Ann Oper Res (2010) 175: 367–407
DOI 10.1007/s10479-009-0657-6

Variable neighbourhood search: methods and applications

Pierre Hansen · Nenad Mladenović · José A. Moreno Pérez

Published online: 28 October 2009

Abstract Variable neighbourhood search (VNS) is a metaheuristic, or a framework for building heuristics, based upon systematic changes of neighbourhoods both in descent phase, to find a local minimum, and in perturbation phase to emerge from the corresponding valley. It was first proposed in 1997 and has since then rapidly developed both in its methods and its applications. In the present paper, these two aspects are thoroughly reviewed and an extensive bibliography is provided. Moreover, one section is devoted to newcomers. It consists of steps for developing a heuristic for any particular problem. Those steps are common to the implementation of other metaheuristics.

Keywords Variable neighbourhood search · Metaheuristic · Heuristic

1 Introduction

The VNS survey in this paper provides an update to the 2008 version which appeared in *4OR. A Quarterly Journal* (Hansen et al. 2008b). A short description of 21 recent successful applications of VNS are added in Sect. 5.

Variable neighbourhood search (VNS) is a metaheuristic, or framework for building heuristics, aimed at solving combinatorial and global optimization problems. Its basic idea

This is an updated version of the paper that appeared in *4OR. A Quarterly Journal of Operations Research* 6(4):319–360 (2008).

P. Hansen
GERAD and HEC Montréal, Montréal, (Québec) H3T 2A7, Canada
e-mail: pierre.hansen@gerad.ca

N. Mladenović (✉)
GERAD and School of Mathematics, Brunel University, Uxbridge, United Kingdom
e-mail: nenad.mladenovic@brunel.ac.uk

J.A. Moreno Pérez
IUDR and DEIOC, University of La Laguna, La Laguna, Spain
e-mail: jamoreno@ull.es

consists in a systematic change of neighbourhood combined with a local search. Since its inception, VNS has undergone many developments and been applied in numerous fields. We review below the basic rules of VNS and of its main extensions. In addition, some of the most successful applications are briefly summarized. Pointers to many other applications are given in the reference list.

A deterministic optimization problem may be formulated as

$$\min\{f(x)|x \in X, X \subseteq \mathcal{S}\}, \tag{1}$$

where $\mathcal{S}$, X, x and f respectively denote the *solution space* and *feasible set*, a *feasible solution* and a real-valued *objective function*. If $\mathcal{S}$ is a finite but large set, a *combinatorial optimization* problem is defined. If $\mathcal{S} = \mathbb{R}^n$, we refer to *continuous optimization*. A solution $x^* \in X$ is *optimal* if

$$f(x^*) \le f(x), \quad \forall x \in X.$$

An *exact algorithm* for problem (1), if one exists, finds an optimal solution x^*, together with the proof of its optimality, or shows that there is no feasible solution, i.e., $X = \emptyset$. Moreover, in practise, the time needed to do so should be finite (and not too long). When one deals with a continuous function, it is reasonable to allow for some degree of tolerance, i.e., to stop when a feasible solution x^* has been found such that

$$f(x^*) < f(x) + \varepsilon, \quad \forall x \in X \quad \text{or} \quad \frac{f(x^*) - f(x)}{f(x^*)} < \varepsilon, \quad \forall x \in X$$

for some small positive ε.

Many practical instances of problems of form (1), arising in Operations Research and other fields, are too great for an exact solution to be found in reasonable time. It is well-known from complexity theory (Garey and Johnson 1978; Papadimitriou 1994) that thousands of problems are *NP-hard*, such that no algorithm with a number of steps polynomial in the size of the instances is known for solving any of them and that if one were found it would be a solution for all. Moreover, in some cases where a problem admits a polynomial algorithm, this algorithm may be such that realistic size instances cannot be solved in reasonable time in the worst case, and sometimes also in the average case or in most cases.

This explains the need to resort to heuristics which speedily yield an approximate solution, or sometimes an optimal solution but one which has no proof of its optimality. Some of these heuristics have a worst-case guarantee, i.e., the solution x_h obtained satisfies

$$\frac{f(x_h) - f(x)}{f(x_h)} \le \varepsilon, \quad \forall x \in X \tag{2}$$

for some ε, though this is rarely small. Moreover, this upper bound ε on the worst-case error is usually much larger than the average error observed in practise and may therefore be a bad guide in selecting a heuristic. In addition to avoiding excessive computing time, heuristics address another problem: local optima. A local optimum x_L of problem (1) is such that

$$f(x_L) \le f(x), \quad \forall x \in N(x_L) \cap X \tag{3}$$

where $N(x_L)$ denotes a neighbourhood of x_L (ways to define such a neighbourhood will be discussed below). If there are many local minima, the range of values which they span may be large. Moreover, the globally optimum value $f(x^*)$ may differ substantially from

the average value of the local minima, or even from the best such value among many, obtained by some simple heuristic such as multistart (a phenomenon called the Tchebycheff catastrophe in Baum 1986). There are, however, many ways to escape from local optima or, more precisely, from the valleys which contain them.
Metaheuristics are general frameworks to build heuristics for combinatorial and global optimization problems. For discussion of the best-known of them, the reader is referred to the following survey books Reeves (1993), Glover and Kochenberger (2003) and Burke and Kendall (2005). Some of the many successful applications of metaheuristics are also mentioned there.
Variable Neighborhood Search (VNS) (Mladenović 1995; Mladenović and Hansen 1997; Hansen and Mladenović 1997, 1999, 2001a, 2001c, 2003) is a metaheuristic which systematically exploits the idea of neighbourhood change, both in descent to local minima and in escape from the valleys which contain them. VNS heavily relies upon the following observations:

Fact 1 A local minimum with respect to one neighbourhood structure is not necessarily a local minimum for another neighbourhood structure.

Fact 2 A global minimum is a local minimum with respect to all possible neighbourhood structures.

Fact 3 For many problems local minima with respect to one or several neighbourhoods are relatively close to each other.

This last observation is empirical. It implies that a local optimum often provides some information about the global optimum. For instance, it may be the case that there are several variables with the same value in both. However, it is not usually known which ones are of this kind. An organized study of the neighbourhood of this local optimum is therefore in order, until a better one is found.

Unlike many other metaheuristics, the basic schemes of VNS and its extensions are simple and require few, and sometimes no parameters. Therefore, in addition to providing very good solutions, often in simpler ways than other methods, VNS gives insight into the reasons for such a performance, which, in turn, can lead to more efficient and sophisticated implementations.

The paper is organized as follows. Background ideas, which in part inspired VNS, are briefly discussed in Sect. 2. Basic schemes are reviewed in Sect. 3. Section 4 is devoted to newcomers. The steps for developing heuristics for any particular problem are given. Most of those steps are common to the implementation of other metaheuristics. Then some tips which can help to improve the current VNS version are listed. Various applications are classified and surveyed in Sect. 5. Section 6 lists those desirable properties of metaheuristics that are enjoyed by VNS.

The purpose of this paper is threefold: (i) to present to researchers the main ideas and schemes of VNS; (ii) to provide an extensive list of successful applications and (iii) to (gently) introduce newcomers into the metaheuristics area.

2 Background

VNS embeds a local search heuristic for solving combinatorial and global optimization problems. This idea has had some predecessors. It allows a change of the neighbourhood structures within this search. In this section, we give a brief introduction to the variable metric algorithm for solving continuous convex problems and local search heuristics for solving combinatorial and global optimization problems.

2.1 Variable metric method

The variable metric method for solving unconstrained continuous optimization problem (1) has been suggested by Davidon (1959) and Fletcher and Powell (1963). The idea is to change the metric (and thus the neighbourhood) at each iteration such that the search direction (steepest descent with respect to the current metric) adapts better to the local shape of the function. In the first iteration a Euclidean unit ball in the n dimensional space is used and the steepest descent (anti-gradient) direction found. At subsequent iterations, ellipsoidal balls are used and the steepest direction of descent is obtained with respect to a new metric resulting from a linear transformation. The purpose of such changes is to build up, iteratively, a good approximation to the inverse of the Hessian matrix A^{-1} of f, that is, to construct a sequence of matrices H_i with the property,

$$\lim_{i\to\infty} H_i = A^{-1}.$$

In the convex quadratic programming case, the limit is achieved after n iterations instead of an infinity of them. In this way the so-called Newton search direction is obtained. The advantages are that: (i) it is not necessary to find the inverse of the Hessian (which requires $O(n^3)$ operations) at each iteration; (ii) the second order information is not needed. Assume that the function $f(x)$ is approximated by its Taylor series

$$f(x) = \frac{1}{2}x^T Ax - b^T x \tag{4}$$

with positive definite matrix A (denoted by $A > 0$). Applying the first order condition $\nabla f(x) = Ax - b = 0$ we have $Ax_{\text{opt}} = b$, where x_{opt} is a minimum point. At the current point we have $Ax_i = \nabla f(x_i) + b$. We will not rigorously derive here the Davidon-Fletcher-Powell (DFP) algorithm for transforming H_i into H_{i+1}. Let us mention only that subtracting one of these last two equations from the other and multiplying (from the left) by the inverse matrix A^{-1}, we have

$$x_{\text{opt}} - x_i = -A^{-1}\nabla f(x_i).$$

Subtracting this last equation evaluated at x_{i+1} from the same equation at x_i gives

$$x_{i+1} - x_i = -A^{-1}(\nabla f(x_{i+1}) - \nabla f(x_i)). \tag{5}$$

Having made the step from x_i to x_{i+1}, we might reasonably require that the new approximation H_{i+1} satisfies (5) as if it were actually A^{-1}; that is,

$$x_{i+1} - x_i = -H_{i+1}(\nabla f(x_{i+1}) - \nabla f(x_i)). \tag{6}$$

We might also assume that the updating formula for matrix H_i should be of the form $H_{i+1} = H_i + U$, where U is a correction. It is possible to obtain different updating formulas for U and thus for H_{i+1}, keeping H_{i+1} positive definite ($H_{i+1} > 0$). In fact, there exists a whole family of updates, the Broyden family. From practical experience, the so-called BFGS method seem to be the most popular (see, e.g., Gill et al. 1981 for details). Steps are listed in Algorithm 1.

Function `VarMetric`(x);
let $x \in \mathbb{R}^n$ be an initial solution;
$H \leftarrow I$; $g \leftarrow -\nabla f(x)$;
for $i = 1$ `to` n **do**
  $\alpha^* \leftarrow \arg\min_\alpha f(x + \alpha \cdot Hg)$;
  $x \leftarrow x + \alpha^* \cdot Hg$;
  $g \leftarrow -\nabla f(x)$;
  $H \leftarrow H + U$;
end

Algorithm 1: Variable metric algorithm

Function `BestImprovement`(x);
repeat
  $x' \leftarrow x$;
  $x \leftarrow \arg\min_{y \in N(x)} f(y)$
until $(f(x) \geq f(x'))$;

Algorithm 2: Best improvement (steepest descent) heuristic

From the above one can conclude that even in solving a convex program, a change of metric, and, thus, a change of the neighborhoods induced by this metric, may produce more efficient algorithms. Thus, using the idea of neighbourhood change for solving NP-hard problems could well lead to even greater benefits.

2.2 Local search

A *local search* heuristic consists in choosing an initial solution x, finding a direction of descent from x, within a neighbourhood $N(x)$, and moving to the minimum of $f(x)$ within $N(x)$ in the same direction. If there is no direction of descent, the heuristic stops; otherwise, it is iterated. Usually the steepest direction of descent, also referred to as *best improvement*, is used. This set of rules is summarized in Algorithm 2, where we assume that an initial solution x is given. The output consists of a local minimum, also denoted by x, and its value. Observe that a neighborhood structure $N(x)$ is defined for all $x \in X$. In discrete optimization problems it usually consists of all vectors obtained from x by some simple modification, e.g., in the case of 0–1 optimization, complementing one or two components of a 0–1 vector. Then, at each step, the neighbourhood $N(x)$ of x is explored completely. As this may be time-consuming, an alternative is to use the *first descent* heuristic. Vectors $x_i \in N(x)$ are then enumerated systematically and a move is made as soon as a direction for the descent is found. This is summarized in Algorithm 3.

3 Basic schemes

Let us denote with $\mathcal{N}_k$, ($k = 1, \ldots, k_{\max}$), a finite set of pre-selected neighbourhood structures, and with $\mathcal{N}_k(x)$ the set of solutions in the kth neighbourhood of x. We will also use the notation $\mathcal{N}'_k$, $k = 1, \ldots, k'_{\max}$, when describing local descent. Neighborhoods $\mathcal{N}_k$ or $\mathcal{N}'_k$ may

```
Function FirstImprovement(x);
repeat
    x' ← x; i ← 0;
    repeat
        i ← i + 1;
        x ← arg min{f(x), f(x_i)}, x_i ∈ N(x)
    until (f(x) < f(x_i) or i = |N(x)|);
until (f(x) ≥ f(x'));
```

Algorithm 3: First improvement heuristic

```
Function NeighbourhoodChange(x, x', k);
1 if f(x') < f(x) then
2     x ← x'; k ← 1 /* Make a move */;
  else
3     k ← k + 1 /* Next neighborhood */;
  end
```

Algorithm 4: Neighbourhood change or move or not function

be induced from one or more metric (or quasi-metric) functions introduced into a solution space S. An *optimal solution* x_{opt} (or global minimum) is a feasible solution where a minimum of problem (1) is reached. We call $x' \in X$ a *local minimum* of problem (1) with respect to $\mathcal{N}_k$ (w.r.t. $\mathcal{N}_k$ for short), if there is no solution $x \in \mathcal{N}_k(x') \subseteq X$ such that $f(x) < f(x')$.

In order to solve problem (1) by using several neighbourhoods, facts 1 to 3 can be used in three different ways: (i) deterministic; (ii) stochastic; (iii) both deterministic and stochastic. We first give in Algorithm 4 the steps of the neighbourhood change function which will be used later.

Function `NeighbourhoodChange()` compares the new value $f(x')$ with the incumbent value $f(x)$ obtained in the neighbourhood k (line 1). If an improvement is obtained, k is returned to its initial value and the new incumbent updated (line 2). Otherwise, the next neighbourhood is considered (line 3).

3.1 Variable Neighbourhood Descent (VND)

The *Variable Neighbourhood Descent* (VND) method is obtained if the change of neighbourhoods is performed in a deterministic way. Its steps are presented in Algorithm 5. In the descriptions of all algorithms that follow, we assume that an initial solution x is given. Most local search heuristics in their descent phase use very few neighbourhoods (usually one or two, i.e., $k'_{\text{Max}} \leq 2$). Note that the final solution should be a local minimum with respect to all $k'_{\max}$ neighbourhoods; hence the chances to reach a global one are larger when using VND than with a single neighbourhood structure. Moreover, this *sequential* order of neighbourhood structures in VND above, one can develop a *nested* strategy. Assume, for example, that $k'_{\max} = 3$. Then a possible nested strategy is: perform VND above for the first two neighbourhoods, in each point x' that belongs to the third ($x' \in \mathcal{N}_3(x)$). Such an approach is applied, e.g., in Brimberg et al. (2000), Hansen and Mladenović (2001b).

Function VND(x, k'_{max});
repeat
  $k \leftarrow 1$;
  repeat
    $x' \leftarrow \arg\min_{y \in \mathcal{N}'_k(x)} f(x)$ /* Find the best neighbor in $\mathcal{N}_k(x)$ */;
    NeighbourhoodChange(x, x', k) /* Change neighbourhood */;
  until $k = k'_{max}$;
until *no improvement is obtained*;

Algorithm 5: Steps of the basic VND

Function RVNS(x, k_{max}, t_{max});
1 **repeat**
2   $k \leftarrow 1$;
3   **repeat**
4     $x' \leftarrow$ Shake(x, k);
5   NeighborhoodChange(x, x', k);
  until $k = k_{max}$;
6   $t \leftarrow$ CpuTime()
until $t > t_{max}$;

Algorithm 6: Steps of the reduced VNS

3.2 Reduced VNS

The *Reduced* VNS (RVNS) method is obtained if random points are selected from $\mathcal{N}_k(x)$ and no descent is made. Rather, the values of these new points are compared with that of the incumbent and updating takes place in case of improvement. We assume that a stopping condition has been chosen, among various possibilities, e.g., the maximum CPU time allowed t_{max}, or the maximum number of iterations between two improvements. To simplify the description of the algorithms we always use t_{max} below. Therefore, RVNS uses two parameters: t_{max} and k_{max}. Its steps are presented in Algorithm 6. With the function Shake represented in line 4, we generate a point x' at random from the kth neighbourhood of x, i.e., $x' \in \mathcal{N}_k(x)$.

RVNS is useful in very large instances, for which local search is costly. It has been observed that the best value for the parameter k_{max} is often 2. In addition, the maximum number of iterations between two improvements is usually used as a stopping condition. RVNS is akin to a Monte-Carlo method, but is more systematic (see, for example, Mladenović et al. 2003b where the results obtained by RVNS were 30 continuous min-max problem). When applied to the p-Median problem, RVNS gave solutions as good as the *Fast Interchange* heuristic of Whitaker (1983) while being 20 to 40 times faster (Hansen et al. 2001).

3.3 Basic VNS

The *Basic VNS* (BVNS) method (Mladenović and Hansen 1997) combines deterministic and stochastic changes of neighbourhood. Its steps are given in Algorithm 7.

