

Búsquedas Dispersa y de Entorno Variable en Minería de Datos

Miguel García Torres, Belén Melián Batista, José A. Moreno Pérez,
J. Marcos Moreno Vega, Raquel Rivero Martín

Dept. de Estadística, I.O. y Computación

Universidad de La Laguna

38271 La Laguna

{mgarcia,mbmelian,jamoreno,jmmoreno}@ull.es,rrivero@arrakis.es

Resumen

Las Metaheurísticas están alcanzando cada vez más relevancia en la aplicación de técnicas de Aprendizaje Automático en tareas minería de datos para el descubrimiento de conocimiento en grandes bases de datos. Dos de las metaheurísticas recientes que han alcanzado éxitos relevantes en diversos campos de aplicación son la metaheurística de Búsqueda Dispersa o Scatter Search (SS) y la Búsqueda de Entorno Variable o VNS (Variable Neighbourhood Search). En este trabajo presentamos las experiencias realizadas en la aplicación de estas técnicas en comparación con otras más usuales en Aprendizaje automático. Las tareas contempladas en los estudios realizados se incluye el agrupamiento o clustering, la clasificación supervisada, la selección de variables y la selección de instancias.

1. Introducción

El procesamiento inteligente de la información actualmente requiere de herramientas inteligentes de Minería de Datos para extraer la información útil almacenada en grandes bases de datos. Los procedimientos heurísticos están siendo cada vez más importantes en la aplicación de técnicas de Aprendizaje Automático en Minería de Datos [1]. Las metaheurísticas son pautas inteligentes para construir procedimientos heurísticos generales en la resolución de problemas [21]. Los algoritmos genéticos son los métodos más generalmente apli-

cados en este contexto. Sin embargo, otras metaheurísticas más desconocidas están alcanzando éxitos relevantes en diversos campos, entre ellos la minería de datos. Dos de estas metaheurísticas son la Búsqueda Dispersa [20] (*Scatter Search*, SS) y la Búsqueda de Entorno Variable [17] (*Variable Neighbourhood Search*, VNS) que cuentan con elementos específicos para obtener ventaja del conocimiento del problema.

El artículo está organizado del siguiente modo; en la sección siguiente se introducen las metaheurísticas de búsqueda entorno variable (Variable Neighbourhood Search, VNS) y de búsqueda dispersa (Scatter Search, SS). La sección 3 se describen los aspectos principales de las aplicaciones de estas metaheurísticas en el campo de la búsqueda dispersa, considerando los paradigmas del agrupamiento o clustering, la clasificación supervisada, la selección de características y la reducción de datos. En la sección 4 describimos con algo más de detalle la aplicación en la selección de variables para la clasificación supervisada. Finalmente, en la última sección, se exponen las conclusiones.

2. Las Metaheurísticas SS y VNS

Las *metaheurísticas* son estrategias inteligentes para diseñar o mejorar procedimientos heurísticos muy generales con un alto rendimiento [21]. El término metaheurística apareció por primera vez en el artículo sem-

inal sobre búsqueda *Tabú* de Fred Glover en 1986 [13]. A partir de entonces han surgido multitud de propuestas de pautas para diseñar buenos procedimientos para resolver ciertos problemas que, al ampliar su campo de aplicación, han adoptado la denominación de metaheurísticas [2].

Las **propiedades** de las distintas metaheurísticas reflejan el interés práctico y teórico de las mismas. Las propiedades deseables indicarán direcciones a las que dirigir los esfuerzos para contribuir al desarrollo científico e ingenieril del área. Una relación de tales propiedades, algunas de ellas parcialmente contrapuestas, incluir las siguientes:

- *Simple*. La metaheurística debe estar basada en un principio sencillo y claro; fácil de comprender.
- *Precisa*. Los pasos y fases de la metaheurística deben estar formulados en términos concretos.
- *Coherente*. Los elementos de la metaheurística debe deducirse naturalmente de sus principios.
- *Efectiva*. Los algoritmos derivados de la metaheurística deben proporcionar soluciones de muy alta calidad; óptimas o muy cercanas a las óptimas.
- *Eficaz*. La probabilidad de alcanzar soluciones óptimas de casos realistas con la metaheurística debe ser alta.
- *Eficiente*. La metaheurística debe realizar un buen aprovechamiento de recursos computacionales; tiempo de ejecución y espacio de memoria.
- *General*. La metaheurística debe ser utilizable con buen rendimiento en una amplia variedad de problemas.
- *Adaptable*. La metaheurística debe ser capaz de adaptarse a diferentes contextos de aplicación o modificaciones importantes del modelo.

