

Búsqueda por Entornos Variables para Planificación Logística*

José Andrés Moreno Pérez
DEIOC. Instituto Universitario de Desarrollo Regional
Universidad de La Laguna,
38271 La Laguna, España
jamoreno@ull.es

Nenad Mladenović
School of Mathematics. Brunel University.
London, Reino Unido
nenad.mladenovic@brunel.ac.uk

Resumen

La Búsqueda por Entornos Variables (Variable Neighbourhood Search, VNS) es una metaheurística reciente para resolver problemas de optimización cuya idea básica es el cambio sistemático de entorno dentro de una búsqueda local. En este artículo presentamos las reglas básicas de la VNS y sus extensiones para la resolución heurística de una variedad de problemas de optimización. Se ofrece un breve recorrido por los aspectos más relevantes de la aplicación de esta metaheurística en problemas de planificación logística. Finalmente se incluyen algunas reflexiones sobre aspectos esenciales de las metaheurísticas y del análisis de los procesos de solución heurística, y la contribución de los trabajos con la VNS para estas cuestiones.

1. Introducción

Las Metaheurísticas son estrategias generales para diseñar procedimientos heurísticos para resolver un problema de optimización mediante un proceso de búsqueda en un cierto espacio de soluciones alternativas. Los procesos de búsqueda heurística están generalmente basados en transformaciones de las alternativas que determinan una estructura de entornos en el espacio de soluciones. La *Búsqueda por Entornos Variables* (*Variable Neighbourhood Search* (VNS)), está basada en un principio simple: cambiar sistemáticamente la estructura de entornos por la que se realiza la búsqueda. Su desarrollo ha sido rápido, con muchos artículos ya publicados o pendientes de aparecer. Se han realizado muchas extensiones, principalmente para permitir la solución de problemas de gran tamaño. En la mayoría de ellas, se ha hecho un esfuerzo por mantener la simplicidad del esquema básico.

En la siguiente sección se exponen las reglas básicas de la VNS. Las extensiones, incluyendo los híbridos, se consideran en la sección 3. En la sección 4 se repasan las aplicaciones prácticas más importantes de la VNS en planificación logística. En la sección 5 se profundiza en algunas cuestiones relevantes de esta metaheurística. El trabajo finaliza con unas breves conclusiones.

2. Esquemas Fundamentales

Un problema de optimización consiste en encontrar, dentro de un conjunto X de soluciones factibles, la que optimiza una función $f(x)$. Si el problema es de minimización se formula como sigue:

$$\min \{ f(x) \mid x \in X \} \quad (1)$$

donde x representa una *solución* alternativa, f es la *función objetivo* y X es el *espacio de soluciones* factibles del problema. Una *solución óptima* x^* (o mínimo global) del problema es una solución factible donde se alcanza el mínimo de (1).

* Este trabajo ha sido parcialmente financiado por los proyectos TIN2005-08404-C04-03 (el 70% son fondos FEDER) y PI042005/044

Si x es un conjunto finito pero de gran tamaño es un problema de *optimización combinatoria*. Si $X = \mathcal{R}^n$, hablamos de *optimización continua*. La mayoría de los problemas de optimización que surgen en aplicaciones prácticas son *NP-duros* y para abordarlos se necesitan métodos de optimización heurística (al menos para instancias de gran tamaño o como solución inicial para algún procedimiento exacto). Las metaheurísticas se han mostrado como una herramienta apropiada para abordar este tipo de tareas.

Una *estructura de entornos* en el espacio de soluciones X es una aplicación $N: X \rightarrow 2^X$ que asocia a cada solución $x \in X$ un entorno de soluciones $N(x) \subseteq X$, que se dicen *vecinas* de x . Las metaheurísticas de búsqueda local aplican una transformación o movimiento a la solución de búsqueda y por tanto utilizan, explícita o implícitamente, una estructura de entornos. El entorno de una solución $x \in X$ estaría constituido por todas aquellas soluciones que se pueden obtener desde x mediante una de las transformaciones o movimientos contemplados.

Una solución factible $x^* \in X$ es un *mínimo global* del problema (1) si no existe una solución $x \in X$ tal que $f(x) < f(x^*)$. Decimos que la solución $x' \in X$ es un *mínimo local* con respecto a N si no existe una solución $x \in N(x') \subseteq X$ tal que $f(x) < f(x')$. Una búsqueda local descendente cambia la solución actual por otra solución mejor de su entorno, por tanto corren el riesgo de quedarse atascada en un mínimo local que no sea óptimo global. Las metaheurísticas basadas en procedimientos de búsqueda local aplican distintas formas de continuar la búsqueda después de encontrar el primer óptimo local. La metaheurística VNS consiste básicamente en cambiar la estructura de entornos de forma sistemática.

La metaheurística VNS se basa en aprovechar sistemáticamente 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 están relativamente próximos entre sí.

Los dos primeros hechos sugieren el empleo de varias estructuras de entornos en las búsquedas locales para abordar un problema de optimización. El último hecho, constatado empíricamente, indica que los óptimos locales proporcionan información acerca del óptimo global. Puede ser, por ejemplo, que tengan características comunes pero, generalmente, no se sabe cuales son esas características. Es conveniente, por tanto, realizar un análisis de las proximidades de cualquier óptimo local buscando información que permita orientar la búsqueda hacia el óptimo global.

Las heurísticas basadas en VNS, al contrario de lo que ocurre con otras metaheurísticas, se mantienen simples; no sólo sus esquemas básicos sino también la mayoría de las extensiones, requiriendo el ajuste de muy pocos parámetros. Esta característica permite que la metaheurística VNS y sus extensiones sean útiles para diseñar rápidamente procedimientos heurísticos con los que proporcionar buenas soluciones con rapidez de manera muy simple dejando al descubierto cuales son las razones que determinan su rendimiento, lo que frecuentemente facilita la elaboración de implementaciones sofisticadas muy eficientes.

Al abordar la resolución de un problema, o tipo de problemas, particular, una de las primeras tareas consiste en la recopilación de información y documentación acerca del problema y de sus características. Esta información debe incluir aspectos referentes a, aspectos de los problemas reales, su dificultad y los procedimientos, heurísticos o no, aplicados al enfrentarse a ellos. Esta información puede completarse con algunos diseños intuitivos de operaciones de mejora de las soluciones propuestas o información aportada por la experiencia del rendimiento de tales operaciones. Esta información y el conocimiento de procedimientos rudimentarios, generalmente búsquedas locales, puede ser aprovechada para el diseño de ingredientes de la VNS que pueden ser determinantes en su éxito.

En los problemas de logística más importantes, las soluciones están constituidas por selecciones u ordenaciones de elementos. Una forma corriente de realizar movimientos en el espacio de soluciones es cambiar algunos de estos elementos. Estos movimientos dan lugar a diversas estructuras de entornos si se fija algunos de los aspectos de estos cambios. En muchos procedimientos de búsqueda se utilizan estos movimientos fijando o acotando el número de elementos de la solución que se pueden cambiar. De esta

forma se obtienen los entornos más utilizados las búsquedas de entornos variables. Un k -intercambio es un movimiento consistente en intercambiar k elementos de la solución. Es decir, cambiar k elementos de la solución por otros k elementos que no estén en la solución, si las soluciones consisten en la selección de un número fijo de elementos, o intercambiar de posición k elementos de la solución, si se trata de ordenaciones. Se suelen utilizar los entornos $N_k(x)$ consistentes en cambiar k elementos en la solución x , para $k = 1, 2, \dots$. En otras aplicaciones el k -ésimo entorno consiste en las soluciones que se obtienen al cambiar, a lo sumo, k elementos de la solución. Este es el tipo de entornos más corriente en las aplicaciones de la VNS a problemas de logísticos estándares como el problema de la p -mediana. Para los problemas de rutas de vehículos, como el problema del vendedor o TSP (Travelling Salesman Problem) también se utilizan otros movimientos estándares en las heurísticas de estos problemas.