Often successive neighbourhoods $\mathcal{N}_k$ will be nested. Observe that point x' is generated at random in Step 4 in order to avoid cycling, which might occur if a deterministic rule

```
   Function VNS(x, k_max, t_max);
1  repeat
2     k ← 1;
3     repeat
4        x' ← Shake(x, k)                /* Shaking */;
5        x'' ← FirstImprovement(x')      /* Local search */;
6        NeighbourhoodChange(x, x'', k)  /* Change neighbourhood */;
      until k = k_max;
7     t ← CpuTime()
   until t > t_max;
```

Algorithm 7: Steps of the basic VNS

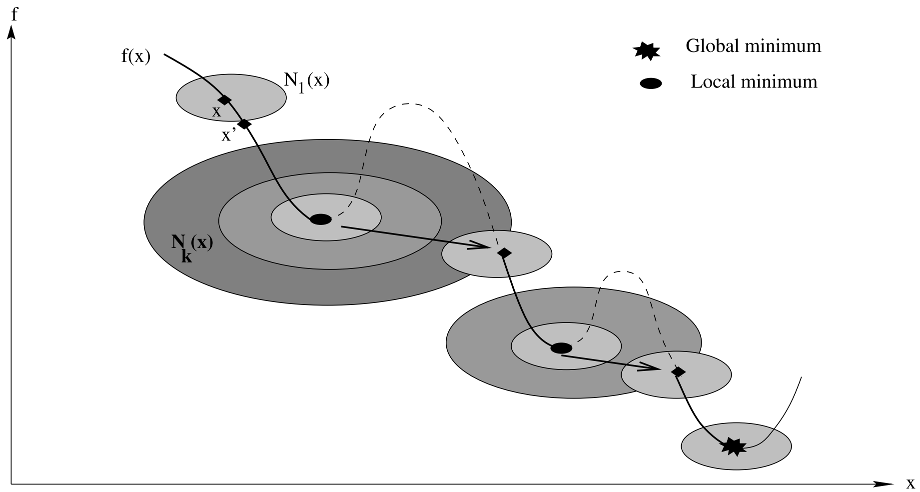


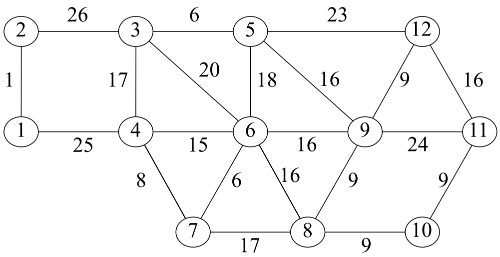
Fig. 1 Basic VNS

were applied. In Step 5 the first improvement local search (Algorithm 3) is usually adopted. However, it can be replaced with best improvement (Algorithm 2).

Example We illustrate the basic step on a minimum k-cardinality tree instance taken from Jornsten and Lokketangen (1997) (see Fig. 2). The minimum k-cardinality tree problem on graph G (k-card for short) consists in finding a subtree of G with exactly k edges whose sum of weights is minimum.

The steps of BVNS are given in Fig. 3. In Step 0 the objective function value, i.e., the sum of edge weights, is equal to 40; it is indicated in the right-hand bottom corner of the figure. This first solution is a local minimum with respect to the edge-exchange neighbourhood structure (one edge in, one out). After shaking, the objective function is 60, and after another local search, we return to the same solution. Then, in Step 3, we take out 2 edges and add another 2 at random, and, after a local search, an improved solution is obtained with a value of 39, etc. In Step 8, we find the optimal solution with an objective function value equal to 36.

Fig. 2 4-cardinality tree problem



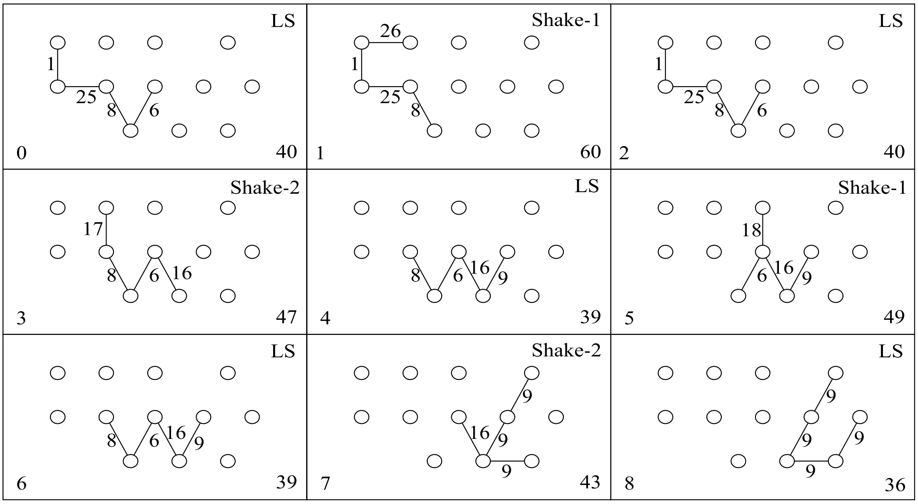


Fig. 3 Steps of the basic VNS for solving 4-card tree problem

3.4 General VNS

Note that the Local search Step 5 may also be replaced by VND (Algorithm 5). Using this general VNS (VNS/VND) approach has led to the most successful applications reported (see, for example, Andreatta and Ribeiro 2002; Brimberg et al. 2000; Canuto et al. 2001; Caporossi and Hansen 2000, 2004; Caporossi et al. 1999a, 1999c; Hansen and Mladenović 2001b; Hansen et al. 2006; Ribeiro and de Souza 2002; Ribeiro et al. 2002). Steps of the general VNS (GVNS) are given in Algorithm 8 below.

3.5 Skewed VNS

The skewed VNS (SVNS) method (Hansen et al. 2000) addresses the problem of exploring valleys far from the incumbent solution. Indeed, once the best solution in a large region has been found, it is necessary to go some way to obtain an improved one. Solutions drawn at random in distant neighborhoods may differ substantially from the incumbent and VNS can then degenerate, to some extent, into the Multistart heuristic (in which descents are made iteratively from solutions generated at random, a heuristic which is known not to be very efficient). Consequently, some compensation for distance from the incumbent must be made and a scheme called Skewed VNS is proposed for this purpose. Its steps are presented in

Function `GVNS`$(x, k'_{max}, k_{max}, t_{max})$;
repeat
    $k \leftarrow 1$;
    repeat
        $x' \leftarrow$ `Shake`(x, k);
        $x'' \leftarrow$ `VND`(x', k'_{max});
        `NeighborhoodChange`(x, x'', k);
    until $k = k_{max}$;
    $t \leftarrow$ `CpuTime()`
until $t > t_{max}$;

Algorithm 8: Steps of the general VNS

Function `NeighbourhoodChangeS`(x, x'', k, α);
if $f(x'') - \alpha\rho(x, x'') < f(x)$ **then**
    $x \leftarrow x''$; $k \leftarrow 1$
else
    $k \leftarrow k + 1$
end

Algorithm 9: Steps of neighbourhood change for the skewed VNS

Function `SVNS`$(x, k_{max}, t_{max}, \alpha)$;
repeat
    $k \leftarrow 1$; $x_{best} \leftarrow x$;
    repeat
        $x' \leftarrow$ `Shake`(x, k);
        $x'' \leftarrow$ `FirstImprovement`(x');
        `KeepBest`(x_{best}, x);
        `NeighbourhoodChangeS`(x, x'', k, α);
    until $k = k_{max}$;
    $x \leftarrow x_{best}$;
    $t \leftarrow$ `CpuTime()`;
until $t > t_{max}$;

Algorithm 10: Steps of the Skewed VNS

Algorithms 10 and 11, where the `KeepBest`(x, x') function simply keeps whichever is the better of x and x': **if** $f(x') < f(x)$ **then** $x \leftarrow x'$.

SVNS makes use of a function $\rho(x, x'')$ to measure the distance between the incumbent solution x and the local optimum found x''. The distance used to define the $\mathcal{N}_k$, as in the above examples, could be used also for this purpose. The parameter α must be chosen in order to accept the exploration of valleys far away from x when $f(x'')$ is larger than $f(x)$ but not too much larger (otherwise one will always leave x). A good value is to be found experimentally in each case. Moreover, in order to avoid frequent moves from x to a close solution,

Function BI-VNS($x, k_{\max}, t_{\max}$);
repeat
  $k \leftarrow 1$;
  $x_{\text{best}} \leftarrow x$;
  repeat
    $x' \leftarrow$ Shake(x, k);
    $x'' \leftarrow$ FirstImprovement(x');
    KeepBest(x_{best}, x'');
    $k \leftarrow k + 1$;
  until $k = k_{\max}$;
  $x \leftarrow x_{\text{best}}$;
  $t \leftarrow$ CpuTime()
until $t > t_{\max}$;

Algorithm 11: Steps of the basic best improvement VNS

one may take a large value for α when $\rho(x, x'')$ is small. More sophisticated choices for a function of $\alpha\rho(x, x'')$ could be made through a learning process.

3.6 Some extensions of basic VNS

Several easy ways to extend the basic VNS are now discussed. The basic VNS is a first improvement descent method with randomization. Without much additional effort it can be transformed into a descent-ascent method: in NeighbourhoodChange() function, replace also x by x'' with some probability, even if the solution is worse than the incumbent. It can also be changed into a best improvement method: make a move to the best neighbourhood k^* among all $k_{\max}$ of them. Its steps are given in Algorithm 11.

Another variant of the basic VNS can be to find a solution x' in the "Shaking" step as the best among b (a parameter) randomly generated solutions from the kth neighborhood. There are two possible variants of this extension: (i) to perform only one local search from the best among b points; (ii) to perform all b local searches and then choose the best. We now give an algorithm of a second type suggested by Fleszar and Hindi (2004). There, the value of parameter b is set to k. In this way, no new parameter is introduced (see Algorithm 12). It is also possible to introduce $k_{\min}$ and k_{step}, two parameters which control the change of neighbourhood process: in the previous algorithms instead of $k \leftarrow 1$ set $k \leftarrow k_{\min}$ and instead of $k \leftarrow k + 1$ set $k \leftarrow k + k_{\text{step}}$. The steps of Jump VNS are given in Algorithms 13 and 14.

3.7 Variable neighbourhood decomposition search

While the basic VNS is clearly useful for obtaining an approximate solution to many combinatorial and global optimization problems, it remains a difficult or lengthy take to solve very large instances. As often, the size of the problems considered is in practice more limited by the tools available to solve them than by the needs of the potential users of these tools. Hence, improvements appear to be highly desirable. Moreover, when heuristics are applied to very large instances, their strengths and weaknesses become clearly apparent. Three improvements of the basic VNS for solving large instances are now considered.

The variable neighbourhood decomposition search (VNDS) method (Hansen et al. 2001) extends the basic VNS into a two-level VNS scheme based upon decomposition of the prob-

Function FH-VNS($x, k_{\max}, t_{\max}$);
repeat
    $k \leftarrow 1$;
    repeat
        for $\ell = 1$ to k **do**
            $x' \leftarrow$ Shake(x, k);
            $x'' \leftarrow$ FirstImprovement(x');
            KeepBest(x, x'');
        end
        NeighbourhoodChange(x, x'', k);
    until $k = k_{\max}$;
    $t \leftarrow$ CpuTime()
until $t > t_{\max}$;

Algorithm 12: Steps of the Fleszar-Hindi extension of the basic VNS

Function J-VNS($x, k_{min}, k_{step}, k_{\max}, t_{\max}$);
repeat
    $k \leftarrow k_{\min}$;
    repeat
        $x' \leftarrow$ Shake(x, k);
        $x'' \leftarrow$ FirstImprovement(x');
        NeighbourhoodChangeJ($x, x'', k, k_{\min}, k_{\text{step}}$);
    until $k = k_{\max}$;
    $t \leftarrow$ CpuTime()
until $t > t_{\max}$;

Algorithm 13: Steps of the Jump VNS

Function NeighborhoodChangeJ($x, x', k, k_{\min}, k_{\text{step}}$);
if $f(x') < f(x)$ **then**
    $x \leftarrow x'$; $k \leftarrow k_{\min}$;
else
    $k \leftarrow k + k_{\text{step}}$;
end

Algorithm 14: Neighbourhood change or move or not function

lem. Its steps are presented in Algorithm 15, where t_d is an additional parameter and represents the running time given for solving decomposed (smaller sized) problems by VNS.

For ease of presentation, but without loss of generality, we assume that the solution x represents the set of some elements. In Step 4 we denote with y a set of k solution attributes present in x' but not in x ($y = x' \setminus x$). In Step 5 we find the local optimum y' in the space of y; then we denote with x'' the corresponding solution in the whole space S ($x'' = (x' \setminus y) \cup y'$). We notice that exploiting some *boundary effects* in a new solution can significantly improve

```
    Function VNDS(x, k_max, t_max, t_d);
1   repeat
2       k ← 2;
3       repeat
4           x' ← Shake(x, k); y ← x' \ x;
5           y' ← VNS(y, k, t_d); x'' = (x' \ y) ∪ y';
6           x''' ← FirstImprovement(x'');
7           NeighborhoodChange(x, x''', k);
        until k = k_max;
    until t > t_max;
```

Algorithm 15: Steps of VNDS

the solution quality. This is why, in Step 6, we find the local optimum x''' in the whole space S using x'' as an initial solution. If this becomes time-consuming, then at least a few local search iterations should be performed.

VNDS can be viewed as embedding the classical successive approximation scheme (which has been used in combinatorial optimization at least since the 1960s; see, for example, Griffith and Stewart (1961) in the VNS framework.

3.8 Parallel VNS

Parallel VNS (PVNS) methods are another extension. Several ways of parallelizing VNS have recently been proposed for solving the p-Median problem. In García-López et al. (2002) three of them are tested: (i) parallelize local search; (ii) augment the number of solutions drawn from the current neighbourhood and make a local search in parallel from each of them and (iii) do the same as (ii) but update the information about the best solution found. The second version gives the best results. It is shown in Crainic et al. (2004) that assigning different neighbourhoods to each processor and interrupting their work as soon as an improved solution is found gives very good results. The best-known solutions have been found on several large instances taken from TSP-LIB Reinelt (1991). Three Parallel VNS strategies are also suggested for solving the Travelling purchaser problem in Ochi et al. (2001). See Moreno-Pérez et al. (2005) for details.

3.9 Primal-dual VNS

For most modern heuristics, the difference in value between the optimal solution and the obtained one is completely unknown. Guaranteed performance of the primal heuristic may be determined if a lower bound on the objective function value is known. To this end, the standard approach is to relax the integrality condition on the primal variables, based on a mathematical programming formulation of the problem. However, when the dimension of the problem is large, even the relaxed problem may be impossible to solve exactly by standard commercial solvers. Therefore, it seems a good idea to solve dual relaxed problems heuristically as well. In this way we obtain guaranteed bounds on the primal heuristics performance. The next problem arises if we want to reach an exact solution within a Branch and bound framework, since having the approximate value of the relaxed dual does not allow us to branch easily, e.g., by exploiting complementary slackness conditions. Thus, the exact value of the dual is necessary.

Function `PD-VNS`($x, k'_{\max}, k_{\max}, t_{\max}$);
1 `BVNS`($x, k'_{\max}, k_{\max}, t_{\max}$) /* Solve primal by VNS */;
2 `DualFeasible`(x, y) /* Find (infeasible) dual such that $f_P = f_D$ */;
3 `DualVNS`(y) /* Use VNS do decrease infeasibility */;
4 `DualExact`(y) /* Find exact (relaxed) dual */;
5 `BandB`(x, y) /* Apply branch-and-bound method */;

Algorithm 16: Steps of the basic PD-VNS

In Primal-dual VNS (PD-VNS) (Hansen et al. 2007a) one possible general way to attain both the guaranteed bounds and the exact solution is proposed. Its steps are given in Algorithm 16.

In the first stage an heuristic procedure based on VNS is used to obtain a near optimal solution. In Hansen et al. (2007a) it is shown that VNS with decomposition is a very powerful technique for large-scale simple plant location problems (SPLP) with up to 15 000 facilities and 15 000 users. In the second phase, this approach is designed to find an exact solution to the relaxed dual problem. Solving SPLP is accomplished in three stages: (i) find an initial dual solution (generally infeasible), using the primal heuristic solution and complementary slackness conditions; (ii) improve the solution by applying VNS to the unconstrained nonlinear form of the dual; (iii) solve the dual exactly using a customized "sliding simplex" algorithm which applies "windows" to the dual variables, substantially reducing the size of the problem. In all the problems tested, including instances much larger than previously reported in the literature, the procedure was able to find the exact dual solution in reasonable computing time. In the third and final phase armed with tight upper and lower bounds, obtained respectively from the heuristic primal solution in phase one and the exact dual solution in phase two, we apply a standard branch-and-bound algorithm to find an optimal solution of the original problem. The lower bounds are updated with the dual sliding simplex method and the upper bounds, whenever new integer solutions are obtained at the nodes of the branching tree. In this way it is possible to solve exactly problem instances with up to $7\,000 \times 7\,000$ for uniform fixed costs and $15\,000 \times 15\,000$ otherwise.

3.10 Variable neighborhood formulation space search

Traditional ways to tackle an optimization problem consider a given formulation and search in some way through its feasible set X. The fact that the same problem may often be formulated in different ways allows search paradigms to be extended to include jumps from one formulation to another. Each formulation should lend itself to some traditional search method, its 'local search' which works totally within this formulation, and yields a final solution when started from some initial solution. Any solution found in one formulation should easily be translatable to its equivalent in any other formulation. We may then move from one formulation to another, using the solution resulting from the former's local search as an initial solution for the latter's local search. Such a strategy will, of course, be useful only in situations where local searches in different formulations behave differently.