- *Robusta*. El comportamiento de la metaheurística debe ser poco sensible a pequeñas alteraciones del modelo o contexto de aplicación.
- *Interactiva*. La metaheurística debe permitir que el usuario pueda aplicar sus conocimientos para mejorar el rendimiento del procedimiento.
- *Múltiple*. La metaheurística debe suministrar diferentes soluciones alternativas de alta calidad entre las que el usuario pueda elegir.
- *Autónoma*. La metaheurística debe permitir un funcionamiento autónomo, libre de parámetros o que se puedan establecer automáticamente.

Varias de estas propiedades están muy relacionadas y apuntan en la misma dirección por lo que pueden agruparse. La simplicidad, la precisión y la coherencia contribuyen a facilitar su aplicabilidad. La evaluación del rendimiento debe atender tanto a la eficiencia como a la efectividad y eficacia de los procedimientos heurísticos obtenidos. La aplicabilidad de una metaheurística debe estar sustentada en la generalidad, pero también en su adaptabilidad y robustez. Para favorecer la utilidad de la metaheurística en la resolución de problemas reales, por ejemplo incorporándolo a Sistemas de Ayuda a la Decisión, son importantes las propiedades de inactividad multiplicidad y autonomía que propician un interface amigable para distintos campos de aplicación.

Varias de las metaheurísticas más relevantes son estrategias para diseñar procesos de búsqueda heurística basados en movimientos o transformaciones de las alternativas que determinan una estructura de entornos en el espacio de soluciones. La **Búsqueda de Entorno Variable** (*Variable Neighbourhood Search*, VNS) es una metaheurística propuesta recientemente [18], [16], [17] que está basada en un principio simple: cambiar sistemáticamente de estructura de entornos dentro de la búsqueda.

Una *estructura de entornos* en el espacio de soluciones de un problema es una aplicación que asocia a cada solución el entorno de las soluciones *vecinas* que se obtiene por uno de los movimientos considerados. Las metaheurísticas de búsqueda local aplican un movimiento de mejora a la solución de búsqueda utilizando una estructura de entornos. Los entornos \mathcal{N}_k pueden ser inducidos por una o más métricas introducidas en el espacio de soluciones X .

La VNS está basada en tres hechos simples:

1. Un mínimo local con una estructura de entornos no lo es necesariamente con otra.
2. Un mínimo global es mínimo local con todas las posibles estructuras de entornos.
3. Para muchos problemas, los mínimos locales con la misma o distinta estructura de entornos están relativamente cerca.

Una búsqueda local determina una mejor solución del entorno de la solución actual y queda atrapada en un óptimo local con respecto a la estructura de entornos utilizada. La búsqueda de entorno variable descendente (VND) realiza un cambio de estructura de entornos de forma determinística cada vez que se llega a un mínimo local. Los pasos de la VND se muestran en la figura 2.

La solución final proporcionada por el algoritmo es un mínimo local con respecto a todas las k_{max} estructuras de entornos, y por tanto la probabilidad de alcanzar un mínimo global es mayor que usando una sola estructura.

La búsqueda de entorno variable general combina cambios determinísticos y aleatorios de estructura de entornos. Mediante cambios sistemáticos en la estructura de entornos en la búsqueda descendente se alcanzan mínimos locales con respecto a varias estructuras de entornos y mediante cambios sistemáticos en un proceso de agitación se escapa de sus zonas de atracción. Los pasos de una VNS general se muestran en la figura 2.