2.1. VNS Descendente

Una búsqueda local descendente consiste básicamente en determinar iterativamente una mejor solución a partir de la solución actual mediante alguna transformación o movimiento. La clásica búsqueda descendente *greedy* o *voraz* consiste en reemplazar siempre la solución actual por la mejor de todas las soluciones que se pueden obtener a partir de la actual mediante uno de los movimientos contemplados. En el extremo opuesto a la estrategia voraz, de “el mejor primero”, se encuentra la estrategia **ansiosa**, de “el primero mejor”, que aplica un movimiento de mejora desde que se detecte alguno. Otras estrategias intermedias son posibles para elegir esta solución del entorno que mejora la solución actual, pero todas ellas deben detenerse cuando no sea posible encontrar dicha mejora. Seguramente la elección de los movimientos a considerar puede ser determinante en el éxito de la búsqueda local, pero difícilmente se puede determinar a priori cual de las posibilidades vislumbradas va a ser la más efectiva. Frente a las estrategias de probarlas una de tras de otra en cada caso o emplearlas conjuntamente contemplando todos los movimientos posibles, que son estrategias poco inteligentes, el hecho reseñado anteriormente indica la posible conveniencia de combinar los distintos tipos de movimientos.

Si en una búsqueda local descendente se realiza un cambio de estructura de entornos cada vez que se llega a un mínimo local, se obtiene la **Búsqueda por Entornos Variables Descendente** (*Variable Neighbourhood Descent*, VND). Denotemos por N_k , $k = 1, \dots, k_{max}$, a una serie finita de estructuras de entornos en el espacio X . Los entornos N_k pueden ser inducidos por una o más métricas introducidas en el espacio de soluciones x .

Los pasos de la VND se muestran en la figura 1.

Inicialización.

Seleccionar una serie de estructuras de entornos N_k , $k = 1, \dots, k_{max}$, y 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 .

(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

Nótese que, dado que se vuelve a iniciar el recorrido de las estructuras de entornos desde la primera cada vez que se produce un cambio, cuando se trata de encontrar la mejor solución del k -ésimo entorno de la solución actual x , esta solución no ha podido ser mejorada con ninguno de los $(k-1)$ entornos anteriores. Por tanto, la solución final proporcionada por el algoritmo es un mínimo local con respecto a cada una de las k_{max}

estructuras de entornos. Como consecuencia de ellos, la probabilidad de alcanzar un mínimo global es mayor que usando una sola estructura.

En la selección de la estructura de entornos para abordar un determinado problema, tanto implementando una búsqueda local, voraz o ansiosa, o cualquier otra metaheurística de búsqueda que utilice los entornos de forma explícita o implícita deben tenerse en cuenta diversas cuestiones sobre los entornos. Entre tales cuestiones está la completitud, complejidad y número de los movimientos a aplicar, la eficiencia de los mismos, la posibilidad de examinarlos en distinto orden, e incluso el grado de precisión. Estos aspectos son también cruciales para el uso de varias estructuras de entornos en una VNS.

La mayoría de las heurísticas de búsqueda local usan en sus descensos simplemente un entorno y algunas veces dos ($k_{max} \leq 2$). Además del orden *secuencial* de las estructuras de entornos en la VND anterior, se puede desarrollar una estrategia *anidada*. Supongamos, por ejemplo, que $k_{max} = 3$. Entonces una posible estrategia anidada es: ejecutar la VND de la figura 1 para las dos primeras estructuras de entornos, sobre cada x' que pertenezca al tercer entorno de x ($x' \in N_3(x)$). Esta VNS ha sido aplicada en , y .

2.2 VNS Reducida

Aparte de la búsqueda local descendente, otro método de búsqueda basado explícitamente o implícitamente en una estructura de entornos y aún más rudimentario pero de uso muy frecuente es la búsqueda por recorrido al azar. Mientras que la búsqueda local concentra su esfuerzo en la explotación o intensificación de la búsqueda entorno a la solución actual, la búsqueda por recorrido al azar, por el contrario, persigue una mayor capacidad de exploración o diversificación de la búsqueda. El procedimiento consiste en realizar transformaciones de la solución actual hasta encontrar una mejora; cuando esto ocurre, se toma la nueva solución como solución actual y se continúa la búsqueda desde ella. En la elección de estas transformaciones o movimientos es conveniente tener en cuenta también una serie de aspectos: en qué dirección realizar los movimientos, qué lejanía abarcar en las transformaciones. Si se repiten los fracasos en la búsqueda de una solución mejor, se puede probar otra forma distinta de realizar las transformaciones. La VNS reducida realiza estos cambios de forma sistemática.

La **Búsqueda por Entornos Variables Reducida** (*Reduced Variable Neighbourhood Search*, RVNS) selecciona al azar soluciones del entorno actual de la solución actual cambiando a la siguiente estructura de entornos si no se obtiene mejora y volviendo a la primera estructura en otro caso. Los pasos de la RVNS se presentan en la figura 2.

Inicialización.

Seleccionar una serie de estructuras de entornos N_k , $k = 1, \dots, k_{max}$, y 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 k -ésimo entorno de x ($x' \in N_k(x)$).

(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 2. VNS Reducida; RVNS.

El cambio de estructura de entornos se puede realizar sólo si la mejora no se produce en cierto número de intentos. En este caso, se considera que varias estructuras de entornos consecutivas coinciden; $N_k = N_{k+1}$, para algunos valores de k . Frecuentemente, la distinta forma de realizar los movimientos que de lugar a las

distintas estructuras de entornos puede significar probar soluciones más distantes o más distintas. En estos casos se está usando una serie de estructuras de entornos *anidadas*, es decir donde $N_k(x) \subseteq N_{k+1}(x)$, y por tanto el cambio de estructura de entornos se interpreta como una ampliación del entorno o del radio de búsqueda de la mejora. La ampliación del radio se realiza por escalones si $N_k(x) = N_{k+1}(x)$ para una serie de iteraciones consecutivas.

La RVNS es útil para instancias muy grandes de problemas en las que la búsqueda local es muy costosa. Como condición de parada se usa generalmente el máximo número de iteraciones entre dos mejoras. La RVNS se ha mostrado superior a un método de Monte-Carlo en un problema minimax continuo y a la heurística del *intercambio rápido* de Whittaker al aplicarla al problema de la p -mediana .

2.3. VNS Básica

La **Búsqueda por Entornos Variables Básica** (*Basic Variable Neighbourhood Search*, BVNS) es una estrategia que alterna búsquedas locales con movimientos aleatorios sobre unas estructuras de entornos que varían de forma sistemática.

Los pasos de la VNS básica se dan en la figura 3.

Inicialización.