This idea was recently investigated in Mladenović et al. (2005) using an approach which systematically changes the formulations for solving circle packing problems (CPP). It is shown there that the stationary point of a non-linear programming formulation of CPP in Cartesian coordinates is not necessarily also a stationary point in a polar coordinate system. A method *Reformulation Descent* (RD) is suggested which alternates between these two

Function `FormulationChange`$(x, x', \phi, \phi', \ell)$;
if $f(\phi', x') < f(\phi, x)$ **then**
  $\phi \leftarrow \phi'$; $x \leftarrow x'$; $\ell \leftarrow \ell_{\min}$
else
  $\ell \leftarrow \ell + \ell_{\text{step}}$;
end

Algorithm 17: Formulation change function

Function `VNFSS`$(x, \phi, \ell_{\max})$;
repeat
  $\ell \leftarrow 1$ /* Initialize formulation in $\mathcal{F}$ */;
  while $\ell \le \ell_{\max}$ **do**
    `ShakeFormulation`(x,x',ϕ,ϕ',ℓ) /* $(\phi',x') \in (N_\ell(\phi),\mathcal{N}(x))$ at random */;
    `FormulationChange`(x,x',ϕ,ϕ',ℓ) /* Change formulation */;
  end
until *some stopping condition is met*;

Algorithm 18: Reduced variable neighborhood FSS

formulations until the final solution is stationary with respect to both. The results obtained were comparable with the best known values, but they were achieved some 150 times faster than by an alternative single formulation approach. In the same paper, the idea suggested above of *Formulation space search* (FSS) is also introduced, using more than two formulations. Some research in this direction has been reported in Mladenović (2005), Plastria et al. (2005), Hertz et al. (2008). One algorithm which uses the variable neighborhood idea in searching through the formulation space is given in Algorithms 17 and 18.

In Fig. 4 we consider the CPP case with $n = 50$. The set consists entirely of mixed formulations, in which some circle centres are given in Cartesian coordinates while the others are given in polar coordinates. The distance between two formulations is then the number of centres whose coordinates are expressed in different systems in each formulation. FSS starts with the RD solution, i.e., with $r_{\text{curr}} = 0.121858$. The values of $k_{\min}$ and k_{step} are set to 3 and the value of $k_{\max}$ is set to $n = 50$. We did not gain any improvement with $k_{\text{curr}} = 3, 6$ and 9. The next improvement was obtained for $k_{\text{curr}} = 12$. This means that a "mixed" formulation with 12 polar and 38 Cartesian coordinates is used. Then we turned again to the formulation with 3 randomly chosen circle centres, which was unsuccessful; but we obtained a better solution with 6, etc. After 11 improvements we ended with a solution with radius $r_{\max} = 0.125798$.

4 Developing VNS

4.1 Getting started

This section is devoted to newcomers. Its purpose is to help students in making a first very simple version of VNS, which would not necessarily be competitive with later more sophisticated versions. Most of the steps are common to the implementation of other metaheuristics.

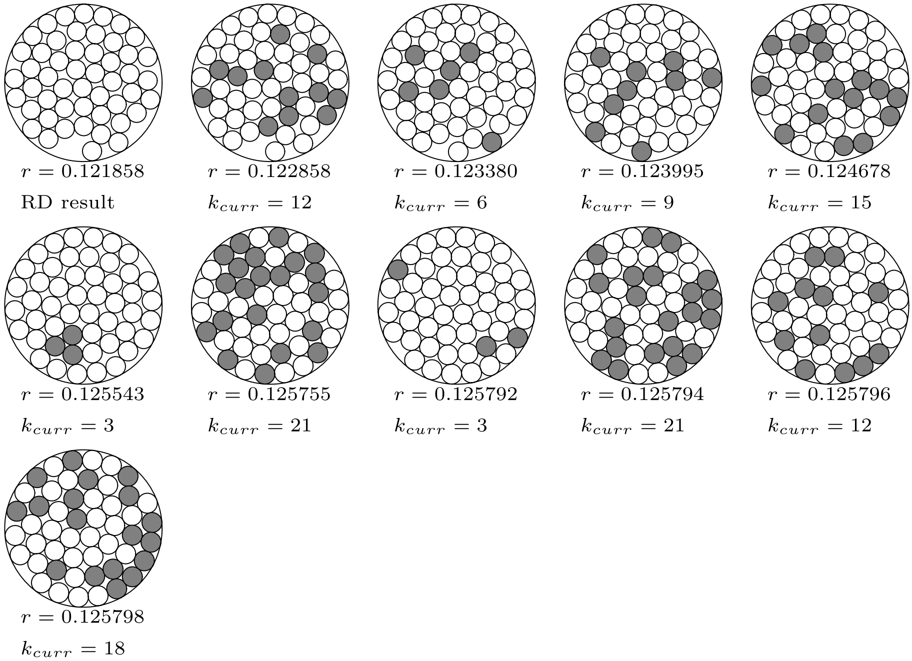


Fig. 4 Reduced FSS for PCC problem and $n = 50$

A step-by-step procedure:

1. **Getting familiar with the problem**. Think about the problem at hand. In order to understand it better, make a simple numerical example and spend some time in trying to solve it by hand in your own way. Try to understand why the problem is hard and why a heuristic is needed.
2. **Read Literature**. Read about the problem and the solution methods in the literature.
3. **Test instances (read data)**. Use your numerical example as a first instance for testing your future code, but if it is not large enough, take some instance from the web, or make a routine for generating random instances. In the second case, read how to generate events using uniformly distributed numbers from the (0, 1) interval (since each programming language has statements for getting such random numbers).
4. **Data structure**. Think about the way in which the solution of the problem will be represented in the memory. Consider two or more representations of the same solution to see if they can reduce the complexity of some routines, i.e., analyze the advantages and disadvantages of each possible presentation.
5. **Initial solution**. Having established a routine for reading or generating instances of the problem, the next step is to obtain an initial solution. For the simple version, any random feasible solution may be used, but the usual way is to develop some *greedy* constructive heuristic. This is normally not hard to do.
6. **Objective value**. Make a procedure for calculating the objective function value for a given solution. Notice that at this stage, we already have all ingredients for a Monte-Carlo method: the generation of a random solution and calculation of the objective function value. Get a solution to your problem by the Monte Carlo heuristic (i.e., repeat steps 5 and 6 many times and keep the best one).

7. **Shaking**. Make a procedure for Shaking. This is a key step of VNS. However, it is easy to implement and usually requires only a few lines of computer code. For example, in solving the multi-source Weber problem, the easiest perturbation of the current solution is to re-allocate a randomly chosen entity ℓ from its cluster to another one, also chosen at random. In fact, in this case, the shaking step (or jump, in the kth neighbourhood) would need only three lines of the computer code:

   For $i = 1$ to k
   $\quad a(1 + n \cdot \text{Rnd}\,1) = 1 + m \cdot \text{Rnd}\,2$
   EndFor

   This solution is saved in an array $a(\ell) \in \{1, \dots, m\}$ which denotes the membership or allocation of entity ℓ ($\ell = 1, \dots, n$); Rnd 1 and Rnd 2 denote random numbers uniformly distributed from the (0,1) interval. Compare the results obtained by the Reduced VNS (take $k_{\max} = 2$) with those of the Monte Carlo method.
8. **Local search**. Choose an off-the-shelf local search heuristic (or develop a new one). In building a new local search, consider several usual moves which define the neighbourhood of the solution, such as *drop, add, swap, interchange*, etc. In addition, for the efficiency (speed) of the method, it is very important to pay special attention to *updating* of the incumbent solution. In other words, it is usually not necessary to use a procedure for calculating the objective function values for each point in the neighbourhood, i.e., it is often possible to reach these values by very simple calculations.
9. **Comparison**. Include the local search routine into RVNS to get the basic VNS, and compare it with other methods from the literature.

4.2 More tips

Sometimes the basic VNS does not provide very good results and it must then be refined in one of the following ways.

1. **First vs. best improvement**. Experimentally compare *first* and *best improvement* strategies within a local search. Previous experience suggest the following: if your initial solution is chosen at random, use the first improvement rule, but if some constructive heuristic is used, use the best improvement rule.
2. **Reduce the neighbourhood**. The reason for the bad behaviour of any local search may be unnecessary visits to all the solutions in the neighbourhood. Try to identify "promising" subsets of the neighbourhood and visit these only; ideally, find a rule which automatically removes solutions from the neighborhood solutions whose objective values are no better than the current one.
3. **Intensified shaking**. In developing a more effective VNS, one must spend some time in checking how sensitive is the objective function to small change (shaking) of the solution. The trade-off between intensification and diversification of the search in VNS is balanced in a Shaking procedure. For some problem instances, a completely random jump in the kth neighborhood is too diversified. In such cases, some *intensify shaking* procedure is in order. For instance, a k-interchange neighbourhood may be reduced by repeating k times *random add* followed by *best drop* moves. A special case of intensified shaking is the so-called *Large neighbourhood search*, where k randomly chosen attributes of the solutions are destroyed (dropped), and then the solution is re-built in the best way (by some constructive heuristic).
4. **VND**. Analyze several possible neighbourhood structures, estimate their sizes, order them, try them out and keep the most efficient ones. In other words, develop a VND and replace the local search routine with this VND to get a general VNS.

5. **Experiments with parameter settings**. The single central parameter of VNS is $k_{\max}$, which should be tuned experimentally. However, the procedure is not usually very sensitive to $k_{\max}$ and, in order to obtain a parameter-free VNS, one can fix its value at the value of some input parameter, e.g., for the p-median, $k_{\max} = p$; for the minimum sum-of-square clustering $k_{\max} = m$, etc.

5 Applications

Applications of VNS, or of hybrids of VNS combined with other metaheuristics, are diverse and numerous. In this section, we review some of them.

5.1 Industrial applications

Regarding the first industrial applications, the oil industry has provided many problems. These include the design of an offshore pipeline network (Brimberg et al. 2003), the pooling problem (Audet et al. 2004) and the scheduling of walkover rigs for Petrobras (Aloise et al. 2006).

5.2 Design problems in communication

Costa et al. (2002) apply a variable neighbourhood decomposition search (VNDS) for the optimization of a power plant cable layout. Mladenović et al. (2003b) use VNS for solving a spread spectrum radar polyphase code design problem. Degila and Sansò (2004) propose a VNS to deal with the topological design of a yotta-bit-per-second (1 yotta $= 10^{24}$) multidimensional network based upon agile optical cores which provides fully meshed connectivity with direct optical paths between edge nodes which are electronically controlled. Lapierre et al. (2004) consider the application of a Tabu Search/VNS hybrid for designing distribution networks with transshipment centres. Meric et al. (2004) apply VNS for optical routing in networks using latin routers. Dias et al. (2006) use a General VNS (GVNS) to improve the quality of the solution obtained with a Greedy Randomized Adaptive Search Procedure (GRASP) for the ring star problem. In Loudni et al. (2006) a difficult real-life network problem of France Telecom R&D, the on-line resources allocation for ATM networks with rerouting is solved by VNS/LDS+CP.

The application of VNS in the design of SDH/WDM networks is proposed in Melián et al. (2008); it is improved with the use of an adaptive memory mechanism in Melián (2006) and by applying a pilot method in Höller et al. (2008). Tagawa et al. (2007) deal with the robust design of Surface Acoustic Wave (SAW) filters. Ribeiro et al. (2007) consider VNS and other metaheuristics for optimization problems in computer communications.

5.3 Location problems

Location problems have also attracted much attention from the VNS researchers and practitioners. Among discrete models the p-median has been the most studied and has played a central rule in the development of a VNS metaheuristic. Brimberg and Mladenović (1996) give the earliest applications of VNS. Hansen et al. (2001) introduces a variable neighbourhood decomposition search solving the p-median problem. García-López et al. (2002) is the first parallel version of the VNS. Hansen and Mladenović (2008) complete the comparative analysis in Alba and Domínguez (2006) with a detailed comparison of several versions of

VNS with other metaheuristics for the p-median problem. See Mladenović et al. (2007a) for the role of VNS in solving the p-median problem.

Other discrete location problems solved with VNS are the p-centre problem (Mladenović et al. 2003a), the maximum capture problem (Benati and Hansen 2002) and several variants of the p-median problem. Domínguez-Marín et al. (2005) deal with solving the discrete ordered median problem, Fathali and Kakhki (2006) apply VNS to the p-median problem with pos/neg weights, Fleszar and Hindi (2008) solve the capacitated p-median problem and Pérez et al. (2007) propose a hybrid which combines VNS with Path Relinking for the p-hub median problem. Osman and Ahmadi (2007) investigate different search and selection strategies, including the variable neighbourhood descent (VND) for the capacitated p-median problem with single source constraint. Moreno-Pérez et al. (2003) propose a variable neighbourhood tabu search hybrid and consider its application to the median cycle problem.

Among continuous models, the multi-source Weber problem is first addressed in Brimberg et al. (2000) and in Brimberg et al. (2004) with constant opening costs. Brimberg et al. (2006a) use VNS in a decomposition strategy for large-scale instances. Brimberg et al. (2008a) apply VNS to the maximum return-on-investment plant location problem with market share. Ljubic (2007) proposes a hybrid VNS for a connected facility location problem which combines the facility location problem and the Steiner tree problem in graphs. Hansen et al. (2007a) apply a primal-dual VNS for the simple plant location problem. Finally, Bischoff and Dächert (2009) use VNS and other heuristics for a generalized class of continuous location-allocation problems and Jabalameli and Ghaderi (2008) propose hybrid algorithms which combine Genetic Algorithm (GA) and VNS for the uncapacitated continuous location-allocation problem.

Drezner et al. (2005) analyse the difficulty in the instances of quadratic assignment problems for metaheuristic approaches and Zhang et al. (2005) use a VNS with permutation distance. Han et al. (2007) use a hybrid of VNS with Ant Colony Optimization and Liu and Abraham (2007) a fuzzy hybrid of VNS with Particle Swarm Optimization (PSO) method. Geiger and Wenger (2009) solve a practical assignment problem in higher education using a VNS approach. Mitrovic-Minic and Punnen (2009) propose a very large-scale VNS for the Multi-Resource Generalized Assignment Problem.

Yang et al. (2007) apply optimization strategies based on Simulated Annealing and VNS for the base station location problem in a WCDMA (Wideband Code-Division Multiple Access) network. Pacheco et al. (2008) use VNS to solve the classical maximum covering location problem for locating health resources. Wollenweber (2008) uses several hybrids with VNS for a multi-stage facility location problem with staircase costs and splitting of commodities.

5.4 Data mining

VNS proved to be a very efficient tool in cluster analysis. In particular, the J-Means heuristic combined with VNS appears to be state-of-the-art for the heuristic solution of minimum sum-of-square clustering (Hansen and Mladenović 2001b; Belacel et al. 2002, 2004a). Combined with stabilized column generation (du Merle et al. 1999) it leads to the most efficient exact algorithm at present for this problem (du Merle et al. 2000). Such an approach has also been applied by Hansen and Perron (2007) to solve the $\mathcal{L}_1$ embeddability problem for data sets. Brusco et al. (2009) use a VNS to select variables in Principal Component Analysis.

Belacel et al. (2004b) use VNS Metaheuristic for Fuzzy Clustering cDNA Microarray Gene Expression Data. Negreiros and Palhano (2006) propose a constructive procedure followed by a VNS to solve the capacitated centred clustering problem. Brusco and Steinley

(2007a) compare a VNS method with the classical k-means for the clustering of two-mode proximity binary matrices and Brusco and Steinley (2007b) compare heuristic procedures for Minimum Within-Cluster Sums of Squares Partitioning. Benati (2008) applies VNS to categorical data fuzzy clustering. Other clustering problem applications appear in Brusco et al. (2008). Hansen et al. (2009) use a primal-dual VNS to solve large p-median clustering problems.

Another important data mining task which has been managed with VNS is classification. Pacheco et al. (2007) use VNS in the variable selection and determination of the linear discrimination function coefficients. Karam et al. (2007) perform arbitrary-norm hyperplane separation by VNS. The same problem has also been attacked with VNS in Plastria et al. (2009). Hansen et al. (2007b) apply VNS for colour image quantization. Belacel et al. (2007) propose a VNS heuristic for learning the parameters of the multiple criteria classification method PROAFTN from data. Carrizosa et al. (2007) use VNS for the selection of the Globally Optimal Prototype Subset for Nearest-Neighbour Classification. Plastria et al. (2009) describe two local descent methods that are embedded into a VNS scheme to solve a linear classification problem. A specific clustering VNS algorithm is proposed in design of balanced MBA student teams in Desrosiers et al. (2005).

5.5 Graph problems

In addition to some design problems in communications and most of the location problems, VNS has been applied to other combinatorial optimization problems on graphs. A VNS is proposed for the max-cut problem in a graph and compared with other metaheuristics in Festa et al. (2002) and an hybridization between a memetic algorithm and VNS is proposed for the same problem by Duarte et al. (2005). Moreno-Pérez et al. (2003) propose a variable neighbourhood tabu search (VNTS) hybrid for the median cycle problem. Hansen et al. (2004) propose and test a basic VNS which combines greedy with the simplicial vertex test in its descent step for the maximum clique problem. For the graph colouring problem, Avanthay et al. (2003) propose an adaptation of the VNS metaheuristic, Galinier and Hertz (2006) present a survey of local search methods which includes VNS and Hertz et al. (2008) analyze the variable space search methodology which extends the Formulation Space Search (FSS). Brimberg et al. (2008b) propose a new heuristic based on VNS for the k-cardinality subgraph problem, in contrast with the constructive heuristics proposed in the literature. Brimberg et al. (2009) use a VNS to solve the heaviest k-subgraph problem. Amaldi et al. (2009) propose a VNS to tackle the minimum fundamental cycle basis problem.