La condición de parada puede ser, por ejemplo, el máximo tiempo de CPU permitido, el máximo número de iteraciones, o el máximo número de iteraciones entre dos mejoras. Frecuentemente los entornos \mathcal{N}_k sucesivos están

Inicialización

Seleccionar el conjunto de estructuras de entornos $\mathcal{N}_k, k = 1, \dots, k_{max}$, que se usarán en el descenso; encontrar una solución inicial x ;

Iteraciones

Repetir, hasta que no se obtenga mejora, la siguiente secuencia:

- (1) Hacer $k \leftarrow 1$;
 - (2) Repetir, hasta que $k = k_{max}$, los pasos:
 - (a) Exploración del entorno Encontrar la mejor solución x' del k -ésimo entorno de x ($x' \in \mathcal{N}_k(x)$);
 - (b) Moverse o no Si la solución obtenida x' es mejor que x , hacer $x \leftarrow x'$ y $k \leftarrow 1$; en otro caso, hacer $k \leftarrow k+1$.
-

Figura 1: VNS Descendente; VND

anidados. Obsérvese que la solución x' se genera al azar en el paso (2a) para evitar el ciclo, que puede ocurrir si se usa cualquier regla determinística.

La **Búsqueda Dispersa** [19], [20] es una metaheurística basada en un pequeño conjunto de buenas soluciones, el *conjunto de referencia*, que evoluciona mediante la combinación inteligente de sus elementos. Los principios de la búsqueda dispersa fueron introducidos en los años 70 como una extensión de ciertas formulaciones para la combinación de reglas de decisión y restricciones de problemas [14].

Las soluciones del conjunto de referencia *RefSet* debe estar constituido por buenas soluciones distribuidas por todo el espacio de soluciones. Este conjunto es generado atendiendo a criterios de calidad y diversidad. Estos criterios son tanto calidad en función el valor de la función objetivo, lo que permite incidir en la intensificación de la de búsqueda, como de dispersión en función de las diferencias con el resto de soluciones, lo que incide en la diversificación del proceso. Ciertos subconjuntos de *RefSet* proporcionan nuevas soluciones que una vez mejoradas e introducidas en *RefSet* lo hacen evolucionar. La actualización del conjunto de referencia se realiza atendiendo a am-

Inicialización

Seleccionar un conjunto de estructuras de entornos \mathcal{N}_k , para $k = 1, \dots, k_{max}$, que se usarán en la agitación; seleccionar un conjunto de estructuras de entornos \mathcal{N}'_j , para $j = 1, \dots, j_{max}$, que se usarán en el descenso; encontrar una solución inicial x ; elegir una condición de parada;

Iteraciones

Repetir, hasta que se cumpla la condición de parada, la siguiente secuencia:

- (1) Hacer $k \leftarrow 1$;
 - (2) Repetir, hasta que $k = k_{max}$, los pasos:
 - (a) Agitación Generar al azar una solución x' del entorno $\mathcal{N}_k(x)$;
 - (b) Búsqueda local Aplicar la VND con las estructuras de entornos \mathcal{N}'_j , para $j = 1, \dots, j_{max}$; denótese con x'' la solución así obtenida.
 - (c) Moverse o no Si la solución obtenida x'' es mejor que x , hacer $x \leftarrow x''$ y $k \leftarrow 1$; en otro caso, hacer $k \leftarrow k+1$.
-

Figura 2: VNS General; GVNS

Los criterios de forma simultánea lo que permite al procedimiento proporcionar un conjunto de buenas soluciones en vez de una única solución heurística tal como sucede con otros procedimientos metaheurísticos.

El procedimiento de búsqueda dispersa incluye cinco métodos principales: *Método de Generación*, que proporciona soluciones dispersas en todo en el espacio de soluciones; *Método de Mejora*, que mejora una solución dada para obtener un óptimo local; *Método de Selección*, que selecciona los subconjuntos de soluciones del conjunto de referencia que serán combinadas; y *Método de Combinación*, que combina las soluciones de los subconjuntos de soluciones y *Método de Actualización*, que actualiza el conjunto de referencia con el resultado de las combinaciones. Una descripción más detallada de los métodos involucrados en la búsqueda dispersa se proporciona en [19].