Seleccionar una serie de estructuras de entornos N_k , $k = 1, \dots, k_{max}$, y 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 k -ésimo entorno de x .
 - (b) Búsqueda local. Aplicar algún método de búsqueda local con x' como solución inicial; denótese con x'' el mínimo local así obtenido.
 - (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 3. VNS Básica; BVNS.

Las estructuras de entornos utilizadas pueden ser anidadas y escalonadas como en la búsqueda por entornos variables reducida. 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. Obsérvese que la solución x' se genera al azar en el paso (2a) para evitar el ciclado prematuro, que puede ocurrir si se usa cualquier regla determinística.

2.4. VNS General

La **Búsqueda por Entornos Variables General** (*General Variable Neighbourhood Search*, GVNS) se obtiene al sustituir la búsqueda local del paso (2b) de la Búsqueda por Entornos Variables básica por una Búsqueda por Entornos Variables descendente; es decir, una VND. Esta estrategia utiliza dos series de estructuras de entornos posiblemente distintas, una para la búsqueda descendente y otra para los movimientos aleatorios de agitación.

Los pasos de la VNS general (GVNS) se muestran en la figura 4.

Inicialización.

Seleccionar una serie de estructuras de entornos $N_k, k = 1, \dots, k_{max}$, que se usarán en la agitación; y una serie de estructuras de entornos $N'_j, j = 1, \dots, j_{max}$, que se usarán en el descenso y 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 k -ésimo entorno de x .
 - (b) Búsqueda local. Aplicar la VND con los entornos $N'_j, j = 1, \dots, j_{max}$, y x' como solución inicial; 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 4. VNS General; GVNS.

El uso de la Búsqueda por Entornos Variables general (GVNS) ha dado lugar a las aplicaciones más exitosas aparecidas recientemente (ver, por ejemplo, ,, , , , y).

2.5. VNS Anidada

Es el caso más sencillo del uso de una VNS general. Ambas series de entorno coinciden en una serie anidada obtenida a partir de un único tipo de movimiento sencillo. El movimiento más usual en este sentido es el del intercambio que consiste en cambiar un elemento de la solución por otro elemento posible. Si las soluciones se identifican como vectores el cambio consiste en modificar una de las componentes; si se trata de la selección de un número determinado de elementos de un universo, el cambio consiste en sustituir un elemento de la solución por otro que no esté en la solución; si se trata de permutaciones, se intercambian las posiciones de dos elementos. El valor de k corresponde con el número de elementos de la solución cambiados (esta ha sido la versión paralelizada en).

La estrategia de **Búsquedas por Entornos Variables Anidados** (*Nested Variable Neighbourhood Search*, NVNS) se obtiene a partir de un único movimiento elemental al que está asociada una estructura de entornos N . Los entornos anidados se definen recursivamente por $N_1(x) = N(x)$ y $N_{k+1}(x) = N(N_k(x)), \forall x \in X$. El número máximo de entornos a usar en la agitación y en la búsqueda local se fijan por los valores k_{max} y j_{max} , respectivamente.

Los pasos de la VNS anidada (NVNS) se muestran en la figura 5.

Inicialización.

Seleccionar una estructura de entornos N y 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 k -ésimo entorno de x .

(b) Búsqueda local con VND.

(i) Hacer $j \leftarrow 1$.

(ii) Repetir, hasta que $j = j_{max}$, los pasos:

- Exploración del entorno. Encontrar la mejor solución $x'' \in N_j(x')$.

- Moverse o no. Si $f(x'') < f(x')$, hacer $x' \leftarrow x''$ y $j \leftarrow 1$; en otro caso, hacer $j \leftarrow j + 1$.

(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 5. VNS anidada; NVNS.

3. Extensiones de la VNS

Se han propuesto en la literatura diversas formas de extender la VNS para dotarla de algunas características adicionales. Describimos en primer lugar varias formas sencillas de realizar estas extensiones en la subsección 3.1. Las tres siguientes subsecciones se dedican a otras tantas extensiones que constituyen mejoras prácticas de la VNS que han permitido resolver con éxito problemas muy grandes: la Búsqueda por Entornos Variables con descomposición (VNDS), la Búsqueda por Entornos Variables sesgada (SVNS) y la Búsqueda por Entornos Variables paralela (PVNS). En la subsección 3.5 se consideran las extensiones por hibridación de la VNS. En algunos casos se han propuesto independientemente procedimientos heurísticos que en esencia explotan ideas del VNS y que pueden considerarse como casos específicos o extensiones de esta metaheurísticas como la búsqueda local iterada (Iterated Local Search ILS) o la búsqueda por entornos grandes (Large Neighbourhood Search, LNS).

3.1. Extensiones básicas

Las primeras extensiones se derivan directamente de la VNS básica. La BVNS es un método descendente de la primera mejora con aleatorización. Sin mucho esfuerzo adicional se transforma en un método ascendente-descendente: en el paso (2c) hacer también $x \leftarrow x''$ con alguna probabilidad, incluso si la solución es peor que la actual (o que la mejor solución encontrada hasta el momento). La búsqueda en los entornos se intensifica si en lugar de una agitación para tomar una única solución al azar se toman en el paso (2a) todas las soluciones del k -ésimo entorno de la solución actual. En este caso es conveniente explotar un posible recorrido sistemático del entorno. También se puede transformar en una búsqueda de la mejor mejora si se realiza la búsqueda local del paso (2b) desde una solución de cada uno de los k_{max} entornos y se aplica el cambio al mejor entorno k^* . Una estrategia distinta con intensificación intermedia se tiene eligiendo la solución x' en el paso (2a) como la mejor entre b (un parámetro) soluciones generadas aleatoriamente en el k -ésimo entorno. Una última extensión sencilla consiste en introducir k_{min} y k_{paso} , dos parámetros que controlan el proceso de cambio de entorno: en el algoritmo anterior, en vez de $k \leftarrow 1$ hacer $k \leftarrow k_{min}$ y en vez de $k \leftarrow k + 1$ hacer $k \leftarrow k + k_{paso}$. Estas extensiones pueden aplicarse a la vez produciendo nuevas variantes de la búsqueda por entornos variables.

3.2. VNS con Descomposición

La **Búsqueda por Entornos Variables con Descomposición** (*Variable Neighbourhood Decomposition Search*, VNDS) extiende la VNS en un esquema de entornos variables en dos niveles basado en la descomposición del problema. Cuando se obtiene al azar la solución agitada, se fijan los atributos comunes entre la solución actual y la agitada y se aborda el problema de optimizar los elementos no comunes, mediante un procedimiento exacto u otra heurística. Sus pasos son los presentados en la figura 5.

Inicialización.

Seleccionar una serie de estructuras de entornos N_k , $k = 1, \dots, k_{max}$, y 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 k -ésimo entorno de x . Definir y como el conjunto de atributos presentes en x' pero no en x ($y = x' \setminus x$).
 - (b) Búsqueda local. Buscar el óptimo parcial en el espacio de los atributos y por alguna heurística. Sea y' la mejor solución encontrada para estos atributos y sea x'' la correspondiente solución del espacio completo obtenida al incorporar a x tales atributos ($x'' = x \setminus x' \cup y'$).
 - (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 6. VNS por Descomposición; VNDS.

