjuicio de que en el futuro se vayan desarrollando nuevas estrategias.

3.3.1. Generación de la secuencia inicial.

La secuencia inicial puede ser de gran importancia para el desempeño de la búsqueda. En general una secuencia inicial con una evaluación cercana al óptimo, permitirá acercarse con mayor rapidez al óptimo. Los siguientes son los criterios para definir la secuencia inicial:
Secuencia Aleatoria: Ordenar los elementos de la secuencia al azar.
Utilizar Secuenciamiento Parcial: Esta es una heurística genérica basada en la heurística NEH [referencia] que consiste en comenzar con una secuencia de un solo elemento e ir insertando nuevos elementos que se ubican en la posición que otorgue la mejor secuencia parcial.
Secuencia sugerida por el usuario: El usuario define la heurística.

3.3.2. Definición de vecinos.

Los vecinos de una secuencia son todas aquellas secuencias que se geñeran al aplicar un cambio pequeño sobre ella. Los vecinos son definidos de acuerdo a tres tipos de cambios básicos que se aplicna a la secuencia actual:
Cambios tipo Swap: Intercambiar los elementos que están en dos posiciones dadas.
Ejemplo: (A B C D E F) swap(5,2) -> (A E C D B F)
Cambio tipo Shift: Insertar un elemento de una posición en otra posición.
Ejemplo: (A B C D E F) shift(5,2) -> (A E B C D F)
Cambio tipo Invert:Invertir la subsecuencia comprendida entre dos posiciones
Ejemplo:(A B C D E F) invert(5,2) -> (A E D C B F)

3.3.3. Como escoger el siguiente vecino.

La manera en que se escoge el siguiente vecino es uno de los puntos más determinantes en el proceso de búsqueda. Los criterios serán divididos en tres tipos:

Reducción del espacio de búsqueda local. Se restringe el número de vecinos en casos en los cuales el número de vecinos es elevado o la evaluación de cada uno es lenta. Para reducir el espacio de búsqueda local se propone hacer muestreos aleatorios.
Uso de inteligencia Se refiere a métodos que permitan orientar la búsqueda, sugiriendo algunos vecinos que deberían, o no, escogerse. En particular están los métodos explícitos para evitar óptimos locales. El primero de éstos son las las listas tabú, el cua consiste en almacenar una lista que identifique las secuencias visitadas en las últimas t iteraciones, las cuales no podrán ser visitadas. Un segundo método es prohibir malos, en el cual se supone que si se realiza un cambio en una secuencia y con éste se generan otra secuencia con muy mala evaluación, este cambio es prohibido por un cierto número de iteraciones.
Criterios de aceptación. Se refiere al cómo escoger la siguiente secuencia una vez que ya se ha reducido el espacio de búsqueda y se han aplicado criterios de inteligencia. Se proponen tres criterios. Escoger al mejor vecino: Se escoge al mejor vecino independiente de sí este mejora o empeora la evaluación actual. Utilizar Hill Climbing: Se escoge al mejor de los vecinos siempre y cuando éste mejore la evalución de la secuencia actual, en caso contrario se mantiene la secuencia actual. Aceptación Probabilística: Este es el criterio utilizados en Simulated Annealing. El mejor de los vecinos se aceptará si mejora la evaluación con respecto a la secuencia actual. En caso contrario se aceptará con cierta probabilidad.

3.3.4. Cuándo detenerse.

Se proveen los siguientes criterios de detención. Límite de tiempo: Fijar la cantidad máxima de segundos para que el sistema realice aproximaciones. Límite de iteraciones: Fijar la cantidad máxima de iteraciones a realizarse. Resultado alcanzado: Detenerse al llegar a una solución óptima, o cercana al óptimo. Resultado alcanzable: Realizar una estimación de cuanto más será posible aproximarse al óptimo. Esta estimación puede realizarse de acuerdo a la velocidad de aproximación. Estos criterios se utilizan en conjunto, produciéndose la detención al satisfacerse el primero de ellos.

3.4. Flexibilidad.

El sistema de planificación propuesto provee de una serie de estrategias y parámetros distintos para realizar la planificación. Si se utilizará exactamente la misma estrategia de búsqueda para todos los problemas, se tendría un sistema extremadamente rígido que puede entregar resultados muy buenos en algunos casos, pero en otros casos otorgar resultados malos o en forma ineficiente.
Se propone que el usuario pueda escoger estrategias para adaptar el sistema a sus problemas concretos. La elección de estrategias puede hacerse de acuerdo a la experiencia acumulada por otras personas, o bien 'sintonizando' el sistema mediante sucesivas prueba. En la práctica el usuario dispone de un Script donde especifica especifica los criterios utilizados para escoger la solución inicial, escoger los vecinos de una solución, escoger la próxima solución de entre los vecinos y el criterio de detención.