La Figura 3 resume una implementación básica de la metaheurística de búsqueda dispersa.

procedimiento Búsqueda Dispersa

begin

Generación de la Población;

Generación del Conjunto de Referencia;

repeat

repeat

Método de Generación de Subconjuntos;

Método de Combinación de Soluciones;

Método de Mejora;

until (*CriterioParada1*);

Método de Actualización del Conjunto de Referencia;

until (*CriterioParada2*);

end.

Figura 3: Búsqueda Dispersa

El algoritmo comienza generando una población de buenas soluciones diversas mediante la ejecución del *Método de Generación* y el *Método de Mejora*. El conjunto de referencia se genera mediante la selección de las b ($b = b_1 + b_2$) mejores soluciones de la población atendiendo tanto al criterio de calidad del valor de la función objetivo como al criterio de diversidad. En primer lugar se seleccionan las b_1 soluciones de la población con mejores valores de la función objetivo. Seguidamente, se seleccionan las b_2 soluciones de la población más dispersas con respecto a las soluciones incluidas en el conjunto de referencia. Una vez construido el conjunto de referencia, se generan diversos subconjuntos de soluciones del mismo que serán utilizados para la ejecución del *Método de Combinación*. Se combinan entonces las soluciones de cada uno de los subconjuntos obtenidos para obtener nuevas soluciones que se mejoran mediante el *Método de Mejora*. Finalmente, el *Método de Actualización del Conjunto de Referencia* actualiza el conjunto de referencia a partir de las soluciones mejoradas.

3. Aplicación de SS y VNS en Minería

Cuatro de los principales paradigmas en aplicaciones del Aprendizaje Automático ([22]) en Minería de Datos son: agrupamiento o clustering, clasificación supervisada, selección de características y reducción de datos. El agrupamiento o clustering es el paradigma principal de aprendizaje no supervisado. El objetivo es encontrar grupos de instancias o casos constituidos por elementos similares. Dado un conjunto de instancias descritas por una serie de características, el problema es encontrar una partición del conjunto completo de instancias en subconjuntos o clases de tal manera que las instancias dentro de una misma clase sean muy similares mientras que las instancias en clases distintas sean disimilares.

Los procedimientos basados en distancia consideran una distancia entre las descripciones para evaluar la similitud y disimilitud entre ellas. Una amplia gama de funciones distancia apropiadas para distintas circunstancias han sido propuestas y consideradas en la literatura (ver, por ejemplo [25]). Entonces una forma usual de establecer las particiones consiste en identificar una selección de ejemplos representativas de las clases (en el caso más simple se elige sólo un ejemplo por clase). Entonces cada nuevo caso se asigna a la clase del ejemplo representativo más cercano. Tanto metaheurística VNS como el Scatter Search han sido aplicados con éxito al problema de la p -mediana que es muy parecido a este problema (ver por ejemplo [12] y [11]). El problema de la p -mediana consiste en elegir p puntos que minimicen la suma de distancias a todos los casos. Una muestra de la extensión de estas técnicas a los problemas de clustering lo encontramos en [4].

En aprendizaje supervisado basado en casos, además de las características que describen los casos, una variable adicional representa la clase que debe ser pronosticada a partir de su descripción. De un conjunto de casos de entrenamiento con clases conocidas se quiere obtener una regla de clasificación para obtener la clase desconocida de un conjunto de in-

stancias test. La metodología basada en distancias también se selecciona un conjunto de instancias representativas del conjunto de entrenamiento y se clasifica los casos test teniendo en cuenta la clase de los casos seleccionados más cercanos. La amplia variedad de distancia entre descripciones puede ser también utilizada para esta tarea. Este tipo de problema también es similar al problema de la p -mediana ya que también se trata de seleccionar un cierto número de elementos con una función a optimizar diferente. Estos pertenecen a la amplia clase de los problemas de selección de para los que la mayoría de los procedimientos heurísticos se basa en movimientos de intercambio. Este tipo de movimientos ha sido utilizado con mucha frecuencia en exitosas aplicaciones de las metaheurísticas VNS y Scatter Search. Se ha aplicado estas metaheurísticas en esta tarea [3].