La diferencia entre la VNS básica y la VNDS está en el paso (2b): en vez de aplicar algún método de búsqueda local en el espacio completo x (empezando desde $x' \in N_k(x)$), en la VNDS se resuelve en cada iteración un subproblema en un subespacio $X_k \subseteq N_k(x)$ con $x' \in X_k$. Cuando la búsqueda local utilizada en este paso es también una VNS, aparece un esquema VNS en dos niveles.

3.3. VNS Sesgada

Una vez que se ha alcanzado la mejor solución en una gran región es necesario buscar estrategias para alejarse bastante de ella para posibilitar una nueva mejora. La **Búsqueda por Entornos Variables Sesgada** (*Skewed Variable Neighbourhood Search*, SVNS) afronta la exploración de regiones de valles alejados de la solución actual. Si el óptimo local alcanzado está considerablemente alejado del anterior y no es significativamente peor puede ser conveniente llevar hasta allí el proceso de búsqueda. Las soluciones generadas al azar en entornos muy lejanos pueden diferenciarse substancialmente de la solución actual; por lo que la VNS degenera, en algún sentido, en una heurística de arranque múltiple (en la que se realizan iterativamente descensos desde soluciones generadas al azar). Por tanto, la VNS sesgada incorpora una compensación por la distancia desde la solución actual para evitar este inconveniente. Sus pasos son presentados en la figura 7.

Inicialización.

Seleccionar una serie de estructuras de entornos N_k , $k = 1, \dots, k_{max}$. Encontrar una solución inicial x y su valor objetiva $f(x)$; hacer $x^* \leftarrow x$ y $f^* \leftarrow f(x)$. Elegir una condición de parada y un parámetro α .

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 k -ésimo entorno de x .

(b) Búsqueda local. Aplicar algún método de búsqueda local con x' como solución inicial; denótese con x'' el mínimo local así obtenido.

(c) Mejora o no. Si $f(x'') < f^*$, hacer $x^* \leftarrow x''$ y $f^* \leftarrow f(x'')$.

(d) Moverse o no. Si $f(x'') - \alpha \cdot \rho(x, x'') < f(x)$, hacer $x \leftarrow x''$ y $k \leftarrow 1$; en otro caso, hacer $k \leftarrow k + 1$.

Figura 7. VNS sesgada; SVNS.

La SVNS usa una función $\rho(x, x'')$ para medir la distancia entre la solución actual x y el óptimo local encontrado x'' . La distancia usada para definir los entornos N_k puede también utilizarse con este propósito. La elección del parámetro α debe permitir la exploración de valles lejanos a x cuando $f(x'')$ es algo peor que $f(x)$, pero no demasiado (en otro caso siempre se abandonaría la solución x). Con ello se pretende evitar frecuentes trayectorias desde x a una solución relativamente cercana para volver a x . Un buen valor para α tiene que determinarse experimentalmente en cada caso.

3.4. VNS Paralela

Las *Búsquedas por Entornos Variables Paralelas* (*Parallel Variable Neighbourhood Search*, PVNS) constituyen la tercera extensión. Se han propuesto en y diversas formas de paralelizar la VNS que han sido aplicadas al problema de la p -mediana. En se analizan tres de ellas: (i) paralelizar la búsqueda local, (ii) aumentar el número de soluciones obtenidas del entorno actual y realizar búsquedas locales en paralelo desde cada una de ellas y (iii) hacer lo mismo que en (ii) pero actualizando la información sobre la mejor solución encontrada. En todas ellas se utiliza la conocida estrategia maestro-esclavo en la que el procesador maestro, distribuye la información a los procesadores esclavos que una vez realizada su tarea la devuelven al procesador maestro.

En la primera estrategia de paralelización, los procesadores disponibles comparten la tarea de realizar cada búsqueda local. Por ejemplo, en una búsqueda local *greedy* basada en una estructura de entorno, cada procesador busca la mejor solución de una parte del entorno y se selecciona la mejor de ellas. Cuando ninguno de los procesadores consigue mejorar a la solución, la búsqueda local se detiene. En la segunda paralelización, cada procesador toma una solución del entorno de la solución actual y aplica la búsqueda local. Se toma el mejor mínimo local de los encontrados por los procesadores y se compara con la solución actual a la que reemplaza si la mejora. Sin embargo, en la tercera paralelización cada vez que un procesador se detiene en un mínimo local se compara la solución actual y la reemplaza si la mejora. Entonces, mientras un procesador realiza la búsqueda local es posible que otro procesador haya mejorado la solución actual y con esta solución con la que se compara el mínimo encontrado. Se utiliza la conocida estrategia maestro-esclavo en la que el procesador maestro, que almacena la solución actual x , envía a los procesadores soluciones x'_p del entorno de la solución actual x , recibe los mínimos locales x''_p encontrados por éstos que si mejoran la solución actual x en el momento de recibirlos. La estructura de entornos se modifica cuando ninguno de los procesadores ha conseguido mejorar la solución actual. El inconveniente de esta estrategia es

que los procesadores quedan ociosos más tiempo que en la anterior. La segunda paralelización, cuyos pasos se muestran en la figura 8, es la que ha dado mejores resultados.

Inicialización.

Seleccionar una serie de estructuras de entornos N_k , $k = 1, \dots, k_{max}$, y una solución inicial x . Elegir una condición de parada

Iteraciones.

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

(3) Hacer $k \leftarrow 1$.

(4) Repetir en paralelo, hasta que $k = k_{max}$, para cada procesador p los pasos:

(a) Agitación. Generar al azar una solución x'_p del k -ésimo entorno de x .

(b) Búsqueda local. Aplicar algún método de búsqueda local con x'_p como solución inicial; denótese con x''_p el mínimo local así obtenido.

(c) Moverse o no. Si la solución obtenida x''_p es mejor que x , hacer $x \leftarrow x''_p$ y $k \leftarrow 1$; en otro caso, hacer $k \leftarrow k + 1$.

Figura 8. VNS Paralela; PVNS.

3.5. Híbridos

Dado que el cambio sistemático de estructura de entornos es una herramienta simple y muy potente, otra forma de extender la VNS ha sido incorporarla a otras metaheurísticas. Estas propuestas han dado lugar a diversas metaheurísticas híbridas.

La búsqueda tabú (Tabu Search, TS) , , , generalmente usa una estructura de entornos con respecto a la que ejecuta movimientos de ascenso y descenso explotando diferentes tipos de memoria. En principio hay dos maneras de hacer híbridos de VNS y TS: usar algún tipo de memoria para orientar la búsqueda dentro de la VNS o usar la VNS dentro de la TS. En , , y se proponen híbridos del primer tipo y en y del segundo tipo.

La metaheurística GRASP (*Greedy Randomized Adaptive Search Procedure*) consta de dos fases; en la primera fase se construyen soluciones usando un procedimiento *greedy* aleatorizado y en la segunda, las soluciones se mejoran por alguna búsqueda local o un método enumerativo. Una forma natural de hibridizar la VNS con GRASP es usar la VNS en la segunda fase de GRASP lo que ha sido aplicado en , , , , .

La búsqueda Multi-arranque (*MultiStart Search*, MS) es una metaheurística clásica que, para evitar el estancamiento de un procedimiento descendente en un óptimo local, sencillamente reinicia la búsqueda desde otra solución. En se propone y analiza una heurística híbrida entre la VNS y la MS consistente en reiniciar la VNS desde otra solución generada al azar del espacio x , cuando se estanca en un mínimo local, por no encontrar ninguna mejora a través de ninguno de los entornos de la solución x .