Several graph problems involving trees have also been tackled with VNS. VNS is used in Canuto et al. (2001) as a post-optimization procedure for a multistart local search algorithm for the prize-collecting Steiner tree problem, based on the generation of initial solutions by a primal-dual algorithm using perturbed node prizes. Ribeiro et al. (2002) use a hybrid VNS-GRASP with perturbations for the Steiner problem in graphs. Mladenović and Urošević (2003) propose the use of a VNS for the edge weighted k-cardinality tree problem Urošević et al. (2004) propose a variable neighbourhood decomposition search (VNDS) for the same problem and Brimberg et al. (2006b) for the vertex weighted k-cardinality tree problem. Ribeiro and de Souza (2002) propose the use of VNS for the degree constrained minimum spanning tree problem and de Souza and Martins (2008) use a Skewed VNS enclosing a second order algorithm for the same problem. Hu et al. (2008) propose a VNS approach which uses three different neighbourhood types to solve the generalized minimum spanning tree problem. A VNS is used in Martins and de Souza (2009) to solve the minimum spanning tree problem with minimum degree constraints in all nodes except the leaves. Finally, VNS is used to solve the minimum labelling spanning tree problem in Consoli et al. (2009a, 2009b).

5.6 Knapsack and packing problems

Another important class of problems solved with VNS and its variants and hybrids are the knapsack and packing problems. In Puchinger et al. (2006) a relaxation guided VNS is applied to the multidimensional knapsack problem and to its core problems. The paper by Puchinger and Raidl (2008) constitutes an excellent illustration of a dynamic ordering of the neighborhood structures embedded in a variable neighborhood descent algorithm which is used to solve also the multidimensional knapsack problem. VNS has also been successfully applied to the bin packing problem (Fleszar and Hindi 2002) and to the strip packing problem (Beltrán et al. 2004). Parreño et al. (2008) present a VNS algorithm for the container loading problem.

Circle and sphere packing have also been approached with VNS. Mladenović et al. (2005) introduce the reformulation descent which is applied to circle packing problems and Mladenović et al. (2007b) the formulation space search for the same problems. Kucherenko et al. (2007) use VNS to solve the kissing number problem, i.e., the problem of determining the maximum number of D-dimensional spheres of radius r that can be adjacent to a central sphere of radius r.

5.7 Mixed integer problems

Heuristics may help in finding a feasible solution or an improved and possibly optimal solution to large and difficult mixed integer programs. The local branching method of Fischetti and Lodi (2003) does this, in the spirit of VNS. For further developments see Fischetti et al. (2004) and Hansen et al. (2006). Gutjahr et al. (2007) use the VNS approach for noisy problems and its application to project portfolio analysis.

5.8 Time tabling

Timetabling and related manpower organization problems can be well solved with VNS. Cote et al. (2005) use a simplified variable neighbourhood descent in a hybrid multi-objective evolutionary algorithm for the uncapacitated exam proximity problem. Sevkli and Sevilgen (2006) propose a VNS approach for the orienteering problem and Archetti et al. (2007) propose VNS to solve the team orienteering problem (TOP), that is, the generalization to the case of multiple tours of the orienteering problem, known also as the selective traveling salesman problem. Schilde et al. (2009) use a VNS to solve a bi-objective orienteering problem.

5.9 Scheduling

In recent years also several scheduling problems have been efficiently solved with VNS approaches. They include single machine and parallel machines, multiobjective scheduling, job shop scheduling, flow shop, resource-constrained project scheduling and other scheduling problems.

5.9.1 Single machine scheduling

Gupta and Smith (2006) use a VNS algorithm for single machine total tardiness scheduling with sequence-dependent setups. Lin and Ying (2008) propose a hybrid Tabu-VNS metaheuristic approach for single-machine tardiness problems with sequence-dependent setup times. Liao and Cheng (2007) propose a VNS for minimizing single machine weighted earliness and tardiness with common due date. Tseng et al. (2009) employ a VNS for large-size instances of the single machine total tardiness problem with controllable processing times. Wang and Tang (2009) propose a population-based variable neighbourhood search for the single machine total weighted tardiness problem.

5.9.2 Parallel machine scheduling

Anghinolfi and Paolucci (2007) propose a hybrid metaheuristic approach which integrates several features from tabu search, simulated annealing and VNS for a parallel machine total tardiness scheduling problem. De Paula et al. (2007) apply VNS for solving parallel machines scheduling problems with sequence-dependent setups. Chen and Chen (2009) propose an approach which integrates the principles of the variable neighbourhood descent approach and tabu search for the unrelated parallel-machine scheduling problem with sequence-dependent setup times. Behnamian et al. (2009b) use an ACO, SA and VNS hybrid for parallel machines scheduling problems with sequence-dependent setup times.

5.9.3 Multiobjective scheduling

Gagné et al. (2005) use compromise programming with Tabu-VNS metaheuristic for the solution of multiple-objective scheduling problems. Qian et al. (2006) deal with multi-objective flow shop scheduling, using differential evolution.

5.9.4 Job shop scheduling

Sevkli and Aydin (2006a, 2006b) use VNS algorithms for job shop scheduling problems. Sevkli and Aydin (2007) propose parallel VNS algorithms and Gao et al. (2008) propose a hybrid GA/VND and Pan et al. (2007b) a PSO/VNS hybrid heuristic for these problems. Liu et al. (2006) propose a variable neighborhood particle swarm optimization for multi-objective flexible job-shop scheduling problems. Aydin and Sevkli (2008) consider sequential and parallel VNS algorithms for job shop scheduling. A VNS is applied by Roshanaei et al. (2009) to tackle the job shop scheduling problem with setup times. Zobolas et al. (2009b) propose a hybrid method that combines VNS with Differential Evolution and a Genetic Algorithm to solve the job shop scheduling problem.

5.9.5 Flow shop scheduling

Blazewicz et al. (2005) use VNS for late work minimization in a two-machine flow shop with common due date. In Pan et al. (2007a) VNS and three other metaheuristic approaches are proposed for a no-wait flow shop problem. In Blazewicz et al. (2008) VNS and two other metaheuristics are presented for the two-machine flow shop problem with weighted late work criterion and common due date. Zobolas et al. (2009a) design a GA/VNS hybrid to minimize makespan in permutation flow shop scheduling problems. In Tasgetiren et al. (2004) a simple but very efficient local search, based on VNS, is embedded in the PSO algorithm in order to solve the permutation flow shop sequencing problem. Liao et al. (2007) apply VNS for flow shop scheduling problems and Tasgetiren et al. (2007) consider the makespan and total flow time minimization in the permutation flow shop sequencing problem. Czogalla and Fink (2008) examine the application of a PSO with variable neighbourhood descent as an embedded local search procedure for the continuous flow-shop scheduling problem. Rahimi-Vahed et al. (2009) devise a hybrid multi-objective algorithm based on shuffled frog-leaping algorithm and VNS for a bi-criteria permutation flow shop scheduling problem. Chyu and Chen (2009) propose several VNS for a lump-sum payment model for the resource-constrained project scheduling problem. Behnamian et al. (2009a) combine VNS with simulated annealing in a population based hybrid for a realistic flow shop problem. Jarboui et al. (2009) add a VNS to an estimation of the distribution algorithm for minimizing the total flow time in permutation flow shop scheduling problems. A VNS is used in Rahimi-Vahed et al. (2009) to find Pareto optimal solutions for a permutation flow shop scheduling problem.

5.9.6 Resource-constrained project scheduling

Fleszar and Hindi (2004) propose a solution for the resource-constrained project scheduling problem by a VNS and Kolisch and Hartmann (2006) include VNS in an experimental investigation of heuristics for resource-constrained project scheduling. Bouffard and Ferland (2007) improve simulated annealing with VNS to solve the resource-constrained scheduling problem.

5.9.7 Car sequencing

Prandtstetter and Raidl (2008) use a hybrid VNS for the car sequencing problem and Gavranović (2008) applies VNS to car-sequencing problems with colours. Ribeiro et al. (2008a) propose a set of heuristics based on the paradigms of the VNS and ILS metaheuristics for a multi-objective real-life car sequencing problem with painting and assembly line constraints and Ribeiro et al. (2008b) provide an efficient implementation of the VNS/ILS heuristic for this real-life car sequencing problem. Joly and Frein (2008) use VNS to tackle an industrial car sequencing problem considering paint and assembly shop objectives. Good results were obtained in Estellon et al. (2006, 2008) by applying VNS-related heuristics for real-life car sequencing problems.

5.9.8 Other scheduling problems

Davidović et al. (2005) use VNS heuristics for multiprocessor scheduling with communication delays. Higgins et al. (2006) apply VNS to the scheduling of brand production and shipping within a sugar supply chain and Lejeune (2006) also consider supply chain planning. Liang and Chen (2007) tackle the redundancy allocation of series-parallel systems, using a VNS algorithm.

Remde et al. (2007) use reduced VNS and hyperheuristic approaches to tackle subproblems in an Exact/Hybrid heuristic for Workforce Scheduling. Xhafa (2007) considers a hybrid evolutionary metaheuristic based on memetic algorithms and VNS to job scheduling on computational grids. Liang et al. (2007) apply VNS to redundancy allocation problems.

Lusa and Potts (2008) use a VNS algorithm for the constrained task allocation problem. Almada-Lobo et al. (2008) report the use of a VNS approach to production planning and scheduling in the glass container industry. Dahal et al. (2008) apply a constructive search and VNS to tackle a complex real world workforce scheduling problem. Abraham et al. (2008) propose a VNS/PSO hybrid for the scheduling problem in distributed data-intensive computing environments. Liao and Liao (2008) apply an ACO algorithm which uses a variable neighbourhood search as the local search to make it more efficient and effective for scheduling in agile manufacturing. Naderi et al. (2008) propose a VNS which uses advanced neighbourhood search structures for flexible flow line problems with sequence dependent setup times. Tavakkoli-Moghaddam et al. (2009) combine a memetic algorithm with a nested VNS to solve the flexible flow line scheduling problem with processor blocking and without intermediate buffers.

5.10 Vehicle routing problems

5.10.1 TSP and extensions

VNS is used for the travelling salesman problem (TSP) and its extensions. Hansen and Mladenović (1999, 2006) consider basic VNS for the euclidean TSP. Burke et al. (2001) apply guided VNS methods for the asymmetric TSP. VNS for the Pickup and Delivery TSP is considered in Carrabs et al. (2007). Hu and Raidl (2008) study the effectiveness of neighbourhood structures within a VNS approach for the Generalized TSP. Felipe et al. (2009) use a VNS approach to solve a double TSP with multiple stacks. A multi-start variant of VNS is applied by Mansini and Tocchella (2009) to solve the travelling purchaser problem with budget constraints.

5.10.2 VRP and extensions

Standard versions of the vehicle routing problem (VRP) have been solved by VNS or hybrids. A variable neighborhood descent is applied to the vehicle routing problem with backhauls in Crispim and Brandao (2001). Rousseau et al. (2002) use a variable neighbourhood descent to take advantage of different neighbourhood structures for the vehicle routing problem. An interesting development of reactive VNS for the vehicle routing problem with time windows appears in Bräysy (2003). Polacek et al. (2004) use a VNS for the multi depot vehicle routing problem with time windows. A hybrid metaheuristic merging VNS and Tabu Search applied to the location-routing problem with non-linear costs can be found in Melechovsky et al. (2005). Repoussis et al. (2006) propose a reactive greedy randomized variable neighbourhood Tabu search for the vehicle routing problem with time windows. Irnich et al. (2006) introduce sequential search as a generic technique for the efficient exploration of local-search neighbourhoods such as VNS and consider its application to vehicle-routing problems. Kytöjoki et al. (2007) propose an efficient VNS heuristic for very large scale vehicle routing problems. Geiger and Wenger (2007) use VNS within an interactive resolution method for multi-objective vehicle routing problems. Fleszar et al. (2009) propose an effective VNS for the open vehicle routing problem. Liu and Chung (2009) apply a variable neighborhood tabu search to the vehicle routing problem with backhauls and inventory.

5.10.3 Practical applications

VNS has also been useful for practical applications of routing problems. Cowling and Keuthen (2005) examine iterated approaches of the Large-Step Markov Chain and VNS type and investigate their performance when used in combination with an embedded search heuristic for routing optimization. A VNS-based on-line method is proposed and tested in Goel and Gruhn (2008) for the general vehicle routing problem. The solution methodology proposed by Repoussis et al. (2007) hybridizes in a reactive fashion systematic diversification mechanisms of Greedy Randomized Adaptive Search Procedures with VNS for intensifying local searching regarding a real life vehicle routing problem.

5.10.4 Arc routing and waste collection

Hertz and Mittaz (2001) use a variable neighbourhood descent algorithm for the undirected capacitated arc routing problem. Polacek et al. (2008) develop a basic VNS algorithm to solve the capacitated arc routing problem with intermediate facilities. Nuortio et al. (2006) use VNS in an improved route planning and scheduling of waste collection and transport and Del Pia and Filippi (2006) use a variable neighbourhood descent algorithm for a real waste collection problem with mobile depots.

5.10.5 Fleet sheet problems

Yepes and Medina (2006) present a three-step local search algorithm based on a probabilistic VNS for the vehicle routing problem with a heterogeneous fleet of vehicles and soft time windows. Paraskevopoulos et al. (2008) present a reactive variable neighbourhood Tabu search for the heterogeneous fleet vehicle routing problem with time windows. Schmid et al. (2008) propose two hybrid procedures based on a combination of an exact algorithm and a VNS approach for the distribution of ready-mixed concrete using a heterogeneous fleet of vehicles. Imran et al. (2009) use a VNS-based heuristic to solve the heterogeneous fleet vehicle routing problem.

5.10.6 Extended vehicle routing problems

Polacek et al. (2007) use VNS to assign customers to days and determine routes for a travelling salesperson for scheduling periodic customer visits. Zhao et al. (2008) apply a variable large neighbourhood search (VLNS) algorithm, which is a special case of VNS for an inventory/routing problem in a three-echelon logistics system. Vogt et al. (2007) present a heuristic for this problem based on a variable neighbourhood Tabu search for the single vehicle routing allocation problem. Hemmelmayr et al. (2009) propose a VNS heuristic for periodic routing problems. Liu and Chung (2009) propose a variable neighbourhood Tabu search for the vehicle routing problem with backhauls and inventory and Liu et al. (2008) propose a modified VNS for solving vehicle routing problems with backhauls and time windows. Subramanian and Dos Anjos Formiga Cabral (2008) present an iterated local search procedure, which uses a variable neighbourhood descent method to perform the local search, for the vehicle routing problem, with simultaneous pickup and delivery and a time limit.

5.11 Problems in biosciences and chemistry

VNS has been useful in recently emerging areas in Bioscience and Chemistry such as Bioinformatics. Andreatta and Ribeiro (2002) propose VNS heuristics for the phylogeny problem and Ribeiro and Vianna (2005) use a GRASP with a VND heuristic for this problem with a new neighbourhood structure. Kawashimo et al. (2006) apply VNS to DNA Sequence Design and Liberti et al. (2009) propose a double VNS with smoothing for the molecular distance geometry problem. Santana et al. (2008) illustrate another example of hybridization of metaheuristics through the combination of VNS and Estimation Distribution Algorithms (EDAs). They present the first attempt to combine these two methods testing it on the protein side chain placement problem. Belacel et al. (2004b) use VNS for Fuzzy Clustering of cDNA microarray gene expression data and Dražić et al. (2008) use a continuous VNS heuristic for finding the three-dimensional structure of a molecule. Montemanni and Smith (2008) consider the construction of constant GC-content DNA vodes via a VNS Algorithm.

A VNS is tested by Polo-Corpa et al. (2009) for curve fitting in experimental data processing in chemistry.

A Multi-Start VNS hybrid (MSVNS) is applied, in Pelta et al. (2008), to the protein structure comparison problem which is a very important problem in the bio-informatics area. The Maximum Contact Map Overlap (Max-CMO) model of protein structure comparison models the proteins as a graph of the contacts between the protein residues to perform the comparison. The proposed MSVNS method is currently the best heuristic algorithm for the Max-CMO model, both in terms of optimization and in terms of the biological relevance of its results. The method is biologically relevant, since the algorithm has proven to be good enough to detect similarities at SCOP's family and CATH's architecture levels.

5.12 Continuous optimization

Several continuous optimization problems have also been successfully approached with VNS. Mladenović et al. (2008) propose a General VNS for continuous optimization and Dražić et al. (2006) a VNS-based software for Global Optimization. Audet et al. (2008) deal with Nonsmooth optimization through Mesh Adaptive Direct Search and VNS. Brimberg et al. (2006a) use VNS in a decomposition strategy for large-scale continuous location-allocation problems. Solving the unconstrained optimization problem by VNS has been successfully achieved in Toksari and Güner (2007). Ling et al. (2008) use a modified VNS metaheuristic for max-bisection problems. Sevkli and Sevilgen (2008) consider the PSO hybridized with Reduced VNS for continuous function optimization.