Sin embargo, no es útil considerar todo el conjunto de características para este y otros paradigmas de clasificación. Por tanto, una tarea importante es la selección de características que trata de a obtención del mejor subconjunto de características para realizar la tarea de clasificación. Una adecuada selección de características tiene, no sólo la ventaja de tomar la información relevante en la descripción de los casos, sino también el evitar la información redundante que harían a los algoritmos y reglas de clasificación mucho más ineficientes. Tanto el Scatter Search cómo la VNS sido ya aplicadas con éxito a este problema ([10], [9], [5] y [6],[8]).

Finalmente, en grandes bases de datos, también la selección de instancias, como un mecanismo para reducir la cantidad de información a tratar sin reducir el rendimiento de los métodos constituye un cuarto paradigma de la minería de datos en las que estas metaheurísticas están siendo aplicadas. Para este problema también se han comenzado a estudiar la aplicación de estas metaheurísticas. [7].

4. Aplicación de SS y VNS en Selección de Instancias

A continuación pasamos a describir los componentes de las aplicaciones de las metaheurísticas SS y VNS para el problema de la selección de Variables para la clasificación supervisada.

Para la población inicial del Scatter Search, se genera mediante un procedimiento constructivo del tipo de los aplicados en la primera fase del GRASP (*Greedy Randomize Adaptive Search Procedure*) ([24],[23]). Para ello se ordenan las variables en función de un vector de pesos asociado a los atributos la evaluación que indican la calidad del atributo para el proceso de clasificación. Se construye la lista restringida de candidatos (LRC) formada por los atributos con mayor peso. La estrategia para la creación de cada solución consiste en escoger, de LRC, una variable al azar y asignarla a la solución. Si mejora, se actualiza el LRC introduciendo la siguiente mejor variable. En caso contrario se finaliza el proceso de generación de la solución.

El conjunto de referencia se forma por soluciones introducidas en función de la intensidad y de la diversidad. De cara a la intensidad contendrá aquellas soluciones que tengan la mejor evaluación. Respecto a la diversidad se incluirán aquellas soluciones que tengan mayor número de variables distintas de las ya presentes en el conjunto de referencia.

Se seleccionan todos los subconjuntos de dos elementos del conjunto de referencia para combinación. En la primera iteración se aplica el método de combinación a todos los subconjuntos de dos elementos posibles; sin embargo en iteraciones posteriores sólo se llevan a cabo las combinaciones no contempladas previamente. El procedimiento de combinación intenta combinar las buenas propiedades de las soluciones. Tiene como finalidad la obtención de buenas soluciones diversas respecto a las soluciones ya consideradas. Se consideraron dos métodos de combinación que utilizan estrategias greedy: la combinación greedy (GC) y a la combinación greedy reducida (RGC). Ambas comienzan añadiendo a las nuevas soluciones los atributos comunes, y en cada iteración

se añade la variable de las no comunes que mejore, en mayor medida, alguna nueva solución. La diferencia entre ambas radica en el subconjunto de atributos no comunes considerados, que en la versión reducida sólo se consideran aquellos que aparecen en las mejores soluciones durante el proceso de búsqueda. Para ello hace uso del porcentaje de aciertos asociados con cada solución obtenida a lo largo del procedimiento de búsqueda.

El *método de mejora* se aplica a todas las soluciones surgidas de la combinación. El conjunto de atributos que no pertenezcan a la solución se ordenan en función de los pesos anteriores para su introducción en la solución. Todas las soluciones resultantes de aplicar el método de mejora se van almacenando en un conjunto que se usa para actualizar el conjunto de referencia de forma estática.

El conjunto de referencia es actualizado de acuerdo a la intensidad y diversidad. Para ello seleccionamos los la mitad de sus elementos de las mejores soluciones de la unión entre el propio conjunto de referencia y las del conjunto anterior y a continuación las restantes soluciones más diversas tal y como se explicó anteriormente.