En combinación con el método Piloto se ha propuesto recientemente . La VNS se combina con operadores genéticos en .

4. La VNS en planificación Logística

La VNS ha alcanzado éxitos relevantes en campos de aplicación diversos entre los que caben destacar los de unas técnicas de descubrimiento que han permitido descubrir una herramienta informática (el Sistema AutoGraphiX ,) que ha conseguido probar o refutar algunas conjeturas abiertas en teoría de Grafos. Se ha aplicado a problemas de optimización continua creando un software específico GLOB . También se ha aplicado con éxito a multitud de problemas de optimización combinatoria importantes en la industrial y la

economía, entre los que se encuentran los problemas más relevantes de planificación logística: problemas de empaquetado, problemas de localización y problemas de rutas.

Los **Problemas de Empaquetado** constituyen una clase de problemas importantes en contextos de logística y distribución. En se usa la VNS básica para una de las versiones básicas: el problema de empaquetado unidimensional (*Bin-Packing Problem*, BPP) en el que se tiene que empaquetar un conjunto de objetos de diferente tamaño en el menor número de cajas o segmentos de capacidad fija.

Los **Problemas de Localización**, muy relevantes en la planificación logística, y los de agrupamiento o *clustering* (dentro de la clasificación no supervisada) tienen características comunes que los hacen similares al ser abordados por búsquedas metaheurísticas. El problema de la p -mediana es el problema más extensamente estudiado en Teoría de Localización. Consiste en determinar p ubicaciones, entre un conjunto de m localizaciones posibles para un servicio, de forma que se minimice la suma de las distancias de n usuarios a su punto de servicio más cercano. Este problema ha sido abordado con éxito en y por medio de una VNS básica, una VNS reducida y una VNDS, respectivamente. En y se proponen y prueban diversas VNS paralelas para este problema. En se aplica la VNS al problema del p -centro en el que hay que minimizar, en lugar de la suma, el máximo de las distancias de los usuarios a sus respectivos puntos de servicios más próximos. Diversas variantes de la VNS han sido aplicadas con éxito en y al problema múltiple de Weber que es la versión continua del problema de la p -mediana, en el que las p localizaciones pueden elegirse en todo el plano donde también están ubicados los usuarios. En se muestra la aplicación de la RVNS y la VNDS para el problema simple de localización de plantas (*Simple Plant Location Problem*, SPLP) en el que se debe decidir el número p de localizaciones, minimizando la suma del coste total que implica la separación de los usuarios a los puntos de servicio, y el coste de la selección de cada ubicación para el servicio. El problema de asignación cuadrática (*Quadratic Assignment Problem*, QAP) es otro problema relevante que surge en localización y ha sido abordado con la VNS en .

Otro tipo de problemas de optimización combinatoria importante, sobre todo en el contexto de la planificación logística, son los **Problemas de Rutas**. Tanto las versiones clásicas del problema del vendedor o viajante de comercio (*Travelling Salesman Problem*, TSP), del problema de rutas de vehículos (*Vehicle Routing Problem*, VRP) y del problema de rutas de arcos (*Arc Routing Problem*, ARP), así como algunas de sus extensiones, han sido abordadas con la VNS. El problema del viajante de comercio o TSP consiste en, dadas n ciudades con las distancias o costes entre ellas, encontrar una ruta de mínimo coste (es decir, una permutación de las ciudades que minimiza la suma de las n distancias entre ciudades adyacentes en la ruta). En , y se aplican procedimientos del tipo de la VNS con distinto tipo de movimiento para el TSP. En se aplica una VNDS similar con dos esquemas de descomposición diferentes. En y se usa una VNS para resolver otra importante extensión del TSP denominada *problema del comprador* en el que, dada una partición del conjunto de clientes, hay que visitar al menos uno de cada conjunto de la partición. Un trabajo interesante sobre el TSP con recogida y distribución es .

Un problema de ruta de vehículos o VRP consiste en diseñar las rutas desde un depósito para visitar un conjunto dado de clientes con mínimo coste. Los clientes tienen demandas conocidas y los vehículos tienen capacidad limitada. En y se aplican la VNS y la VND en la resolución de la variante del problema en el que cada cliente puede ser visitado en cierto intervalo de tiempo que se denomina VRPTW (*VRP with Time Windows*) y en se trata un problema múltiple de este tipo. En se explora el uso de una VND para un VRP con recogida. En se considera un problema on-line muy general.

En se propone una combinación de la VNS y la TS que usa varias estructuras de entornos para el problema del ciclo mediano (*Median Cycle Problem*, MCP). En este problema se debe determinar la ruta de menor longitud que recorra parte de las ciudades con una cota superior para la suma de las distancias desde las ciudades no incluidas en el recorrido hasta la ciudad más cercana de la ruta. En se aplica una VNS a una versión del problema del viajante de comercio con clientes de recogida y entrega de mercancías. La ruta tiene que recorrer todos los clientes de recogida antes que los de entrega, partiendo y llegando al depósito. En , y se aborda con una VNS de forma exitosa problemas de rutas de arcos en los que las rutas deben recorrer todas las aristas o arcos de un grafo o red.

5. Conclusiones.

Las heurísticas más tradicionales en optimización realizaban búsquedas locales descendentes por lo que se bloquean con el primer óptimo local encontrado. Las metaheurísticas proporcionaron métodos para escaparse de los óptimos locales de mala calidad. El valor de tales óptimos locales difiere considerablemente del valor del óptimo global y, especialmente si hay muchos óptimos locales, la mejora en la calidad de los nuevos óptimos alcanzados era insuficiente para acercarse suficientemente. Sin embargo, las mejoras introducidas en las estrategias básicas y los recursos computacionales de las implementaciones han permitido obtener procedimientos cada vez rápidos en acercarse con garantías a la solución óptima por lo que el impacto práctico de las metaheurísticas ha sido inmenso.

En contraste con este éxito, el desarrollo teórico de resultados sobre metaheurísticas está más retrasado. Frecuentemente se obtienen buenas heurísticas con algo de inventiva, experiencia e intuición. y con un gran esfuerzo en el ajuste de numerosos parámetros se van mejorando. Pero las razones de porqué unas estrategias funcionan tan bien como lo hacen para algunos problemas y otras no, y estas funcionan mejor que aquellas para otros problemas son desconocidas. La situación es incluso peor con los híbridos y las mejoras computacionales; muchas veces es difícil discernir si el mérito lo tiene una de las componentes o se está obteniendo beneficio de la interacción.