5.13 Other optimization problems

Some further optimization problems solved with VNS include the study of the dynamics of handwriting (Caporossi et al. 2004), the problem of multi-item, single level, capacitated, dynamic lot-sizing with set-up times (Hindi et al. 2003), the linear ordering problem (García et al. 2006), the minimum cost berth allocation problem (Hansen et al. 2008c) and the run orders problem in the presence of serial correlation (Garroi et al. 2009).

Mori and Tsunokawa (2005) use a variable neighbourhood Tabu search for capacitor placement in distribution systems. Haugland (2007) develops a randomized search heuristic, which in some sense resembles VNS, for the subspace selection problem. Hemmelmayr et al. (2008) apply solution approaches based on integer programming and VNS to organize the delivery of blood products to Austrian hospitals for the blood bank of the Austrian Red Cross for Eastern Austria. Claro and Sousa (2008) propose a hybrid approach, combining Tabu Search and VNS for a mean-risk multistage capacity investment problem. Mladenović et al. (2009) use a VNS based heuristic to solve the problem of reducing the bandwidth of a matrix.

VNS is used to solve satisfiability problems. Hansen et al. (2000) use VNS for the weighted maximum satisfiability problem. Ognjanović et al. (2005), Jovanović et al. (2007) and Sevkli and Aydin (2007) use VNS for the probabilistic satisfiability problem. Hansen and Perron (2008) use VNS to solve the subproblem in a column generation approach which merges the local and global approaches to probabilistic satisfiability. Loudni and Boizumault (2008) apply the (VNS/LDS+CP) hybrid for solving optimization problems in anytime contexts. The (VNS/LDS+CP) procedure combines a VNS scheme with Limited Discrepancy Search (LDS) using Constraint Propagation (CP).

5.14 Discovery science

In all these applications VNS is used as an optimization tool. It can also lead to results in "discovery science", i.e., help in the development of theories. This has been done for graph theory in a long series of papers with the common title "Variable neighborhood search for extremal graphs" and reporting on the development and applications of the system AutoGraphiX (AGX) (Caporossi and Hansen 2000, 2004; Aouchiche et al. 2005a). This system addresses the following problems:

– Find a graph satisfying given constraints;
– Find optimal or near optimal graphs for an invariant subject to constraints;
– Refute a conjecture;
– Suggest a conjecture (or repair or sharpen one);
– Provide a proof (in simple cases) or suggest an idea of proof.

A basic idea is then to consider all of these problems as parametric combinatorial optimization problems on the infinite set of all graphs (or in practice some smaller subset) with a generic heuristic. This is done by applying VNS to find extremal graphs, with a given number n of vertices (and possibly also a given number of edges). Then a VND with many neighbourhoods is used. Those neighborhoods are defined by modifications of the graphs such as the removal or addition of an edge, rotation of an edge, and so forth. Once a set of extremal graphs, parameterized by their order, is found, their properties are explored with various data mining techniques, leading to conjectures, refutations and simple proofs or ideas of proof.

The current list of titles of papers in the series "VNS for extremal graphs" is given in Table 1 below.

Another list of papers, not included in this series is given in the following Table 2.

Papers in these two lists cover a variety of topics:

(i) Principles of the approach (1.1, 1.5) and its implementation (1.14);
(ii) Applications to spectral graph theory, e.g., bounds on the index for various families of graphs, graphs maximizing the index subject to some conditions (1.3, 1.11, 1.16, 1.17, 2.7);
(iii) Studies of classical graph parameters, e.g., independence, chromatic number, clique number, average distance (1.13, 1.21, 1.22, 1.24, 1.25, 1.26, 2.8);
(iv) Studies of little known or new parameters of graphs, e.g., irregularity, proximity and remoteness (1.9, 2.9)
(v) New families of graphs discovered by AGX, e.g., bags, which are obtained from complete graphs by replacing an edge by a path, and bugs, which are obtained by cutting the paths of a bag (1.15, 1.27);
(vi) Applications to mathematical chemistry, e.g., study of chemical graph energy, and of the Randić index (1.4, 1.6, 1.7, 1.10, 1.18, 1.19, 2.2, 2.3, 2.6);
(vii) Results of a systematic study of 20 graph invariants, which led to almost 1500 new conjectures, more than half of which were proved by AGX and over 300 by various mathematicians (1.20);
(viii) Refutation or strengthening of conjectures from the literature (1.8, 2.1, 2.6);
(ix) Surveys and discussions about various discovery systems in graph theory, assessment of the state-of-the-art and the forms of interesting conjectures together with proposals for the design of more powerful systems (2.4, 2.5).

Table 1 List of papers in the series "VNS for extremal graphs"

	Author(s)	Title
1.1	Caporossi and Hansen (2000)	*The AutoGraphiX system*
1.2	Caporossi et al. (1999a)	*Finding graphs with extremal energy*
1.3	Cvetkovic et al. (2001)	*On the largest eigenvalue of color-constrained trees*
1.4	Caporossi et al. (1999c)	*Chemical trees with extremal connectivity index*
1.5	Caporossi and Hansen (2004)	*Three ways to automate finding conjectures*
1.6	Hansen and Mélot (2003)	*Analysing bounds for the connectivity index*
1.7	Fowler et al. (2001)	*Polyenes with maximum HOMO-LUMO gap*
1.8	Aouchiche et al. (2001)	*Variations on Graffiti 105*
1.9	Hansen and Mélot (2005)	*Bounding the irregularity of a graph*
1.10	Gutman et al. (2005)	*Comparison of irregularity indices for chemical trees*
1.11	Belhaiza et al. (2007)	*Bounds on algebraic connectivity*
1.12	Hansen et al. (2005b)	*A note on the variance of bounded degrees in graphs*
1.13	Aouchiche and Hansen (2005)	*'À propos de la maille' (French)*
1.14	Aouchiche et al. (2005a)	*The AutoGraphiX 2 system*
1.15	Hansen and Stevanović (2005)	*On bags and bugs*
1.16	Aouchiche et al. (2008)	*Some conjectures related to the largest eigenvalue of a graph*
1.17	Aouchiche et al. (2005c)	*Further conjectures and results about the index*
1.18	Aouchiche et al. (2006)	*Conjectures and results about the Randic index*
1.19	Aouchiche et al. (2007d)	*Further conjectures and results about the Randic index*
1.20	Aouchiche et al. (2007a)	*Automated comparison of graph invariants*
1.21	Aouchiche et al. (2009a)	*Conjectures and results about the independence number*
1.22	Aouchiche et al. (2009b)	*Extending bounds for independence to upper irredundance*
1.23	Hansen and Vukičević (2006)	*On the Randic index and the chromatic number*
1.24	Sedlar et al. (2007a)	*Conjectures and results about the clique number*
1.25	Sedlar et al. (2007b)	*Products of connectivity and distance measures*
1.26	Aouchiche et al. (2007c)	*'Nouveaux rèsultats sur la maille' (French)*
1.27	Aouchiche et al. (2007b)	*Families of extremal graphs*

6 Conclusions

The general schemes of variable neighborhood search have been presented, discussed and illustrated by examples. In order to evaluate the VNS research program, one needs a list

Table 2 A further list of papers on AGX

	Author(s)	Title
2.1	Caporossi et al. (1999b)	*Trees with palindromic Hosoya polynomials*
2.2	Gutman et al. (1999)	*Alkanes with small and large Randić connectivity indices*
2.3	Hansen (2002)	*Computers in graph theory*
2.4	Hansen and Mélot (2002)	*Computers and discovery in algebraic graph theory*
2.5	Caporossi et al. (2003)	*Graphs with maximum connectivity index*
2.6	Hansen (2005)	*How far is, should and could be conjecture-making in graph theory an automated process?*
2.7	Hansen et al. (2005a)	*What forms do interesting conjectures have in graph theory?*
2.8	Aouchiche et al. (2005b)	*AutoGraphiX: A survey*
2.9	Aouchiche and Hansen (2007a)	*Automated results and conjectures on average distance in graphs*
2.10	Aouchiche and Hansen (2007b)	*On a conjecture about the Randic index*
2.11	Stevanovic et al. (2008)	*On the spectral radius of graphs with a given domination number*
2.12	Aouchiche and Hansen (2008a)	*Bounding average distance using minimum degree*
2.13	Aouchiche and Hansen (2008b)	*Nordhaus-Gaddum relations for proximity and remoteness in graphs*

of the desirable properties of metaheuristics. The following eight of these are presented in Hansen and Mladenović (2003):

(i) *Simplicity:* the metaheuristic should be based on a simple and clear principle, which should be widely applicable;
(ii) *Precision:* the steps of the metaheuristic should be formulated in precise mathematical terms, independent of possible physical or biological analogies which may have been the initial source of inspiration;
(iii) *Coherence:* all steps of the heuristics for particular problems should follow naturally from the principle of the metaheuristic;
(iv) *Efficiency:* heuristics for particular problems should provide optimal or near-optimal solutions for all or at least most realistic instances. Preferably, they should find optimal solutions for most problems of benchmarks for which such solutions are known, when available;
(v) *Effectiveness:* heuristics for particular problems should take a moderate computing time to provide optimal or near-optimal solutions;
(vi) *Robustness:* the performance of heuristics should be consistent over a variety of instances, i.e., not merely fine-tuned to some training set and less good elsewhere;
(vii) *User-friendliness:* heuristics should be clearly expressed, easy to understand and, most important, easy to use. This implies they should have as few parameters as possible, ideally none;
(viii) *Innovation:* preferably, the principle of the metaheuristic and/or the efficiency and effectiveness of the heuristics derived from it should lead to new types of application.

This list has been completed with three more items added by one member of the present team and his collaborators:

(ix) *Generality:* the metaheuristic should lead to good results for a wide variety of problems;
(x) *Interactivity:* the metaheuristic should allow the user to incorporate his knowledge to improve the resolution process;
(xi) *Multiplicity:* the metaheuristic should be able to present several near optimal solutions from which the user can choose one.

As argued here and above, VNS possesses, to a great extent, all of the above properties. This has led to heuristics which are among the very best ones for many problems. Interest in VNS is clearly growing at speed. This is evidenced by the increasing number of papers published each year on this topic (ten years ago, only a few; five years ago, about a dozen; and about 50 in 2007). Moreover, the 18th EURO Mini conference held in Tenerife in November 2005 was entirely devoted to VNS. It led to special issues of *IMA Journal of Management Mathematics* in 2007 (Melián and Mladenović 2007), and *European Journal of Operational Research* (Hansen et al. 2008a) and *Journal of Heuristics* (Moreno-Vega and Melián 2008) in 2008. In retrospect, it appears that the good shape of the VNS research program is due to the following decisions, strongly influenced by Karl Popper's philosophy of science (Popper 1959): (i) in devising heuristics favour insight over efficiency (which comes later) and (ii) learn from the heuristics mistakes.

Acknowledgements The first author was partially supported by NSERC grant number FQRNT. The second author was partially supported by the Serbian Ministry of Science, grant number 144007. The third author was partially supported by projects TIN2005-08404-C04-03 and TIN2008-06872-C04-01 of the Spanish Government (with financial support from the E.U. under the FEDER project).

References

Abraham, A., Liu, H., & Zhao, M. (2008). Particle swarm scheduling for work-flow applications in distributed computing environments. *Studies in Computational Intelligence*, *128*, 327–342.
Alba, E., & Domínguez, E. (2006). Comparative analysis of modern optimization tools for the p-median problem. *Statistics and Computing*, *16*(3), 251–260.
Almada-Lobo, B., Oliveira, J. F., & Carravilla, M. A. (2008). Production planning and scheduling in the glass container industry: A VNS approach. *International Journal of Production Economics*, *114*(1), 363–375.
Aloise, D. J., Aloise, D., Rocha, C. T. M., Ribeiro, C. C., Ribeiro, J. C., & Moura, L. S. S. (2006). Scheduling workover rigs for onshore oil production. *Discrete Applied Mathematics*, *154*(5), 695–702.
Amaldi, E., Liberti, L., Maffioli, F., & Maculan, N. (2009). Edge-swapping algorithms for the minimum fundamental cycle basis problem. *Mathematical Methods of Operations Research*, *69*(2), 205–233.
Andreatta, A., & Ribeiro, C. (2002). Heuristics for the phylogeny problem. *Journal of Heuristics*, *8*(4), 429–447.
Anghinolfi, D., & Paolucci, M. (2007). Parallel machine total tardiness scheduling with a new hybrid metaheuristic approach. *Computers and Operations Research*, *34*(11), 3471–3490.
Aouchiche, M., & Hansen, P. (2005). Recherche à voisinage variable de graphes extrèmes 13. À propos de la maille. *RAIRO Operations Research*, *39*, 275–293 (French).
Aouchiche, M., & Hansen, P. (2007a). Automated results and conjectures on average distance in graphs. *Graph Theory in Paris, Trends in Mathematics*, *VI*, 21–36.
Aouchiche, M., & Hansen, P. (2007b). On a conjecture about the Randic index. *Discrete Mathematics*, *307*, 262–265.
Aouchiche, M., & Hansen, P. (2008a). Bounding average distance using order and minimum degree. *Les Cahiers du GERAD G-2008-35*. To appear in *Graph Theory Notes of New York*.
Aouchiche, M., & Hansen, P. (2008b). Nordhaus-Gaddum relations for proximity and remoteness in graphs. *Les Cahiers du GERAD G-2008-36*.
Aouchiche, M., Caporossi, G., & Cvetković, D. (2001). Variable neighborhood search for extremal Variations on Graffiti 105graphs 8. *Congressus Numerantium*, *148*, 129–144.

Aouchiche, M., Bonnefoy, J. M., Fidahoussen, A., Caporossi, G., Hansen, P., Hiesse, L., Lacheré, J., & Monhait, A. (2005a). Variable neighborhood search for extremal graphs 14. The AutoGraphiX 2 system. In L. Liberti & N. Maculan (Eds.), *Global optimization: from theory to implementation* (pp. 281–309). Berlin: Springer.

Aouchiche, M., Caporossi, G., Hansen, P., & Laffay, M. (2005b). AutoGraphiX: a survey. *Electronic Notes in Discrete Mathematics*, *22*, 515–520.

Aouchiche, M., Hansen, P., & Stevanović, D. (2005c). Variable neighborhood search for extremal graphs 17. Further conjectures and results about the index. *Les Cahiers du GERAD G-2005-78*. To appear in *Discusiones Mathematicae: Graph Theory*.

Aouchiche, M., Hansen, P., & Zheng, M. (2006). Variable neighborhood search for extremal graphs 18. Conjectures and results about the Randic index. *MATCH Communications in Mathematical and Computer Chemistry*, *56*(3), 541–550.

Aouchiche, M., Caporossi, G., & Hansen, P. (2007a). Variable neighborhood search for extremal graphs 20. Automated comparison of graph invariants. *MATCH Communications in Mathematical and Computer Chemistry*, *58*(2), 365–384.

Aouchiche, M., Caporossi, G., & Hansen, P. (2007b). Variable neighborhood search for extremal graphs 27. Families of extremal graphs. *Les Cahiers du GERAD G-2007-87*.

Aouchiche, M., Favaron, O., & Hansen, P. (2007c). Recherche à voisinage variable de graphes extrèmes 26. Nouveaux résultats sur la maille (French). *Les Cahiers du GERAD G-2007-55*.

Aouchiche, M., Hansen, P., & Zheng, M. (2007d). Variable neighborhood search for extremal graphs 19. Further conjectures and results about the Randic index. *MATCH Communications in Mathematical and Computer Chemistry*, *58*(1), 83–102.

Aouchiche, M., Bell, F. K., Cvetković, D., Hansen, P., Rowlinson, P., Simić, S. K., & Stevanović, D. (2008). Variable neighborhood search for extremal graphs 16. Some conjectures related to the largest eigenvalue of a graph. *European Journal of Operational Research*, *191*(3), 661–676.

Aouchiche, M., Brinkmann, G., & Hansen, P. (2009a). Variable neighborhood search for extremal graphs 21. Conjectures and results about the independence number. *Discrete Applied Mathematics*, *156*(13), 2530–2542.

Aouchiche, M., Favaron, O., & Hansen, P. (2009b). Variable neighborhood search for extremal graphs 22. Extending bounds for independence to upper irredundance. *Discrete Applied Mathematics*. doi:10.1016/j.dam.2009.04.004.

Archetti, C., Hertz, A., & Speranza, M. G. (2007). Metaheuristics for the team orienteering problem. *Journal of Heuristics*, *13*(1), 49–76.

Audet, C., Brimberg, J., Hansen, P., & Mladenović, N. (2004). Pooling problem: alternate formulation and solution methods. *Management Science*, *50*, 761–776.

Audet, C., Báchard, V., & Le Digabel, S. (2008). Nonsmooth optimization through mesh adaptive direct search and variable neighborhood search. *Journal of Global Optimization*, *41*(2), 299–318.

Avanthay, C., Hertz, A., & Zufferey, N. (2003). A variable neighborhood search for graph coloring. *European Journal of Operational Research*, *151*(2), 379–388.

Aydin, M. E., & Sevkli, M. (2008). Sequential and parallel variable neighborhood search algorithms for job shop scheduling. *Studies in Computational Intelligence*, *128*, 125–144.

Baum, E. B. (1986). Toward practical 'neural' computation for combinatorial optimization problems. In J. Denker (Ed.), *Neural networks for computing*. American Institute of Physics.

Behnamian, J., Fatemi Ghomi, S. M. T., & Zandieh, M. (2009a). A multi-phase covering Pareto-optimal front method to multi-objective scheduling in a realistic hybrid flowshop using a hybrid metaheuristic. *Expert Systems with Applications*, *36*(8), 11057–11069.