La versión de la **Búsqueda por Entorno Variable** (VNS) adaptada al problema de la selección de atributos es la VNS Básica ([17]). En primer lugar hay que establecer la estructura de la entornos. La k -ésima vecindad de una solución consiste en todas las soluciones que pueden ser alcanzadas mediante el cambio de k atributos en la solución con k atributos no pertenecientes a ella. La distancia entre soluciones se establece por el cardinal de la diferencia simétrica y entonces la k -ésima vecindad es el conjunto de soluciones a distancia igual o inferior a k .

Los elementos a adaptar del procedimiento del VNS fueron la inicialización, la fase de agitación y la búsqueda local. Para la generación de la solución inicial se combinaron las estrategias SFS y SBE ejecutándolas una detrás de la otra hasta que no haya mejora. Se consideró otra estrategia consistente en generar al azar la solución inicial con un número de atributos igual el máximo entre el número

de atributos del SFS-SBE y el 10% del total de atributos del problema a considerar. En la etapa de agitación una solución agitada es generada al azar desde la k -ésima vecindad de la solución actual. Además, en cada iteración se consideran $2k$ atributos tabú; de esta forma la condición de parada se alcanza cuando el número de atributos para intercambiar, k , es mayor que el mínimo entre el número de atributos de la solución y de atributos no pertenecientes a la solución. La búsqueda local aplicada en nuestro procedimiento consiste en ejecutar, mientras no haya mejora, las estrategias SFS y SBE. El óptimo local alcanzado se compara con la mejor solución encontrada hasta el momento por la estructura k -ésima. Si dicho óptimo local mejora la mejor solución se reinician las estructuras de entorno ($k = 1$), en caso contrario, se pasa a la siguiente $k = k + 1$.

5. Conclusiones

La función de distancia, que esta presente con un significado relevante en estos problemas, juega un papel importante en las aplicaciones de las metaheurísticas VNS y Scatter Search. La función de distancia permite crear las estructuras de entorno anidadas para el VNS. La función de distancia permite modular la diversificación y la intensificación en el Scatter Search. Dada una función de distancia entre los elementos de los conjuntos que constituyen las soluciones (instancias o características), la distancia entre dos soluciones se puede definir como la mínima, máxima o la suma de las distancias de los elementos de un conjunto al otro conjunto. De esta forma se puede definir la distancia entre una solución y un conjunto de soluciones para modular la diversidad de todo el conjunto de referencia.

Referencias

- [1] H.A. Abbass, C.S. Newton, R. Sarker. *Data Mining: A heuristic Approach*. Idea Group (2002).
- [2] Brito Santana, J., Campos Rodríguez, C., García López, F., García Torres M., Melián Batista, B., Moreno Pérez, J.A. and Moreno Vega, J.M. *Metaheurísticas: Una revisión actualizada*. Documentos de Trabajo del DEIOC. Universidad de La Laguna, Número 2/2004.
- [3] M. García Torres, J.A. Moreno Pérez, J. Marcos Moreno Vega. "VNS para Clasificación supervisada". Gijón. 5-7 de Febrero de 2003. MAEB 2003,
- [4] M. García Torres, José A. Moreno Pérez, J. Marcos Moreno Vega. Comparación de estrategias de arranque múltiple para clasificación no supervisada."Sevilla, España. Noviembre de 2002 IBERAMIA 2002.
- [5] M. García-Torres, Félix García-López, Belén Melián-Batista, José A. Moreno-Pérez, J. Marcos Moreno-Vega. "Solving Feature Subset Selection Problem by a Hybrid Metaheuristic."HM 2004. First International Workshop in Hybrid Metaheuristics at ECCAI 2004 Valencia, 22-23 de Agosto de 2004.
- [6] Félix García López, M. García Torres, Belén Melián Batista, José A. Moreno Pérez, J. Marcos Moreno Vega. "Búsqueda Dispersa y Algoritmo Genético para el problema de la selección de Variables". Córdoba, . 4-6 de Febrero de 2004. Tercer Congreso Español de Metaheurísticas, Algoritmos Evolutivos y Bioinspirados, MAEB 2004, Pág. 609-616. ISBN: 84-688-4224-9
- [7] M. García Torres, Belén Melián Batista, José Andrés Moreno Pérez, Raquel Rivero Martín, J. Marcos Moreno Vega. "Búsqueda Dispersa en Selección de Instancias". MAEB 2005. Granada, 15-17 Septiembre 2005.
- [8] M. García Torres, B. Melián Batista, J.A. Moreno Pérez and J.M. Moreno Vega. "Variable Neighbourhood Tabu Search for Feature Selection". Working paper submitted to MIC 2005.