El desarrollo y aplicación de la VNS ha contribuye a profundizar en algunas cuestiones que influyen en el rendimiento general de las metaheurísticas. Tres de estas ideas son la topografía de los óptimos locales, el las trayectorias seguidas por los procesos en su acercamiento al óptimo, y la dicotomía entre la mejor o la primera mejora. El estudio de la topografía de los mínimos locales se realiza en términos de una descripción de los *perfiles de montañas y valles* encontrados durante la búsqueda. Cuando se aplica la VNS, el descenso desde una solución x' , seleccionada al azar, puede volver al óptimo local x alrededor del que están centrados los entornos actuales, o a otro óptimo local x'' cuyo valor puede ser mejor o no que el de x . Se ha estudiado en la probabilidad de estos tres sucesos como una función de la distancia de x a x' al aplicarla al problema de la máxima satisfabilidad ponderada. Conviene también observar si x'' está más cerca de x que x' (lo que puede ser interpretado como que x'' pertenece al mismo valle que x , con rugosidades locales en relieve) o no (lo que puede interpretarse como un indicio de que se ha encontrado un nuevo gran valle). El relieve suministra esta información en base a una determinada cantidad de descensos desde puntos en entornos sucesivamente más amplios. Los perfiles varían considerablemente con la calidad del óptimo local x . Cuando x es una mala solución es suficiente alejarse un poco para obtener, con alta probabilidad, un óptimo local mejor. Cuando el óptimo local x es bueno, o muy bueno, debe alejarse bastante más para encontrar un nuevo gran valle y, además, la probabilidad de encontrar una solución mejor que la actual es entonces baja. Esto ilustra una debilidad del esquema básico de la VNS que tiende a degenerar en un *arranque múltiple* cuando la distancia de x a x' se hace grande. El remedio es el proporcionado por el esquema de la VNS sesgada.

Para algunos problemas se dispone de instancias para las que se conoce el óptimo global y se puede analizar la *distancia al blanco* durante el proceso de búsqueda. Es corriente analizar las búsquedas heurística determinando lo frecuente que se alcanza el óptimo global o el promedio de la diferencia relativa entre el valor óptimo del objetivo y el alcanzado por la búsqueda. Sin embargo, se obtiene mucha más información si se consideran las propias soluciones y no sólo su valoración. Una herramienta para poner en práctica esta idea ha sido desarrollada para aplicar una VNS para el problema del vendedor en . Esta herramienta presenta en pantalla la solución óptima para el caso de entrenamiento bajo estudio, la solución actual y la diferencia simétrica entre estas dos soluciones. Esto indica cuanta mejora queda por hacer y donde. Además, una rutina permite también la representación de la *diferencia* entre la solución actual en una iteración y en la siguiente. Finalmente, como las representaciones de las soluciones para problemas grandes pueden ser difíciles de leer, y muchos problemas, en particular los problemas euclídeos, permiten una descomposición natural, una rutina de *enfoque* permite la representación de la información mencionada anteriormente para subproblemas seleccionados; es decir, en alguna región del espacio en la que se traza la ruta. Además de usar esta información para guiar la búsqueda, se puede evaluar el trabajo realizado paso a paso por una VNS u otra metaheurística. Se pueden estudiar variantes en cada uno de esos pasos y recopilar información detallada

para sacar conclusiones que pueden no ser evidentes en un rendimiento global de la heurística. Esto puede dar lugar al descubrimiento de fenómenos inesperados y, la profundización proporcionada por su explicación, puede a su vez dar lugar a principios para mejorar las heurísticas.

En muchos trabajos se ha observado que cuando se aplica un paso de una la heurística buscando iterativamente un movimiento de mejora dentro de un conjunto de posibilidades se puede optar por dos tácticas bien diferenciadas: la primera mejora (es decir, el primer movimiento que reduce el valor de la función objetivo) o la mejor mejora, (es decir, el movimiento que más reduce el valor de la función objetivo). Del análisis de las características de los entornos se deduce si al acercarse al óptimo local es conveniente realizar el esfuerzo de recorrer completamente el entorno de la solución actual o en los estadios iniciales de la búsqueda las mejoras encontradas inicialmente son suficientemente significativas para hacer irrelevante el análisis exhaustivo del entorno.

Los criterios de comparación entre las metaheurísticas deben referirse al mayor o menor grado de cumplimiento de las propiedades *deseables* de las metaheurísticas. Tales buenas cualidades son las que propician o garantizan el interés práctico y teórico de las propuestas. En se propone una propuesta de relación de dichas propiedades. Esta relación es similar a las propuestas por otros autores. En se analiza en que medida la VNS en relación a otras metaheurísticas se ajustan a tales propiedades. Algunas de tales propiedades son las siguientes: la simplicidad, la precisión, la coherencia, la efectividad, la eficacia, la eficiencia, la robustez, la interactividad y la innovación

La VNS está basada en un principio simple poco explorado: el cambio sistemático de la estructura de entornos durante la búsqueda. Esta *simplicidad* de la metaheurística contribuye a dotarla de la amplia aplicabilidad que se refleja en la variedad de las aplicaciones aparecidas en la naturaleza. La VNS está dotada de reglas *precisas* que describen la forma de efectuar tales cambios. Todos los pasos de los esquemas básicos y extendidos de la VNS se traducen *coherentemente* de los principios en que se inspira. La *eficacia* de la VNS se sustenta en la probabilidad de encontrar soluciones óptimas para una gran cantidad de problemas en los que ha sido probada superior o equivalente a otras metaheurísticas con la que ha sido comparada. La VNS tiene probada efectividad en la resolución de problemas de varios bancos de prueba con resultados óptimos o muy cercanos a los óptimos, y con tiempo computacional moderado (o al menos razonable). La metaheurística VNS se ha mostrado considerablemente *eficiente* en muchos experimentos al obtener de una forma más rápida resultados mejores o equivalentes a los de otras metaheurísticas. Además, la VNS se muestra *robusta* ya que ha probado su rendimiento en multitud de problemas sin necesidad de realizar un ajuste específico de parámetros al conjunto de entrenamiento. Las metaheurísticas VNS se ha extendido considerablemente al hibridizarse con otros procedimientos mejorando su efectividad. Cabe destacar además la frecuencia con la que las metaheurísticas incorporan las ideas en las que se basa la VNS para beneficiarse de sus efectos positivos. La *amigabilidad* de los sistemas basados en la VNS cuenta con la ventaja de que los principios básicos en que se basa son fáciles de usar con un número de parámetros muy pequeño, incluso inexistentes en algunos casos. Las posibilidades de la VNS para la *innovación* están ampliamente corroborada con los resultados obtenidos con el programa AutoGraphiX AGX de descubrimiento científico asistido por ordenador que está basado en VNS y del que ya se han publicado gran cantidad de resultados bajo el título común *Variable Neighborhood Search for Extremal Grapas*.

Las líneas futuras tienen que ir por las mejoras necesarias para poder resolver eficientemente problemas difíciles y muy grandes. El tamaño de los problemas abordados se limita en la práctica por las herramientas disponibles para resolverlos más que por la necesidad de los potenciales usuarios de estas herramientas.

Referencias.

- [1] A. Andreatta, C. Ribeiro. Heuristics for the phylogeny problem. *Journal of Heuristics*, 8:429-447, 2002.
- [2] R.K. Ahuja, J.B. Orlin, D. Sharma (2000). Very large-scale neighborhood search, *International Transactions in Operations Research*, 7: 301-317.