Behnamian, J., Zandieh, M., & Fatemi Ghomi, S. M. T. (2009b). Parallel-machine scheduling problems with sequence-dependent setup times using an ACO, SA and VNS hybrid algorithm. *Expert Systems with Applications*, *36*(6), 9637–9644.

Belacel, N., Hansen, P., & Mladenović, N. (2002). Fuzzy J-means: a new heuristic for fuzzy clustering. *Pattern Recognition*, *35*(10), 2193–2200.

Belacel, N., Čuperlović-Culf, M., Laflamme, M., & Ouellette, R. (2004a). Fuzzy J-means and VNS methods for clustering genes from microarray data. *Bioinformatics*, *20*(11), 1690–1701.

Belacel, N., Čuperlović-Culf, M., Ouellette, R., & Boulassel, M. R. (2004b). The variable neighborhood search metaheuristic for fuzzy clustering cDNA microarray gene expression data. In M. H. Hamza (Ed.), *Artificial intelligence and applications*. Calgary: Acta Press.

Belacel, N., Raval, H. B., & Punnen, A. P. (2007). Learning multicriteria fuzzy classification method PROAFTN from data. *Computers and Operations Research*, *34*(7), 1885–1898.

Belhaiza, S., de Abreu, N., Hansen, P., & Oliveira, C. (2007). Variable neighborhood search for extremal graphs 11. Bounds on algebraic connectivity. In D. Avis, A. Hertz, & O. Marcotte (Eds.), *Graph theory and combinatorial optimization* (pp. 1–16).

Beltrán, J. D., Calderón, J. E., Jorge-Cabrera, R., Moreno-Pérez, J. A., & Moreno-Vega, J. M. (2004). GRASP-VNS hybrid for the strip packing problem. In *Hybrid metaheuristics 2004* (pp. 79–90).

Benati, S. (2008). Categorical data fuzzy clustering: an analysis of local search heuristics. *Computers and Operations Research*, *35*(3), 766–775.

Benati, S., & Hansen, P. (2002). The maximum capture problem with random utilities: problem formulation and algorithms. *European Journal of Operational Research*, *143*(3), 518–530.

Bischoff, M., & Dächert, K. (2009). Allocation search methods for a generalized class of location-allocation problems. *European Journal of Operational Research*, *192*(3), 793–807.

Blazewicz, J., Pesch, E., Sterna, M., & Werner, F. (2005). Metaheuristics for late work minimization in two-machine flow shop with common due date. In *Lecture notes in artificial intelligence* (Vol. 3698, pp. 222–234). Berlin: Springer.

Blazewicz, J., Pesch, E., Sterna, M., & Werner, F. (2008). Metaheuristic approaches for the two-machine flow-shop problem with weighted late work criterion and common due date. *Computers and Operations Research*, *35*(2), 574–599.

Bouffard, V., & Ferland, J. A. (2007). Improving simulated annealing with variable neighborhood search to solve the resource-constrained scheduling problem. *Journal of Scheduling*, *10*(6), 375–386.

Bräysy, O. (2003). A reactive variable neighborhood search for the vehicle routing problem with time windows. *INFORMS Journal on Computing*, *15*(4), 347–368.

Brimberg, J., & Mladenović, N. (1996). A variable neighborhood algorithm for solving the continuous location-allocation problem. *Studies in Locational Analysis*, *10*, 1–12.

Brimberg, J., Hansen, P., Mladenović, N., & Taillard, É. (2000). Improvements and comparison of heuristics for solving the multisource Weber problem. *Operations Research*, *48*(3), 444–460.

Brimberg, J., Hansen, P., Lih, K.-W., Mladenović, N., & Breton, M. (2003). An oil pipeline design problem. *Operations Research*, *51*(2), 228–239.

Brimberg, J., Mladenović, N., & Salhi, S. (2004). The multi-source Weber problem with constant opening cost. *Journal of the Operational Research Society*, *55*, 640–646.

Brimberg, J., Hansen, P., & Mladenović, N. (2006a). Decomposition strategies for large-scale continuous location–allocation problems. *IMA Journal of Management Mathematics*, *17*, 307–316.

Brimberg, J., Urošević, D., & Mladenović, N. (2006b). Variable neighborhood search for the vertex weighted k-cardinality tree problem. *European Journal of Operational Research*, *171*(1), 74–84.

Brimberg, J., Hansen, P., Laporte, G., Mladenović, N., & Urošević, D. (2008a). The maximum return-on-investment plant location problem with market share. *Journal of the Operational Research Society*, *59*(3), 399–406.

Brimberg, J., Mladenović, N., & Urošević, D. (2008b). Local and variable neighborhood search for the k-cardinality subgraph problem. *Journal of Heuristics*, *14*(5), 501–517.

Brimberg, J., Mladenović, N., Urošević, D., & Ngai, E. (2009). Variable neighborhood search for the heaviest k-subgraph. *Computers and Operations Research*, *36*(11), 2885–2891.

Brusco, M., & Steinley, D. (2007a). A variable neighborhood search method for generalized blockmodeling of two-mode binary matrices. *Journal of Mathematical Psychology*, *51*(5), 325–338.

Brusco, M. J., & Steinley, D. (2007b). A comparison of heuristic procedures for minimum within-cluster sums of squares partitioning. *Psychometrika*, *72*(4), 583–600.

Brusco, M. J., Köhn, H.-F., & Stahl, S. (2008). Heuristic implementation of dynamic programming for matrix permutation problems in combinatorial data analysis. *Psychometrika*, *73*(3), 503–522.

Brusco, M. J., Singh, R., & Steinley, D. (2009). Variable neighborhood search heuristics for selecting a subset of variables in principal component analysis. *Psychometrika*. doi:10.1007/s11336-009-9130-3.

Burke, E. K., & Kendall, G. (2005). *Search methodologies. Introductory tutorials in optimization and decision support techniques*. Berlin: Springer.

Burke, E. K., Cowling, P., & Keuthen, R. (2001). Effective local and guided variable neighborhood search methods for the asymmetric travelling salesman problem. In *Lecture notes in computer science* (Vol. 2037, pp. 203–212). Berlin: Springer.

Canuto, S., Resende, M., & Ribeiro, C. (2001). Local search with perturbations for the prize-collecting Steiner tree problem in graphs. *Networks*, *31*(3), 201–206.

Caporossi, G., & Hansen, P. (2000). Variable neighborhood search for extremal graphs 1. The AutoGraphiX system. *Discrete Mathematics*, *212*, 29–44.

Caporossi, G., & Hansen, P. (2004). Variable neighborhood search for extremal graphs 5. Three ways to automate finding conjectures. *Discrete Mathematics*, *276*(1–3), 81–94.

Caporossi, G., Cvetković, D., Gutman, I., & Hansen, P. (1999a). Variable neighborhood search for extremal graphs 2. Finding graphs with extremal energy. *Journal of Chemical Information and Computer Sciences*, *39*, 984–996.

Caporossi, G., Dobrynin, A. A., Gutman, I., & Hansen, P. (1999b). Trees with palindromic Hosoya polynomials. *Graph Theory Notes of New York*, *37*, 10–16.

Caporossi, G., Gutman, I., & Hansen, P. (1999c). Variable neighborhood search for extremal graphs 4. Chemical trees with extremal connectivity index. *Computers and Chemistry*, *23*(5), 469–477.
Caporossi, G., Gutman, I., Hansen, P., & Pavlović, L. (2003). Graphs with maximum connectivity index. *Computational Biology and Chemistry*, *27*, 85–90.
Caporossi, G., Alamargot, D., & Chesnet, D. (2004). Using the computer to study the dynamics of the handwriting processes. In *Lecture notes in computer science* (Vol. 3245, pp. 242–254). Berlin: Springer.
Carrabs, F., Cordeau, J.-F., & Laporte, G. (2007). Variable neighbourhood search for the pickup and delivery traveling salesman problem with LIFO loading. *INFORMS Journal on Computing*, *19*(4), 618–632.
Carrizosa, E., Martín-Barragán, B., Plastria, F., & Romero Morales, D. (2007). On the selection of the globally optimal prototype subset for nearest-neighbor classification. *INFORMS Journal on Computing*, *19*(3), 470–479.
Chen, C.-L., & Chen, C.-L. (2009). Hybrid metaheuristic for unrelated parallel machine scheduling with sequence-dependent setup times. *International Journal of Advanced Manufacturing Technology*, *43*(1–2), 161–169.
Chyu, C.-C., & Chen, Z.-J. (2009). Scheduling jobs under constant period-by-period resource availability to maximize project profit at a due date. *International Journal of Advanced Manufacturing Technology*, *42*(5–6), 569–580.
Claro, J., & de Sousa, J. P. (2008). A multiobjective metaheuristic for a mean-risk multistage capacity investment problem. *Journal of Heuristics*. doi:10.1007/s10732-008-9090-2.
Consoli, S., Darby-Dowman, K., Mladenović, N., & Moreno Pérez, J. A. (2009a). Greedy randomized adaptive search and variable neighbourhood search for the minimum labelling spanning tree problem. *European Journal of Operational Research*, *196*(2), 440–449.
Consoli, S., Darby-Dowman, K., Mladenović, N., & Moreno-Pérez, J. A. (2009b). Variable neighbourhood search for the minimum labelling Steiner tree problem. *Annals of Operations Research*. doi:10.1007/s10479-008-0507-y.
Costa, M. C., Monclar, F. R., & Zrikem, M. (2002). Variable neighborhood decomposition search for the optimization of power plant cable layout. *Journal of Intelligent Manufacturing*, *13*(5), 353–365.
Cote, P., Wong, T., & Sabourin, R. (2005). A hybrid multi-objective evolutionary algorithm for the uncapacitated exam proximity problem. In *Lecture notes in computer science* (Vol. 3616, pp. 294–312). Berlin: Springer.
Cowling, P. I., & Keuthen, R. (2005). Embedded local search approaches for routing optimization. *Computers and Operations Research*, *32*(3), 465–490.
Crainic, T., Gendreau, M., Hansen, P., & Mladenović, N. (2004). Cooperative parallel variable neighborhood search for the p-median. *Journal of Heuristics*, *10*, 289–310.
Crispim, J., & Brandao, J. (2001). Reactive tabu search and variable neighborhood descent applied to the vehicle routing problem with backhauls. In *MIC'2001* (pp. 631–636). Porto, 2001.
Cvetkovic, D., Simic, S., Caporossi, G., & Hansen, P. (2001). Variable neighborhood search for extremal graphs 3. On the largest eigenvalue of color-constrained trees. *Linear and Multilinear Algebra*, *49*, 143–160.
Czogalla, J., & Fink, A. (2008). On the effectiveness of particle swarm optimization and variable neighborhood descent for the continuous flow-shop scheduling problem. *Studies in Computational Intelligence*, *128*, 61–89.
Dahal, K., Remde, S., Cowling, P., & Colledge, N. (2008). Improving metaheuristic performance by evolving a variable fitness function. In *Lecture notes in computer science* (Vol. 4972, pp. 170–181). Berlin: Springer.
Davidon, W. C. (1959). Variable metric algorithm for minimization. *Argonne National Laboratory Report ANL-5990*.
Davidović, T., Hansen, P., & Mladenović, N. (2005). Permutation-based genetic, tabu, and variable neighborhood search heuristics for multiprocessor scheduling with communication delays. *Asia-Pacific Journal of Operational Research*, *22*(3), 297–326.
De Paula, M. R., Ravetti, M. G., Mateus, G. R., & Pardalos, P. M. (2007). Solving parallel machines scheduling problems with sequence-dependent setup times using Variable Neighbourhood Search. *IMA Journal of Management Mathematics*, *18*(2), 101–115.
de Souza, M. C., & Martins, P. (2008). Skewed VNS enclosing second order algorithm for the degree constrained minimum spanning tree problem. *European Journal of Operational Research*, *191*(3), 677–690.
Degila, J. R., & Sansò, B. (2004). Topological design optimization of a Yottabit-per-second lattice network. *IEEE Journal on Selected Areas in Communications*, *22*(9), 1613–1625.
Del Pia, A., & Filippi, C. (2006). A variable neighborhood descent algorithm for a real waste collection problem with mobile depots. *International Transactions in Operational Research*, *13*(2), 125–141.

Desrosiers, J., Mladenović, N., & Villeneuve, D. (2005). Design of balanced MBA student teams. *Journal of the Operational Research Society*, *56*(1), 60–66.

Dias, T. C. S., de Sousa, G. F., Macambira, E. M., Cabral, L. D. A. F., & Fampa, M. H. C. (2006). An efficient heuristic for the ring star problem. In *Lecture notes in computer science* (Vol. 4007, pp. 24–35). Berlin: Springer.

Domínguez-Marín, P., Nickel, S., Hansen, P., & Mladenović, N. (2005). Heuristic procedures for solving the discrete ordered median problem. *Annals of Operations Research*, *136*(1), 145–173.

Dražić, M., Kovacevic-Vujcić, V., Cangalović, M., & Mladenović, N. (2006). GLOB—A new VNS-based software for global optimization. In L. Liberti & N. Maculan (Eds.), *Global optimization: from theory to implementation* (pp. 135–144). Berlin: Springer.

Dražić, M., Lavor, C., Maculan, N., & Mladenović, N. (2008). A continuous variable neighborhood search heuristic for finding the three-dimensional structure of a molecule. *European Journal of Operational Research*, *185*(3), 1265–1273.

Drezner, Z., Hahn, P. M., & Taillard, E. D. (2005). Recent advances for the quadratic assignment problem with special emphasis on instances that are difficult for meta-heuristic methods. *Annals of Operations Research*, *139*(1), 65–94.

du Merle, O., Villeneuve, D., Desrosiers, J., & Hansen, P. (1999). Stabilized column generation. *Discrete Mathematics*, *194*(1–3), 229–237.

du Merle, O., Hansen, P., Jaumard, B., & Mladenović, N. (2000). An interior point algorithm for minimum sum-of-squares clustering. *SIAM Journal on Scientific Computing*, *21*, 1485–1505.

Duarte, A., Sanchez, A., Fernandez, F., & Cabido, R. (2005). A low-level hybridization between memetic algorithm and VNS for the max-cut problem. In *GECCO 2005—Genetic and evolutionary computation conference* (pp. 999–1006).

Estellon, B., Gardi, F., & Nouioua, K. (2006). Large neighborhood improvements for solving car sequencing problems. *RAIRO Operations Research*, *40*(4), 355–379.

Estellon, B., Gardi, F., & Nouioua, K. (2008). Two local search approaches for solving real-life car sequencing problems. *European Journal of Operational Research*, *191*(3), 928–944.

Fathali, J., & Kakhki, H. T. (2006). Solving the p-median problem with pos/neg weights by variable neighborhood search and some results for special cases. *European Journal of Operational Research*, *170*(2), 440–462.

Felipe, Á., Ortuño, M. T., & Tirado, G. (2009). The double traveling salesman problem with multiple stacks: a variable neighborhood search approach. *Computers and Operations Research*, *36*(11), 2983–2993.

Festa, P., Pardalos, P. M., Resende, M. G. C., & Ribeiro, C. C. (2002). Randomized heuristics for the MAX-CUT problem. *Optimization Methods and Software*, *17*(6), 1033–1058.

Fischetti, M., & Lodi, A. (2003). Local branching. *Mathematical Programming*, *98*(1–3), 23–47.

Fischetti, M., Polo, C., & Scantamburlo, M. (2004). A local branching heuristic for mixed-integer programs with 2-level variables, with an application to a telecommunication network design problem. *Networks*, *44*(2), 61–72.

Fleszar, K., & Hindi, K. S. (2002). New heuristics for one-dimensional bin-packing. *Computers and Operations Research*, *29*, 821–839.

Fleszar, K., & Hindi, K. S. (2004). Solving the resource-constrained project scheduling problem by a variable neighborhood search. *European Journal of Operational Research*, *155*(2), 402–413.

Fleszar, K., & Hindi, K. S. (2008). An effective VNS for the capacitated p-median problem. *European Journal of Operational Research*, *191*(3), 612–622.

Fleszar, K., Osman, I. H., & Hindi, K. S. (2009). A variable neighbourhood search algorithm for the open vehicle routing problem. *European Journal of Operational Research*, *195*(3), 803–809.

Fletcher, R., & Powell, M. J. D. (1963). Rapidly convergent descent method for minimization. *The Computer Journal*, *6*, 163–168.

Fowler, P. W., Hansen, P., Caporossi, G., & Soncini, A. (2001). Variable neighborhood search for extremal graphs 7. Polyenes with maximum HOMO-LUMO gap. *Chemical Physics Letters*, *49*, 143–146.

Gagné, C., Gravel, M., & Price, W. L. (2005). Using metaheuristic compromise programming for the solution of multiple-objective scheduling problems. *Journal of the Operational Research Society*, *56*, 687–698.

Galinier, P., & Hertz, A. (2006). A survey of local search methods for graph coloring. *Computers and Operations Research*, *33*(9), 2547–2562.

Gao, J., Sun, L., & Gen, M. (2008). A hybrid genetic and variable neighborhood descent algorithm for flexible job shop scheduling problems. *Computers and Operations Research*, *35*(9), 2892–2907.

García, C. G., Pérez-Brito, D., Campos, V., & Martí, R. (2006). Variable neighborhood search for the linear ordering problem. *Computers and Operations Research*, *33*(12), 3549–3565.

García-López, F., Melián-Batista, B., Moreno-Pérez, J. A., & Moreno-Vega, J. M. (2002). The parallel variable neighborhood search for the p-median problem. *Journal of Heuristics*, *8*(3), 375–388.