4. Implementación de la Shell.

Con el fin de implementar la metodología propuesta en la sección anterior y ofrecer al usuario una herramienta que le permita enfrentar cualquier tipo de problema en forma expedita, se desarollo una shell.

4.1. Descripción de la Shell.

4.1.1. Componentes.

En la siguiente figura se aprecia un esquema del sistema:

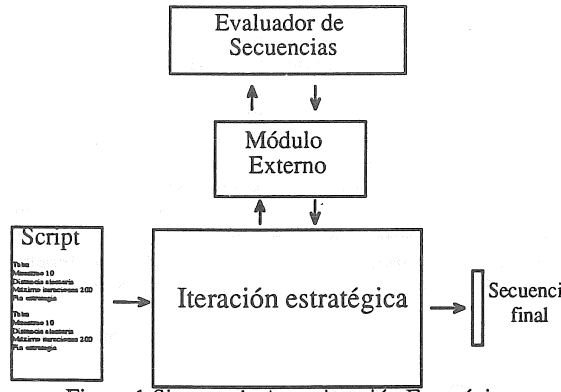

Figura 1 Sistema de Aproximación Estratégica

Iteración estratégica. Componente principal formada por una serie de módulos que permiten realizar cada iteración de acuerdo a las estrategias definidas por el usuario.
Script. Archivo de texto en el cual el usuario puede especificar las estrategias que desea utilizar para realizar la búsqueda. Para ello se provee de un lenguaje declarativo que le permite disponer de los distintos criterios como más le convenga. La Figura 2 ilustra un ejemplo.

```
tipo_movida swap
muestreo 5
distancia_aleatoria
tabu 7
prohibir_malos 4
máximo_iteraciones 1000
máximo_sin_mejora 50
máximo_rango 10
fin_estrategia
```

Figura 2. Script que especifica las estrategias a utilizar
Módulo Externo. Permite la comunicación del módulo principal con el planificador de secuencias (sistema externo independiente).

4.1.2.Funcionamiento.

Con el fin de utilizar la shell, el usuario debe realizar los siguientes pasos:
Modelación del problema. Modelar el problema de manera tal que la planificación dependa de la existencia de un secuenciamiento de elementos. El evaluador de secuencias debe ser capaz de evaluar la bondad de cada secuenciación.
Creación del módulo externo. Crear las funciones componentes del módulo externo descrito en la sección anterior. Una vez realizada esta interfaz, el usuario dispondrá rápidamente de una herramienta para solucionar su problema.
Script. Una vez establecida la comunicación del evaluador con el sistema, el usuario puede establecer cual estrategia es la que permite abordar mejor su problema utilizando el script.
Output. Los resultados de la planificación son desplegados en una serie de archivos de salida que contienen la secuenciación óptima del problema el valor óptimo encontrado.

4.1.3.Ambiente de desarrollo.

La shell fue desarollada en lenguaje C, en ambiente Unix. La plataforma de hardware utilizada fue una SUN SPARC Station 1.

4.2. Ejemplo Real: Asignación de tareas a máquinas.

Descripción.
Una empresa debe realizar N trabajos cumpliendo ciertas restricciones de fechas de inicio y término.Para realizar los trabajos se dispone de M máquinas. Cada trabajo se descompone en una serie de etapas, cada una de las cuales debe realizarse en una máquina por un período determinado de tiempo. Se tiene la restricción de que las máquinas no pueden utilizarse para más de una tarea

simultáneamente, y éstas requieren un tiempo de puesta en marcha entre tarea y tarea. Distintos trabajos pueden requerir las mismas máquinas. De esta manera no todos los trabajos se podrán realizar en cualquier instante, sino que dependen de la utilización de las máquinas por el resto de los trabajos. El problema consiste en asignarle a cada máquina un calendario de tareas de manera que se puedan cumplir todos los trabajos con el mínimo tiempo de retraso. Un enfoque para este problema es el planificar sólo una tarea a la vez, de acuerdo con la disponibilidad de maquinas. Esta asignación se puede hacer con una heurística simple. Luego el problema se transforma en determinar el orden en el cual se deben hacer las asignaciones de las N tareas, osea encontrar una secuencia óptima de N elementos. En este caso la evaluación de cada secuencia implica todo un proceso de planificación de cada elemento en forma individual. La función utilizada para evaluar la planificación es la Tardanza Cuadrática Acumulativa (TCA), la cual depende de los días de atraso de cada trabajo. Esta función castiga los días de atraso en forma polinomial.

Modelación.

De acuerdo a lo señalado en la sección 4.1.2, los elementos son los trabajos pedidos para los cuales se debe encontrar un secuenciamiento óptimo.

Determinación de estrategias más apropiadas.

Generación de la secuencia inicial. Para la generación de la secuencia inicial se utilizó el secuenciamiento parcial. Los resultados obtenidos permitieron generar un buen punto de partida, en algunos casos muy cercanos al óptimo.