- [9] García López, F., García Torres M., Moreno Pérez, J.A. and Moreno Vega, J.M. *Scatter Search for the Future Selection Problem*. Lecture Notes in Computer Science, Vol. 3040 pp. 517-525 (2004)
- [10] F. García López, M. García Torres, B. Melián Batista, J.A. Moreno Pérez and J.M. Moreno Vega. "Solving Feature Subset Selection Problem by a Parallel Scatter Search", *European Journal of Operational Research*, 2005, to appear.
- [11] F. García López, B. Melián Batista, J.A. Moreno Pérez and J.M. Moreno Vega. *The parallel variable neighbourhood search for the p-median problem*. Journal of Heuristics, 8 pp. 377-390 (2002).
- [12] F. García López, B. Melián Batista, J.A. Moreno Pérez and J.M. Moreno Vega. Parallelization of the Scatter Search for the p -median problem, *Parallel Computing*, 29 (2003) 575-589.
- [13] Glover, F., "Future Paths for Integer Programming and Links to Artificial Intelligence. Computers and Operations Research. 5, (1986) 533-549.
- [14] Glover, F., Heuristics for Integer Programming using Surrogate Constraints, *Decision Sciences* 8, (1977) 156-166.
- [15] Glover, F., Laguna, M., Martí, R. Fundamentals of Scatter Search and Path Relinking *Control and Cybernetics*, 39, (2000) 653-684.
- [16] P. Hansen, N. Mladenovic. (2003) Variable neighbourhood search. In Fred Glover and Gary A. Kochenberger (eds.) *Handbook of Metaheuristics*, chapter 6, Kluwer, 2003.
- [17] P. Hansen, N. Mladenovic., J.A. Moreno Pérez (2003) *Búsqueda de Entorno Variable* Inteligencia Artificial. Revista Iberoamericana de Inteligencia Artificial. Numero 19, Volumen 2, páginas 77-92.
- [18] N. Mladenović, P. Hansen (1997) Variable neighborhood search. *Computers & Operations Research*, 24:1097-1100.
- [19] Laguna, M. and R. Martí, *Scatter Search: Methodology and Implementations in C*, Kluwer Academic Press, (2003).
- [20] R. Martí, Laguna, M. *Scatter Search: diseño básico y estrategias avanzadas*, Inteligencia Artificial. Revista Iberoamericana de Inteligencia Artificial. Numero 19, Volumen 2, (2003) páginas 123-130.
- [21] B. Melián, J.A. Moreno Perez, J.M. Moreno-Vega, J. páginas (2003) *Metaheurísticas: Una visión global* Inteligencia Artificial. Revista Iberoamericana de Inteligencia Artificial. Numero 19, Volumen 2, páginas 7-28.
- [22] Mitchell, T. *Machine Learning*, Series in Computer Science, McGraw-Hill, (1997).
- [23] M. Resende, C. Ribeiro. (2003) *Greedy Randomized Adaptive Search Procedures* In F. Glover and G. Kochenberger, editors, *Handbook on MetaHeuristics*, chapter 8. 2003.
- [24] Resende, G.C.M., González Velarde, J.L. (2003) *GRASP: Procedimientos de búsquedas miopes aleatorizados y adaptativos* Inteligencia Artificial. Revista Iberoamericana de Inteligencia Artificial. Numero 19, Volumen 2, páginas 61-76
- [25] D. R. Wilson, T. R. Matinez, Improved heterogeneous distance functions, *Journal of Artificial Intelligence Research* 6 (1997) 1-34.