Garey, M. R., & Johnson, D. S. (1978). *Computers and intractability: A guide to the theory of NP-completeness*. New York: Freeman.

Garroi, J.-J., Goos, P., & Sörensen, K. (2009). A variable-neighbourhood search algorithm for finding optimal run orders in the presence of serial correlation. *Journal of Statistical Planning and Inference*, *139*(1), 30–44.

Gavranović, H. (2008). Local search and suffix tree for car-sequencing problem with colors. *European Journal of Operational Research*, *191*(3), 972–980.

Geiger, M. J., & Wenger, W. (2007). On the interactive resolution of multi-objective vehicle routing problems. In *Lecture notes in artificial intelligence* (Vol. 4403, pp. 687–699). Berlin: Springer.

Geiger, M. J., & Wenger, W. (2009). On the assignment of students to topics: a Variable Neighborhood Search approach. *Socio-Economic Planning Sciences*. doi:10.1016/j.seps.2009.03.001.

Gill, P., Murray, W., & Wright, M. (1981). *Practical optimization*. London: Academic Press.

Glover, F., & Kochenberger, G. (Eds.) (2003). *Handbook of metaheuristics*. Amsterdam: Kluwer.

Goel, A., & Gruhn, V. (2008). A general vehicle routing problem. *European Journal of Operational Research*, *191*(3), 650–660.

Griffith, R. E., & Stewart, R. A. (1961). A nonlinear programming technique for the optimization of continuous processing systems. *Management Science*, *7*, 379–392.

Gupta, S. R., & Smith, J. S. (2006). Algorithms for single machine total tardiness scheduling with sequence dependent setups. *European Journal of Operational Research*, *175*(2), 722–739.

Gutjahr, W. J., Katzensteiner, S., & Reiter, P. (2007). A VNS algorithm for noisy problems and its application to project portfolio analysis. In *Lecture notes in computer science* (Vol. 4665, pp. 93–104). Berlin: Springer.

Gutman, I., Miljković, O., Caporossi, G., & Hansen, P. (1999). Alkanes with small and large Randić connectivity indices. *Chemical Physics Letters*, *306*, 366–372.

Gutman, I., Hansen, P., & Mélot, H. (2005). Variable neighborhood search for extremal graphs 10. Comparison of irregularity indices for chemical trees. *Journal of Chemical Information and Modeling*, *45*, 222–230.

Han, H., Ye, J., & Lv, Q. (2007). A VNS-ANT algorithm to QAP. In *Third international conference on natural computation* (Vol. 3, pp. 426–430).

Hansen, P. (2002). Computers in graph theory. *Graph Theory Notes of New York*, *XLIII*, 20–39.

Hansen, P. (2005). How far is, should and could be conjecture-making in graph theory an automated process? In *Dimacs series in discrete mathematics and theoretical computer science: Vol. 69. Graph and discovery* (pp. 189–229). Providence: AMS.

Hansen, P., & Mélot, H. (2002). Computers and discovery in algebraic graph theory. *Linear Algebra and Applications*, *356*(1–3), 211–230.

Hansen, P., & Mélot, H. (2003). Variable neighborhood search for extremal graphs 6. Analysing bounds for the connectivity index. *Journal of Chemical Information and Computer Sciences*, *43*, 1–14.

Hansen, P., & Mélot, H. (2005). The irregularity of a graph. In *Dimacs series in discrete mathematics and theoretical computer science: Vol. 69. Graph and discovery* (pp. 253–264). Providence: AMS.

Hansen, P., & Mladenović, N. (1997). Variable neighborhood search for the p-median. *Location Science*, *5*, 207–226.

Hansen, P., & Mladenović, N. (1999). An introduction to variable neighborhood search. In S. Voss et al. (Eds.), *Metaheuristics, advances, trends in local search paradigms for optimization* (pp. 433–458). Amsterdam: Kluwer.

Hansen, P., & Mladenović, N. (2001a). Variable neighborhood search: principles and applications. *European Journal of Operational Research*, *130*, 449–467.

Hansen, P., & Mladenović, N. (2001b). J-Means: a new local search heuristic for minimum sum-of-squares clustering. *Pattern Recognition*, *34*, 405–413.

Hansen, P., & Mladenović, N. (2001c). Developments of variable neighborhood search. In C. Ribeiro & P. Hansen (Eds.), *Essays, surveys in metaheuristics* (pp. 415–440). Amsterdam: Kluwer.

Hansen, P., & Mladenović, N. (2003). Variable neighborhood search. In F. Glover & G. Kochenberger (Eds.), *Handbook of metaheuristics* (pp. 145–184). Amsterdam: Kluwer.

Hansen, P., & Mladenović, N. (2006). First improvement may be better than best improvement: An empirical study. *Discrete Applied Mathematics*, *154*, 802–817.

Hansen, P., & Mladenović, N. (2008). Complement to a comparative analysis of heuristics for the p-median problem. *Statistics and Computing*, *18*(1), 41–46.

Hansen, P., & Perron, S. (2007). Algorithms for $\mathcal{L}_1$-embeddability and related problems. *Journal of Classification*, *24*(2), 251–275.

Hansen, P., & Perron, S. (2008). Merging the local and global approaches to probabilistic satisfiability. *International Journal of Approximate Reasoning*, *47*(2), 125–140.

Hansen, P., & Stevanović, D. (2005). Variable neighborhood search for extremal graphs 15. On bags and bugs. *Discrete Applied Mathematics*, *156*(7), 986–997.

Hansen, P., & Vukičević, D. (2006). Variable neighborhood search for extremal graphs 23. On the Randic index and the chromatic number. *Les Cahiers du GERAD G-2006-58*. To appear in *Discrete Mathematics*.

Hansen, P., Jaumard, B., Mladenović, N., & Parreira, A. (2000). Variable neighborhood search for weighted maximum satisfiability problem. *Les Cahiers du GERAD* G–2000–62. HEC Montréal, Canada.

Hansen, P., Mladenović, N., & Pérez-Brito, D. (2001). Variable neighborhood decomposition search. *Journal of Heuristics*, *7*(4), 335–350.

Hansen, P., Mladenović, N., & Urošević, D. (2004). Variable neighborhood search for the maximum clique. *Discrete Applied Mathematics*, *145*(1), 117–125.

Hansen, P., Aouchiche, M., Caporossi, G., Mélot, H., & Stevanović, D. (2005a). What forms do interesting conjectures have in graph theory? In *Dimacs series in discrete mathematics and theoretical computer science: Vol. 69*. *Graph and discovery* (pp. 231–251). Providence: AMS.

Hansen, P., Mélot, H., & Gutman, I. (2005b). Variable neighborhood search for extremal graphs 12. A note on the variance of bounded degrees in graphs. *MATCH Communications in Mathematical and in Computer Chemistry*, *54*, 221–232.

Hansen, P., Mladenović, N., & Urošević, D. (2006). Variable neighborhood search and local branching. *Computers and Operations Research*, *33*(10), 3034–3045.

Hansen, P., Brimberg, J., Urošević, D., & Mladenović, N. (2007a). Primal-dual variable neighborhood search for the simple plant location problem. *INFORMS Journal on Computing*, *19*(4), 552–564.

Hansen, P., Lazić, J., & Mladenović, N. (2007b). Variable neighbourhood search for colour image quantization. *IMA Journal of Management Mathematics*, *18*(2), 207–221.

Hansen, P., Mladenović, N., & Moreno Pérez, J. A. (2008a). Variable neighborhood search. *European Journal of Operational Research*, *191*(3), 593–595.

Hansen, P., Mladenović, N., & Moreno Pérez, J. A. (2008b). Variable neighborhood search: methods and applications. *4OR A Quarterly Journal of Operations Research*, *6*(4), 319–360.

Hansen, P., Oğuz, C., & Mladenović, N. (2008c). Variable neighborhood search for minimum cost berth allocation. *European Journal of Operational Research*, *191*(3), 636–649.

Hansen, P., Brimberg, J., Urošević, D., & Mladenović, N. (2009). Solving large p-median clustering problems by primal-dual variable neighborhood search. *Data Mining and Knowledge Discovery*. doi:10.1007/s10618-009-0135-4.

Haugland, D. (2007). A bidirectional greedy heuristic for the subspace selection problem. In *Lecture notes in computer science* (Vol. 4638, pp. 162–176). Berlin: Springer.

Hemmelmayr, V., Doerner, K. F., Hartl, R. F., & Savelsbergh, M. W. P. (2008). Delivery strategies for blood products supplies. *OR Spectrum*, *31*(4), 707–725.

Hemmelmayr, V. C., Doerner, K. F., & Hartl, R. F. (2009). A variable neighborhood search heuristic for the periodic routing problems. *European Journal of Operational Research*, *195*(3), 791–802.

Hertz, A., & Mittaz, M. (2001). A variable neighborhood descent algorithm for the undirected capacitated arc routing problem. *Transportation Science*, *35*(4), 425–434.

Hertz, A., Plumettaz, M., & Zufferey, N. (2008). Variable space search for graph coloring. *Discrete Applied Mathematics*, *156*(13), 2551–2560.

Higgins, A., Beashel, G., & Harrison, A. (2006). Scheduling of brand production and shipping within a sugar supply chain. *Journal of the Operational Research Society*, *57*, 490–498.

Hindi, K. S., Fleszar, K., & Charalambous, C. (2003). An effective heuristic for the CLSP with setup times. *Journal of the Operational Research Society*, *54*(5), 490–498.

Höller, H., Melián, B., & Voss, S. (2008). Applying the pilot method to improve VNS and GRASP metaheuristics for the design of SDH/WDM networks. *European Journal of Operational Research*, *191*(3), 691–704.

Hu, B., & Raidl, G. R. (2008). Effective neighborhood structures for the generalized traveling salesman problem. In *Lecture notes in computer science* (Vol. 4972, pp. 36–47). Berlin: Springer.

Hu, B., Leitner, M., & Raidl, G. R. (2008). Combining variable neighborhood search with integer linear programming for the generalized minimum spanning tree problem. *Journal of Heuristics*, *14*(5), 501–517.

Imran, A., Salhi, S., & Wassan, N. A. (2009). A variable neighborhood-based heuristic for the heterogeneous fleet vehicle routing problem. *European Journal of Operational Research*, *197*(2), 509–518.

Irnich, S., Funke, B., & Grünert, T. (2006). Sequential search and its application to vehicle-routing problems. *Computers and Operations Research*, *33*(8), 2405–2429.

Jabalameli, M. S., & Ghaderi, A. (2008). Hybrid algorithms for the uncapacitated continuous location-allocation problem. *The International Journal of Advanced Manufacturing Technology*, *37*(1–2), 202–209.

Jarboui, B., Eddaly, M., & Siarry, P. (2009). An estimation of distribution algorithm for minimizing the total flowtime in permutation flowshop scheduling problems. *Computers and Operations Research*, *36*(9), 2638–2646.

Joly, A., & Frein, Y. (2008). Heuristics, for, an, industrial, car, sequencing, problem, considering, paint, and, assembly, shop, objectives. *Computers and Industrial Engineering*, *55*(2), 295–310.

Jornsten, K., & Lokketangen, A. (1997). Tabu, search, for, weighted, k-cardinality, trees. *Asia-Pacific Journal of Operational Research*, *14*(2), 9–26.

Jovanović, D., Mladenović, N., & Ognjanović, Z. (2007). Variable neighborhood search for the probabilistic satisfiability problem. In K. F. Doerner, M. Gendreau, P. Greistorfer, W. Gutjahr, R. F. Hartl, & M. Reimann (Eds.), *Metaheuristics. Progress in complex systems optimization* (pp. 173–188). Berlin: Springer.

Karam, A., Caporossi, G., & Hansen, P. (2007). Arbitrary-norm hyperplane separation by Variable Neighbourhood Search. *IMA Journal of Management Mathematics*, *18*(2), 173–190.

Kawashimo, S., Ono, H., Sadakane, K., & Yamashita, M. (2006). DNA sequence design by dynamic neighborhood searches. In *Lecture notes in computer science* (Vol. 4287, pp. 157–171). Berlin: Springer.

Kolisch, R., & Hartmann, S. (2006). Experimental investigation of heuristics for resource-constrained project scheduling: an update. *European Journal of Operational Research*, *174*(1), 23–37.

Kucherenko, S., Belotti, P., Liberti, L., & Maculan, N. (2007). New formulations for the Kissing number problem. *Discrete Applied Mathematics*, *155*(14), 1837–1841.

Kytöjoki, J., Nuortio, T., Bräysy, O., & Gendreau, M. (2007). An efficient variable neighborhood search heuristic for very large scale vehicle routing problems. *Computers and Operations Research*, *34*(9), 2743–2757.

Lapierre, S. D., Ruiz, A. B., & Soriano, P. (2004). Designing distribution networks: Formulations and solution heuristic. *Transportation Science*, *38*(2), 174–187.

Lejeune, M. A. (2006). A variable neighborhood decomposition search method for supply chain management planning problems. *European Journal of Operational Research*, *175*(2), 959–976.

Liang, Y.-C., & Chen, Y. C. (2007). Redundancy allocation of series-parallel systems using a variable neighborhood search algorithm. *Reliability Engineering and System Safety*, *92*(3), 323–331.

Liang, Y.-C., Lo, M.-H., & Chen, Y. C. (2007). Variable neighbourhood search for redundancy allocation problems. *IMA Journal of Management Mathematics*, *18*(2), 135–156.

Liao, C. J., & Cheng, C. C. (2007). A variable neighborhood search for minimizing single machine weighted earliness and tardiness with common due date. *Computers and Industrial Engineering*, *52*(4), 404–413.

Liao, C.-J., & Liao, C.-C. (2008). An ant colony optimisation algorithm for scheduling in agile manufacturing. *International Journal of Production Research*, *46*(7), 1813–1824.

Liao, C. J., Chao-Tang, T., & Luarn, P. (2007). A discrete version of particle swarm optimization for flowshop scheduling problems. *Computers and Operations Research*, *34*(10), 3099–3111.

Liberti, L., Lavor, C., Maculan, N., & Marinelli, F. (2009). Double variable neighbourhood search with smoothing for the molecular distance geometry problem. *Journal of Global Optimization*, *43*(2–3), 207–218.

Lin, S.-W., & Ying, K.-C. (2008). A hybrid approach for single-machine tardiness problems with sequence-dependent setup times. *Journal of the Operational Research Society*, *59*(8), 1109–1119.

Ling, A., Xu, C., & Tang, L. (2008). A modified VNS metaheuristic for max-bisection problems. *Journal of Computational and Applied Mathematics*, *220*(1–2), 413–421.

Liu, H., & Abraham, A. (2007). An hybrid fuzzy variable neighborhood particle swarm optimization algorithm for solving quadratic assignment problems. *Journal of Universal Computer Science*, *13*(9), 1309–1331.

Liu, S.-C., & Chung, C.-H. (2009). A heuristic method for the vehicle routing problem with backhauls and inventory. *Journal of Intelligent Manufacturing*, *20*(1), 29–42.

Liu, H. B., Abraham, A., Choi, O., & Moon, S. H. (2006). Variable neighborhood particle swarm optimization for multi-objective flexible job-shop scheduling problems. In *Lecture notes in computer science* (Vol. 4247, pp. 197–204). Berlin: Springer.

Liu, S.-X., Liu, L., & Zhang, T. (2008). Variable neighborhood search for solving vehicle routing problems with backhauls and time windows. *Journal of Northeastern University*, *29*(3), 316–319.

Ljubic, I. (2007). A hybrid VNS for connected facility location. In *Lecture notes in computer science* (Vol. 4771, pp. 157–169). Berlin: Springer.

Loudni, S., & Boizumault, P. (2008). Combining VNS with constraint programming for solving anytime optimization problems. *European Journal of Operational Research*, *191*(3), 705–735.

Loudni, S., Boizumault, P., & David, P. (2006). On-line resources allocation for ATM networks with rerouting. *Computers and Operations Research*, *33*(10), 2891–2917.

Lusa, A., & Potts, C. N. (2008). A variable neighbourhood search algorithm for the constrained task allocation problem. *Journal of the Operational Research Society*, *59*, 812–822.