Tipos de cambio. Para seleccionar el tipo de cambio se realizaron pruebas utilizando distintas movidas. El Gráfico 1 muestra los resultados obtenidos los cuales sugieren que la estrategia adecuuada al problema es preferir movidas del tipo swap.

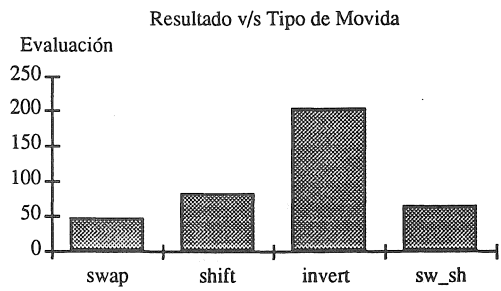

Gráfico 1. Tipos de movida

Reducción del espacio de búsqueda local. Se analizaron ocho tipos de muestreo (A...H) para determinar la mejor forma de acotar el espacio de búsqueda. Estos se constrastaron con el valor óptimo posible de la función de costos (0) y el tiempo de CPU requerido. (Gráfico 2).

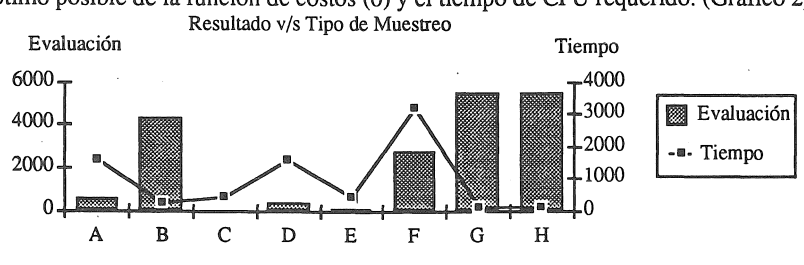

Gráfico 2. Tipos de Muestreo

Los resultados obtenidos sugieren que para este problema se debe utilizar un muestreo del tipo C, el cual consiste en escoger 10 elementos de la secuencia y evaluar la factibilidad de hacer un swap de ellos con otras 5 posiciones de la secuencia.

Uso de inteligencia. Gráfico 3 ilustra los resultados de aplicar listas tabú y listas de prohibir malos. Se realizaron pruebas con listas tabú de distintos tamaños comprobándose que para el problema en estudio es preferible no utilizar dichas listas. Distinto es el caso con listas de prohibir malos donde, para este problema, es conveniente fijar el período de prohibición en 3 iteraciones.

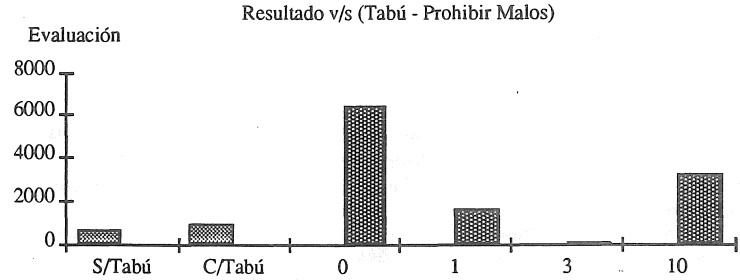

Gráfico3. Uso de inteligencia

Criterios de Aceptación. Es necesario decidir si se avanza o no a la mejor secuencia encontrada entre las vecinas. Las pruebas realizadas (Gráfico 5.4) determinan que es conveniente, para el problema estudiado, aceptar el nuevo estado en forma probabilística.

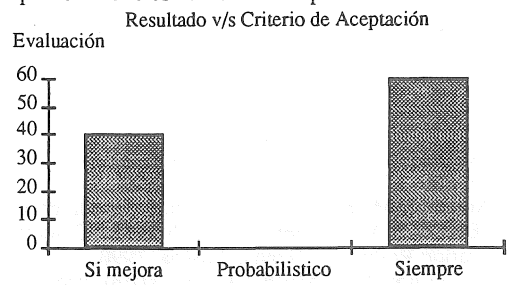

Gráfico 5.4. Criterios de Aceptación

Cuándo detenerse. Haciendo un estudio de todas las pruebas, en las cuales se utilizaron límites relativamente grandes, se llega a los siguientes valores:

Límite de tiempo	1800 seg
Límite de iteraciones	3000
Límite de iteraciones Sin Mejora	600
Resultado óptimo	0

5. Conclusiones

En este trabajo se presenta un enfoque muy simple y poderoso a la vez para enfrentar problemas de planificación. El sistema propuesto permite enfrentar en forma sencilla diversos problemas de planificación obteniendo excelentes resultados. Estos son obtenidos con bajo costo de desarrollo al no tener que desarrollar un planificador a la medida. Los resultados obtenidos con los problemas reales han validado la utilidad del sistema y los métodos de búsqueda que están bajo él.