- [3] M. Aouchiche, J.M. Bonneyoy, A. Fidahoussen, G. Caporossi, P. Hansen, L. Hiesse, J. Lacheré, A. Monhait (2005). *Variable Neighborhood Search for Extremal Graphs, 14. The AutoGraphiX 2 System*, En: Liberti L., Maculan N. (eds), *Global Optimization: from Theory to Implementation*, Springer, 2005.
- [4] N. Belacel, P. Hansen, N. Mladenović (2002). Fuzzy J-means: a new heuristic for fuzzy clustering. *Pattern Recognition*, 35(10):2193-2200.
- [5] O. Bräysy (2003). A reactive variable neighborhood search algorithm for the vehicle routing problem with time windows. *INFORMS Journal on Computing*, 15, 347–368..
- [6] J. Brimberg, P. Hansen, N. Mladenović, E. Taillard (2000). Improvements and comparison of heuristics for solving the multisource weber problem. *Operations Research*, 48(3): 444-460.
- [7] J. Brimberg, N. Mladenović (1996). A variable neighborhood algorithm for solving the continuous location-allocation problem. *Studies in Locational Analysis*, 10:1-12
- [8] E.K. Burke, P. De Causmaecker, S. Petrović, G.V. Berghe (2001). Variable neighborhood search for nurse rostering. In *MIC'2001*, pages 755-760, Porto.
- [9] E.K. Burke, P. Cowling, R. Keuthen (2001). Effective local and guided variable neighborhood search methods for the asymmetric travelling salesman problem. *Lecture Notes in Computer Science*, 2037:203-212.
- [10] S. Canuto, M. Resende, C. Ribeiro (2001). Local search with perturbations for the prize-collecting Steiner tree problem in graphs. *Networks*, 38:50-58.
- [11] G. Caporossi, D. Cvetković, I. Gutman, P. Hansen (1999). Variable neighborhood search for extremal graphs 2: Finding graphs with extremal energy. *Journal of Chemical Information and Computer Science*, 39:984-996.
- [12] G. Caporossi, I. Gutman, P. Hansen (1999). Variable neighborhood search for extremal graphs 4: Chemical trees with extremal connectivity index. *Computers and Chemistry*, 23:469-477
- [13] G. Caporossi, P. Hansen (2000). Variable neighborhood search for extremal graphs 1: The Auto GraphiX system. *Discrete Mathematics*, 212:29-44.
- [14] F. Carrabs, J.-F. Cordeau, G., Laporte (2006). Variable Neighbourhood Search for the Pickup and Delivery Traveling Salesman Problem with LIFO Loading, to appear in *INFORMS Journal on Computing*
- [15] R. Cordone, R. W. Calvo (2001). A heuristic for the vehicle routing problem with time windows. *Journal of Heuristics*, 7(2):107-129.
- [16] M.C. Costa, F.R. Monclar, M. Zrikem (2001). Variable neighborhood search for the optimization of cable layout problem. In *MIC'2001*, pages 749-753, Porto.
- [17] T.G. Crainic, M. Gendreau, P. Hansen, N. Mladenović (2004). Cooperative parallel variable neighbourhood search for the p -median. *Journal of Heuristics* 10, 293–314.
- [18] J. Crispim, J. Brandao (2001). Reactive tabu search and variable neighborhood descent applied to the vehicle routing problem with backhauls. In *MIC'2001*, pages 631-636, Porto.
- [19] T. Davidović, P. Hansen, N. Mladenović (2005). Permutation based genetic, tabu and variable Neighborhood search heuristics for multiprocessor scheduling with communication delays. *Asia-Pacific Journal of Operational Research* 22, 297–326.
- [20] M. Drazic, V. Kovacevic-Vujcic, M. Cangalovic, N. Mladenovic (2006). GLOB - A new VNS-based software for global optimization. In L. Liberti, N. Maculan, (eds.), *Global Optimization: from Theory to Implementation*, Vol. 84 de *Nonconvex Optimization and its Application Series*, Springer 2006.

- [21] C.W. Duin, S. Voss (1999). The pilot method: A strategy for heuristic repetition with application to the Steiner problem in graphs. *Networks*, 34:181–191.
- [22] T. Feo, M. Resende (1995). Greedy randomized adaptive search. *Journal of Global Optimization*, 6:109-133.
- [23] P. Festa, P. Pardalos, M. Resende, C. Ribeiro (2001). GRASP and VNS for max-cut. In *MIC'2001*, pages 371-376, Porto..
- [24] K. Fleszar, K.S. Hindi (2002). New heuristics for one-dimensional bin-packing. *Computers & Operations Research*, 29(7):821-839, 2002.
- [25] F. García López, M.B. Melián Batista, J.A. Moreno Pérez, J.M. Moreno Vega (2001). The parallel variable neighborhood search for the p -median problem. *Journal of Heuristics*, 8, 377-390.
- [26] C.G. García, D. Pérez-Brito, V. Campos, R. Martí (2006). Variable Neighborhood Search for the Linear Ordering Problem. *Computers and Operations Research*, 33:3549-3565.
- [27] M.R. Garey, D.S. Johnson (1978). *Computers and Intractability: A Guide to the Theory of NP-Completeness*, Freeman, New-York.
- [28] M. Gendreau, P. Hansen, M. Labbé, N. Mladenović (2001). Variable neighborhood search for the traveling salesman problem. (in preparation).
- [29] G. Ghiani, A. Hertz, G. Laporte (2000). Recent algorithmic advances for arc routing problems. *Les Cahiers du GERAD*, G-2000-40.
- [30] F. Glover (1989). Tabu search - part I. *ORSA Journal of Computing*, 1:190-206.
- [31] F. Glover (1990). Tabu search - part II. *ORSA Journal of Computing*, 2:4-32.
- [32] F. Glover, M. Laguna (1997). *Tabu search*. Kluwer.
- [33] F. Glover, G. Kochenberger (eds.) (2003). *Handbook of Metaheuristics*. volume 57 of *International Series in Operations Research & Management Science*, Kluwer.
- [34] F. Glover, B. Melián (2003). Tabu Search. *Inteligencia Artificial, Revista Iberoamericana de Inteligencia Artificial*, 19(2):29-48, 2003.
- [35] A. Goel, V. Gruhn (2006). The General Vehicle Routing Problem. *EJOR*, por aparecer
- [36] P. Hansen, J. Brimberg, D. Urosevic, N. Mladenovic (2003). Primal - dual variable neighbourhood search for exact solution of the simple plant location problem (in preparation).
- [37] P. Hansen, J. Jaumard, N. Mladenović, A.D. Parreira (2000). Variable neighbourhood search for maximum weight satisfiability problem. *Les Cahiers du GERAD*, G-2000-62.
- [38] P. Hansen, N. Mladenović (1997). Variable neighborhood search for the p -median. *Location Science*, 5:207-226.
- [39] P. Hansen, N. Mladenović (1999). First improvement may be better than best improvement: An empirical study. *Les Cahiers du GERAD*, G-99-40.
- [40] P. Hansen, N. Mladenović (1999). An introduction to variable neighborhood search. Capítulo 30 en S. Voss, S. Martello, I Osman, C. Roucairol (eds.), *Metaheuristics: Advances and Trends in Local Search Paradigms for Optimization*, pág. 433-458. Kluwer, 1999.
- [41] P. Hansen, N. Mladenović (2000). A separable approximation dynamic programming algorithm for economic dispatch with transmission losses. *Les Cahiers du GERAD*, G-2000-31.
- [42] P. Hansen, N. Mladenović (2001). Developments of variable neighborhood search. En C. Ribeiro, P. Hansen (eds.), *Essays and Surveys in Metaheuristics*, pág. 415-440. Kluwer, 2001.