En el futuro cercano se espera incorporar mayor inteligencia a los métodos de búsqueda a fin de hacer más eficientes los procesos. En el sistema actual el usuario debe tener un gran conocimiento tanto del problema como de la herramienta optimizadora. Un desafío es que el sistema se sintonice automáticamente para operar con la mejor estrategia para diferentes problemas de planificación. De esta manera, el usuario podrá efectivamente desentenderse del cómo se resuelven los problemas.

6. Bibliografía.

[1]: Parra E. and Nussbaum M., "A Production Scheduling System", Departamento de Ciencia de la Computación, Pontificia Universidad Católica de Chile, Casilla 6177, Santiago, Chile, 1990. Enviado al Journal on Computing Operations Research Society.

[2]: Papadimitrou C., Steiglitz K., "Combinatorial Optimization: Algorithms and Complexity", Prentice-Hall, Englewood Cliffs, N.J.,1982.

[3]: Glover F., "Candidate List Strategies and Tabu Search", 1989a.
[4]: Kirkpatrick S, Gelatt Jr. and Vecchi M.P., "Optimization by Simulated Annealing", Science 220, 671-680, 1983.
[5]: Laarhoven P.J.M van, "Theorical and computational aspects of simulated annealing", Ph. D. thesis, Erasmus Universiteit Rotterdam, 1988.
[6]: Glover F. "Tabu Search-Part I", Orsa Journal on Computing, Vol 1, No 3, 190-206, 1989b.
[7]: Glover F. "Tabu Search-Part II", Orsa Journal on Computing, Vol 2, No 1, 4-32, 1990.
[8]: de Werra D. and Hertz A., "Tabu Search Techniques: A Tutorial and Application to Neuronal Networks", OR Spektrum 11, 131-141, 1989.
[9]: Dueck G. and Scheuer T., "Threshold Accepting. A General Purpose Optimization Algorithm Appearing Superior to Simulated Annealing", 1988.
[10]: Faigle U. and Kern W. , "Some Convergence Results for Probabilistic Tabu Search", Memorandum No 822, Faculty of Applied Mathematics, University of Twente, 1989.
[11]: Glover F. and Greenberg H., "New Approaches for Heuristic Search: A Bilateral Linkage with Artificial Intelligence", European Journal of Operational Research, vol. 39, No. 2, 119-130, 1989.
[12]: Glover F. and Laguna M., "Target Analysis to Improve a Tabu Search Method for Machine Scheduling", Technical Report, Advanced Knowledge Research Group, US West Advanced Technologies, Boulder, Colorado, 1989.
[13]: Goldberg D., "Genetic algorithms in search, optimization, and machine learning, Addison-Wesley Pub. Co., Inc, 1989.
[14]: Hertz A. and de Werra D., "The Tabu Search Metaheuristic: How we used it", Research Report ORWP 88/13, Ecole Polytechnique Fédérale de Lausanne, Departement de Mathématiques, 1988, to appear in Annals of Mathematics and Artificial Intelligence.
[15]: Khuri S. and Batarekh A., "Genetic Algorithms and Discrete Optimization", International Conference On Operation Research, Vienna, Austria, 1990.
[16]: Kuik R. and Salomon M., "Multi-level lot-sizing problem: Evaluation of a simulated annealing heuristic", European Journal of Operational Research 45, 25-37, 1990.
[17]: Laguna M., Barnes J. and Glover F., "Scheduling Jobs with Linear Delay Penalties and Sequence Dependent Setup Costs and Times Using Tabu Search", Research Report, Department of Mechanical Engineering, The University of Texas-Austin, 1989.
[18]: Nawaz M., Enscore E.E. and Ham I., "A Heuristic Algorithm for the m-machine, n-Job Flow Shop Sequencing Problem", OMEGA, Int. J. of Mgmt Sci, vol 11, No 1, 1983.
[19]: Papadimitrou C., Steiglitz K., "Combinatorial Optimization: Algorithms and Complexity", Prentice-Hall, Englewood Cliffs, N.J.,1982.
[20]: Tsang E. and Warwick T., "Applying Genetic Algorithms to Constraint Satisfaction Optimization Problems". Proceedings of the Ninth European Conference on Artificial Intelligence, 649-654, 1990
[21]: Whitley D., Starkweather T. and Fuquay D., "Scheduling Problems and Traveling Salesmen: The Genetic Edge Recombination Operator", Proceedings of the Third International Conference on Genetic Algorithms, 133-140, 1989.
[22]: Widmer M. and Hertz A., "A new Approach for Solving the Flow Shop Sequencing Problem", ORWP 87/15, Department of Mathematics, University of Lausanne, 1987, to appear in European Journal of Operational Research.