- [43] P. Hansen, N. Mladenović (2001). Industrial applications of the variable neighborhood search metaheuristics. En G. Zaccour (editor), *Decisions and Control in Management Science*, págs 261-274. Kluwer, 2001.
- [44] P. Hansen, N. Mladenović (2001). J-means: A new local search heuristic for minimum sum-of-squares clustering. *Pattern Recognition*, 34(2):405-413, 2001.
- [45] P. Hansen, N. Mladenovic (2001). Variable neighborhood search: Principles and applications. *European Journal of Operational Research*, 130:449-467.
- [46] P. Hansen, N. Mladenovic (2003). Variable neighbourhood search. Chapter 6 en F.W. Glover, G.A. Kochenberger (eds.), *Handbook of Metaheuristics*, pp. 145-184. Kluwer, 2003.
- [47] P. Hansen, N. Mladenovic (2002). Variable neighbourhood search. In Panos M. Pardalos, Mauricio G. C. Resende (eds.), *Handbook of Applied Optimization*.
- [48] P. Hansen, N. Mladenovic (2003). A tutorial on variable neighborhood search. *Les Cahiers du GERAD*, G-2003-46.
- [49] P. Hansen, N. Mladenovic, J. Brimberg (2002), Convergence of variable neighbourhood search, *Les Cahiers du GERAD* G-2002-21.
- [50] P. Hansen, N. Mladenovic, J.A. Moreno Pérez (2003). Búsqueda de Entorno Variable. *Inteligencia Artificial. Revista Iberoamericana de Inteligencia Artificial*. Número 19. Vol. 2, 77-92
- [51] P. Hansen, N. Mladenović, D. Pérez Brito (2001). Variable neighborhood decomposition search. *Journal of Heuristics*, 7(4):335-350.
- [52] H. Höller, B. Melián, S. Voss (2006). Applying the pilot method to improve VNS and GRASP metaheuristics for the design of SDH/WDM networks. *EJOR*, por aparecer
- [53] V. Kovacevic, M. Cangalovic, M. Asic, D. Drazic, L. Ivanovic (1999). Tabu search methodology in global optimization. *Computers & Mathematics with Applications*, 37:125--133.
- [54] L. Lobjois, M. Lemaitre, G. Verfaillie (2001). Large Neighbourhood Search using Constraint Propagation and Greedy Reconstruction for Valued CSP resolution. In *14th European Conference on Artificial Intelligence (ECAI'2000)*, Berlin, August 2000, 2001. Workshop on Modelling and Solving Problems with Constraints.
- [55] S. Loudin, P. Boizumault (2001). VNS/ LDS + Cp: A hybrid method for constraint optimization in anytime contexts. In *MIC'2001*, pages 761-765, Porto.
- [56] R. Martí (2002). Multi-start methods. In Fred Glover, Gary A. Kochenberger, editors, *Handbook of Metaheuristics*, chapter 12. Kluwer Academic.
- [57] R. Martí, J.M. Moreno Vega (2003). Métodos Multiarranque. *Inteligencia Artificial. Revista Iberoamericana de Inteligencia Artificial*, Vol. 19 pp. 49-60
- [58] S.L. Martins, M.G.C. Resende, C.C. Ribeiro, P. Pardalos (2000). A parallel GRASP for the Steiner tree problem in graphs using a hybrid local search strategy. *Journal of Global Optimization*, 17:267-283.
- [59] B. Melián Batista, J.A. Moreno Pérez, J.M. Moreno Vega (2003). Metaheurísticas: una visión global. *Inteligencia Artificial. Revista Iberoamericana de Inteligencia Artificial*. 2003. Número 19. Vol. 2, 7-28
- [60] N. Mladenović (1995). A variable neighborhood algorithm - a new metaheuristic for combinatorial optimization. In *Abstracts of Papers Presented at Optimization Days*, pág. 112.
- [61] N. Mladenovic, M. Drazic, V. Kovacevic-Vujcic, M. Cangalovic (2006). General variable neighborhood search for the continuous optimization. Sometido a *EJOR*

- [62] N. Mladenović, P. Hansen (1997). Variable neighborhood search. *Computers & Operations Research*, 24:1097-1100.
- [63] N. Mladenovic, M. Labbe, P. Hansen (2003). Solving the p -center problem with tabu search and variable neighborhood search. *Networks*, 48(1) 48-64.
- [64] N. Mladenovic, J. Petrovic, V. Kovacevic-Vujcic, M. Cangalovic (2004). Solving spread spectrum radar polyphase code design problem by tabu search and variable neighborhood search. *European Journal of Operational Research*, (to appear).
- [65] N. Mladenović, D. Urõsević (2001). Variable neighborhood search for the k -cardinality. In *MIC'2001*, pages 749-753, Porto, 2001.
- [66] J.A. Moreno Pérez, P. Hansen, N. Mladenovic (2005). Parallel Variable Neighborhood Search. Capítulo 11 en E. Alba, editor, *Parallel Metaheuristics: A New Class of Algorithms*, Wiley, 2005.
- [67] J.A. Moreno Pérez, J.M. Moreno Vega, I. Rodríguez Martín, (2003). Variable neighborhood tabu search and its application to the median cycle problem. *European Journal of Operations Research*, 151(2):365-378.
- [68] L.S. Ochi, M.B. Silva, L. Drummond (2001). Metaheuristics based on GRASP and VNS for solving traveling purchaser problem. In *MIC'2001*, pages 489-494.
- [69] G. Pesant, M. Gendreau (1996). A view of local search in constraint programming, principles and practice of constraint programming. *Lecture Notes in Computer Science*, 1118:353-366.
- [70] G. Pesant, M. Gendreau (1999). A constraint programming framework for local search methods. *Journal of Heuristics*, 5:255-279, 1999.
- [71] M. Polacek, R. Hartl, K. Doerner, M. Reimann (2004). A variable neighbourhood search for the multi depot vehicle routing problem with time windows. *Journal of Heuristics* 10, 613–627.
- [72] M.G.C. Resende nad R. Werneck (2003). On the implementation of a swap-based local search procedure for the p -median problem. In: Proceedings of the *Fifth Workshop on Algorithm Engineering and Experiments (ALENEX'03)*, R.E. Ladner (ed.), SIAM, Philadelphia, pp. 119-127.
- [73] M.G.C. Resende, J.L. [González Velarde, J.L.](#) (2003). GRASP: Procedimientos de búsquedas miopes aleatorizados y adaptativos. *Inteligencia Artificial. Revista Iberoamericana de Inteligencia Artificial*. Número: 19 Páginas: 61 - 76
- [74] C.C. Ribeiro, M.C. Souza (2002). Variable neighborhood descent for the degree-constrained minimum spanning tree problem. *Discrete Applied Mathematics*, 118:43-54.
- [75] C.C. Ribeiro, E. Uchoa, R. Werneck (2002). A hybrid GRASP with perturbations for the Steiner problem in graphs. *INFORMS Journal of Computing*, 14:228-246.
- [76] M.B. Silva, L. Drummond, L.S. Ochi (2000). Variable neighborhood search for the traveling purchaser problem. In *27th Int. Conference on Computational Industrial Engineering*, 2000.
- [77] E. Taillard, S. Voss (2001). POPMUSIC - partial optimization metaheuristic under special intensification conditions. In C. Ribeiro, P. Hansen, editors, *Essays and Surveys in Metaheuristics*, pages 613-630. Kluwer.
- [78] Q.-H. Zhao, S. Chen, C.-X. Zang. (2006). Model and Algorithm for Inventory & Routing Decision in a Three-Echelon Logistics System. Submitted to EJOR