Mansini, R., & Tocchella, B. (2009). The traveling purchaser problem with budget constraint. *Computers and Operations Research*, *36*(7), 2263–2274.
Martins, P., & de Souza, M. C. (2009). VNS and second order heuristics for the min-degree constrained minimum spanning tree problem. *Computers and Operations Research*, *36*(11), 2969–2982.
Melechovsky, J., Prins, C., & Calvo, R. (2005). A metaheuristic to solve a location-routing problem with non-linear costs. *Journal of Heuristics*, *11*(5–6), 375–391.
Melián, B. (2006). Using memory to improve the VNS metaheuristic for the design of SDH/WDM networks. In *Lecture notes in computer science* (Vol. 4030, pp. 82–93). Berlin: Springer.
Melián, B., & Mladenović, N. (2007). Editorial. *IMA Journal of Management Mathematics*, *18*(2), 99–100.
Melián, B., Höller, H., & Voss, S. (2008). Designing WDM networks by a variable neighborhood search. *Journal of Telecommunications and Information Technology*, *4/2006*, 15–20.
Meric, L., Pesant, G., & Pierre, S. (2004). Variable neighborhood search for optical routing in networks using latin routers. *Annales des Télécommunications/Annals of Telecommunications*, *59*(3–4), 261–286.
Mitrovic-Minic, S., & Punnen, A. P. (2009). Local search intensified: very large-scale Variable Neighborhood Search for the multi-resource generalized assignment problem. *Discrete Optimization*. doi:10.1016/j.disopt.2009.04.004.
Mladenović, N. (1995). A variable neighborhood algorithm—a new metaheuristic for combinatorial optimization. Abstracts of papers presented at *Optimization days* (p. 112). Montréal.
Mladenović, N. (2005). Formulation space search—a new approach to optimization (plenary talk). In J. Vuleta (Ed.), *Proceedings of XXXII SYMOPIS'05* (pp. 3). Vrnjacka BanjA., Serbia.
Mladenović, N., & Hansen, P. (1997). Variable neighborhood search. *Computers and Operations Research*, *24*, 1097–1100.
Mladenović, N., & Urošević, D. (2003). Variable neighborhood search for the k-cardinality tree. *Applied Optimization*, *86*, 481–500.
Mladenović, N., Labbé, M., & Hansen, P. (2003a). Solving the p-center problem by tabu search and Variable Neighborhood Search. *Networks*, *42*, 48–64.
Mladenović, N., Petrović, J., Kovačević-Vujčić, V., & Čangalović, M. (2003b). Solving spread spectrum radar polyphase code design problem by tabu search and variable neighborhood search. *European Journal of Operational Research*, *151*, 389–399.
Mladenović, N., Plastria, F., & Urošević, D. (2005). Reformulation descent applied to circle packing problems. *Computers and Operations Research*, *32*, 2419–2434.
Mladenović, N., Brimberg, J., Hansen, P., & Moreno Pérez, J. A. (2007a). The p-median problem: a survey of metaheuristic approaches. *European Journal of Operational Research*, *179*(3), 927–939.
Mladenović, N., Plastria, F., & Uroševic, D. (2007b). Formulation space search for circle packing problems. In *Lecture notes on computer science* (Vol. 4638, pp. 212–216). Berlin: Springer.
Mladenović, N., Dražić, M., Kovačevic-Vujčić, V., & Čangalović, M. (2008). General variable neighborhood search for the continuous optimization. *European Journal of Operational Research*, *191*(3), 753–770.
Mladenović, N., Urošević, D., Pérez-Brito, D., & García-González, C. G. (2009). Variable neighbourhood search for bandwidth reduction. *European Journal of Operational Research*. doi:10.1016/j.ejor.2008.12.015.
Montemanni, R., & Smith, D. H. (2008). Construction of constant GC-content DNA codes via a Variable Neighbourhood Search algorithm. *Journal of Mathematical Modelling and Algorithms*, *7*(3), 311–326.
Moreno-Pérez, J. A., Moreno-Vega, J. M., & Rodríguez-Martín, I. (2003). Variable neighborhood tabu search and its application to the median cycle problem. *European Journal of Operational Research*, *151*(2), 365–378.
Moreno-Pérez, J. A., Hansen, P., & Mladenović, N. (2005). Parallel variable neighborhood search. In E. Alba (Ed.), *Parallel metaheuristics: a new class of algorithms*. New York: Wiley.
Moreno-Vega, J. M., & Melián, B. (2008). Introduction to the special issue on variable neighborhood search. *Journal of Heuristics*, *14*(5), 403–404.
Mori, H., & Tsunokawa, S. (2005). Variable neighborhood tabu search for capacitor placement in distribution systems. *IEEE International Symposium on Circuits and Systems*, *5*, 4747–4750.
Naderi, B., Zandieh, M., & Fatemi Ghomi, S. M. T. (2008). A study on integrating sequence dependent setup time flexible flow lines and preventive maintenance scheduling. *Journal of Intelligent Manufacturing*. doi:10.1007/s10845-008-0157-6.
Negreiros, M., & Palhano, A. (2006). The capacitated centred clustering problem. *Computers and Operations Research*, *33*(6), 1639–1663.
Nuortio, T., Kytöjoki, J., Niska, H., & Bräysy, O. (2006). Improved route planning and scheduling of waste collection and transport. *Expert Systems with Applications*, *30*(2), 223–232.
Ochi, L. S., Silva, M. B., & Drummond, L. (2001). Metaheuristics based on GRASP and VNS for solving traveling purchaser problem. In *MIC'2001* (pp. 489–494). Porto.

Ognjanović, Z., Midić, S., & Mladenović, N. (2005). A hybrid genetic and variable neighborhood descent for probabilistic SAT problem. In *Lecture notes in computer science* (Vol. 3636, pp. 42–53). Berlin: Springer.

Osman, I. H., & Ahmadi, S. (2007). Guided construction search metaheuristics for the capacitated p-median problem with single source constraint. *Journal of the Operational Research Society*, *58*(1), 100–114.

Pacheco, J., Casado, S., & Nuñez, L. (2007). Use of VNS and TS in classification: variable selection and determination of the linear discrimination function coefficients. *IMA Journal of Management Mathematics*, *18*(2), 191–206.

Pacheco, J. A., Casado, S., Alegre, J. F., & Álvarez, A. (2008). Heuristic solutions for locating health resources. *IEEE Intelligent Systems*, *23*(1), 57–63.

Pan, Q.-K., Wang, W.-H., & Zhu, J.-Y. (2007a). Some meta-heuristics for no-wait flow shop problem. *Computer Integrated Manufacturing Systems, CIMS*, *13*(5), 967–970.

Pan, Q.-K., Wang, W.-H., Zhu, J.-Y., & Zhao, B.-H. (2007b). Hybrid heuristics based on particle swarm optimization and variable neighborhood search for job shop scheduling. *Computer Integrated Manufacturing Systems, CIMS*, *13*(2), 323–328.

Papadimitriou, C. (1994). *Computational complexity*. Reading: Addison-Wesley.

Paraskevopoulos, D. C., Repoussis, P. P., Tarantilis, C. D., Ioannou, G., & Prastacos, G. P. (2008). A reactive variable neighborhood tabu search for the heterogeneous fleet routing problem with time windows. *Journal of Heuristics*, *14*(5), 425–455.

Parreño, F., Alvarez-Valdes, R., Oliveira, J. F., & Tamarit, J. M. (2008). Neighborhood structures for the container loading problem: a VNS implementation. *Journal of Heuristics*. doi:10.1007/s10732-008-9081-3.

Pelta, D., González, J. R., & Moreno-Vega, J. M. (2008). A simple and fast heuristic for protein structure comparison. *BMC Bioinformatics*, *9*, 161.

Pérez, M., Almeida Rodríguez, F., & Moreno-Vega, J. M. (2007). A hybrid VNS-path relinking for the p-hub median problem. *IMA Journal of Management Mathematics*, *18*(2), 157–172.

Plastria, F., Mladenović, N., & Urošević, D. (2005). Variable neighborhood formulation space search for circle packing. In *18th mini Euro conference VNS*. Tenerife, Spain.

Plastria, F., De Bruyne, S., & Carrizosa, E. (2009). Alternating local search based VNS for linear classification. *Annals of Operations Research*. doi:10.1007/s10479-009-0538-z.

Polacek, M., Hartl, R. F., Doerner, K., & Reimann, M. (2004). A variable neighborhood search for the multi depot vehicle routing problem with time windows. *Journal of Heuristics*, *10*(6), 613–627.

Polacek, M., Doerner, K. F., Hartl, R. F., Kiechle, G., & Reimann, M. (2007). Scheduling periodic customer visits for a traveling salesperson. *European Journal of Operational Research*, *179*(3), 823–837.

Polacek, M., Doerner, K. F., Hartl, R. F., & Maniezzo, V. (2008). A variable neighborhood search for the capacitated arc routing problem with intermediate facilities. *Journal of Heuristics*, *14*(5), 405–423.

Polo-Corpa, M. J., Salcedo-Sanz, S., Pérez-Bellido, A. M., López-Espí, P., Benavente, R., & Pérez, E. (2009). Curve fitting using heuristics and bio-inspired optimization algorithms for experimental data processing in chemistry. *Chemometrics and Intelligent Laboratory Systems*, *96*(1), 34–42.

Popper, K. (1959). *The logic of scientific discovery*. London: Hutchinson.

Prandtstetter, M., & Raidl, G. R. (2008). An integer linear programming approach and a hybrid variable neighborhood search for the car sequencing problem. *European Journal of Operational Research*, *191*(3), 1004–1022.

Puchinger, J., & Raidl, G. R. (2008). Bringing order into the neighborhoods: relaxation guided variable neighborhood search. *Journal of Heuristics*, *14*(5), 405–423.

Puchinger, J., Raidl, G. R., & Pferschy, U. (2006). The core concept for the multidimensional knapsack problem. In *Lecture notes in computer science* (Vol. 3906, pp. 195–208). Berlin: Springer.

Qian, B., Wang, L., Huang, D. X., & Wang, X. (2006). Multi-objective flow shop scheduling using differential evolution. In *Lecture notes in control and information sciences* (Vol. 345, pp. 1125–1136). Berlin: Springer.

Rahimi-Vahed, A., Dangchi, M., Rafiei, H., & Salimi, E. (2009). A novel hybrid multi-objective shuffled frog-leaping algorithm for a bi-criteria permutation flow shop scheduling problem. *International Journal of Advanced Manufacturing Technology*, *41*(11–12), 1227–1239.

Reeves, C. R. (Ed.) (1993). *Modern heuristic techniques for combinatorial problems*. Oxford: Blackwell Scientific.

Reinelt, G. (1991). TSLIB—A traveling salesman library. *ORSA Journal on Computing*, *3*, 376–384.

Remde, S., Cowling, P., Dahal, K., & Colledge, N. (2007). Exact/heuristic hybrids using rVNS and hyperheuristics for workforce scheduling. In *Lecture notes in computer science* (Vol. 4446, pp. 188–197). Berlin: Springer.

Repoussis, P. P., Paraskevopoulos, D. C., Tarantilis, C. D., & Ioannou, G. (2006). A reactive greedy randomized variable neighborhood tabu search for the vehicle routing problem with time windows. In *Lecture notes in computer science* (Vol. 4030, pp. 134–138). Berlin: Springer.

Repoussis, P. P., Tarantilis, C. D., & Ioannouo, G. (2007). A hybrid metaheuristic for a real life vehicle routing problem. In *Lecture notes in computer science* (Vol. 4310, pp. 247–254). Berlin: Springer.

Ribeiro, C. C., & de Souza, M. C. (2002). Variable neighborhood search for the degree-constrained minimum spanning tree problem. *Discrete Applied Mathematics*, *118*(1–2), 43–54.

Ribeiro, C. C., & Vianna, D. S. (2005). A GRASP/VND heuristic for the phylogeny problem using a new neighborhood structure. *International Transactions in Operational Research*, *12*(3), 325–338.

Ribeiro, C. C., Uchoa, E., & Werneck, R. (2002). A hybrid GRASP with perturbations for the Steiner problem in graphs. *INFORMS Journal on Computing*, *14*(3), 228–246.

Ribeiro, C. C., Martins, S. L., & Rosseti, I. (2007). Metaheuristics for optimization problems in computer communications. *Computer Communications*, *30*(4), 656–669.

Ribeiro, C. C., Aloise, D., Noronha, T. F., Rocha, C., & Urrutia, S. (2008a). A hybrid heuristic for a multi-objective real-life car sequencing problem with painting and assembly line constraints. *European Journal of Operational Research*, *191*(3), 981–992.

Ribeiro, C. C., Aloise, D., Noronha, T. F., Rocha, C., & Urrutia, S. (2008b). An efficient implementation of a VNS/ILS heuristic for a real-life car sequencing problem. *European Journal of Operational Research*, *191*(3), 596–611.

Roshanaei, V., Naderi, B., Jolai, F., & Khalili, M. (2009). A variable neighborhood search for job shop scheduling with set-up times to minimize makespan. *Future Generation Computer Systems*, *25*(6), 654–661.

Rousseau, L. M., Gendreau, M., & Pesant, G. (2002). Using constraint-based operators to solve the vehicle routing problem with time windows. *Journal of Heuristics*, *8*(1), 43–58.

Santana, R., Larrañaga, P., & Lozano, J. A. (2008). Combining variable neighborhood search and estimation of distribution algorithms in the protein side chain placement problem. *Journal of Heuristics*, *14*(5), 519–547.

Schilde, M., Doerner, K. F., Hartl, R. F., & Kiechle, G. (2009). Metaheuristics for the bi-objective orienteering problem. *Swarm Intelligence*, *3*(3), 179–201.

Schmid, V., Doerner, K. F., Hartl, R. F., & Salazar-González, J. J. (2008). Hybridization of very large neighborhood search for ready-mixed concrete delivery problems. *Computers and Operations Research*. doi:10.1016/j.cor.2008.07.010.

Sedlar, J., Vukičević, D., Aouchiche, M., & Hansen, P. (2007a). Variable neighborhood search for extremal graphs 24. Conjectures and results about the clique number. *Les Cahiers du GERAD G-2007-33*.

Sedlar, J., Vukičević, D., Aouchiche, M., & Hansen, P. (2007b). Variable neighborhood search for extremal graphs 25. Products of connectivity and distance measures. *Les Cahiers du GERAD G-2007-47*.

Sevkli, M., & Aydin, M. E. (2006a). A variable neighbourhood search algorithm for job shop scheduling problems. In *Lecture notes in computer science* (Vol. 3906, pp. 261–271). Berlin: Springer.

Sevkli, M., & Aydin, M. E. (2006b). Variable Neighbourhood Search for job shop scheduling problems. *Journal of Software*, *1*(2), 34–39.

Sevkli, M., & Aydin, M. E. (2007). Parallel variable neighbourhood search algorithms for job shop scheduling problems. *IMA Journal of Management Mathematics*, *18*(2), 117–134.

Sevkli, Z., & Sevilgen, F. E. (2006). Variable neighborhood search for the orienteering problem. In *Lecture notes in computer science* (Vol. 4263, pp. 134–143). Berlin: Springer.

Sevkli, Z., & Sevilgen, F. E. (2008). A hybrid particle swarm optimization algorithm for function optimization. In *Lecture notes in computer science* (Vol. 4974, pp. 585–595). Berlin: Springer.

Stevanovic, D., Aouchiche, M., & Hansen, P. (2008). On the spectral radius of graphs with a given domination number. *Linear Algebra and Its Applications*, *428*(8–9), 1854–1864.

Subramanian, A., & Dos Anjos Formiga Cabral, L. (2008). An ILS based heuristic for the vehicle routing problem with simultaneous pickup and delivery and time limit. In *Lecture notes in computer science* (Vol. 4972, pp. 135–146). Berlin: Springer.

Tagawa, K., Ohtani, T., Igaki, T., Seki, S., & Inoue, K. (2007). Robust optimum design of SAW filters by the penalty function method. *Electrical Engineering in Japan*, *158*(3), 45–54.

Tasgetiren, M. F., Sevkli, M., Liang, Y.-C., & Gencyilmaz, G. (2004). Particle swarm optimization algorithm for permutation flowshop sequencing problem. In *Lecture notes in computer science* (Vol. 3172, pp. 382–389). Berlin: Springer.

Tasgetiren, M. F., Liang, Y.-C., Sevkli, M., & Gencyilmaz, G. (2007). A particle swarm optimization algorithm for makespan and total flowtime minimization in the permutation flowshop sequencing problem. *European Journal of Operational Research*, *177*(3), 1930–1947.

Tavakkoli-Moghaddam, R., Safaei, N., & Sassani, F. (2009). A memetic algorithm for the flexible flow line scheduling problem with processor blocking. *Computers and Operations Research*, *36*(2), 402–414.

Toksari, A. D., & Güner, E. (2007). Solving the unconstrained optimization problem by a variable neighborhood search. *Journal of Mathematical Analysis and Applications*, *328*(2), 1178–1187.

Tseng, C.-T., Liao, C.-J., & Huang, K.-L. (2009). Minimizing total tardiness on a single machine with controllable processing times. *Computers and Operations Research*, *36*(6), 1852–1858.

Urošević, D., Brimberg, J., & Mladenović, N. (2004). Variable neighborhood decomposition search for the edge weighted k-cardinality tree problem. *Computers and Operations Research*, *31*(8), 1205–1213.

Vogt, L., Poojari, C. A., & Beasley, J. E. (2007). A tabu search algorithm for the single vehicle routing allocation problem. *Journal of the Operational Research Society*, *58*, 467–480.

Wang, X., & Tang, L. (2009). A population-based variable neighborhood search for the single machine total weighted tardiness problem. *Computers and Operations Research*, *36*(6), 2105–2110.

Whitaker, R. (1983). A fast algorithm for the greedy interchange of large-scale clustering and median location problems. *INFOR*, *21*, 95–108.

Wollenweber, J. (2008). A multi-stage facility location problem with staircase costs and splitting of commodities: model, heuristic approach and application. *OR Spectrum*, *30*(4), 655–673.

Xhafa, F. (2007). A hybrid evolutionary heuristic for job scheduling on computational grids. *Studies in Computational Intelligence*, *75*, 269–311.

Yang, J., Zhang, J., Aydin, M. E., & Wu, J. Y. (2007). A novel programming model and optimisation algorithms for WCDMA networks. In *IEEE vehicular technology conference* (pp. 1182–1187).

Yepes, V., & Medina, J. (2006). Economic heuristic optimization for heterogeneous fleet VRPHESTW. *Journal of Transportation engineering*, *132*(4), 303–311.

Zhang, C., Lin, Z., & Lin, Z. (2005). Variable neighborhood search with permutation distance for QAP. In *Lecture notes in computer science* (Vol. 3684, pp. 81–88). Berlin: Springer.

Zhao, Q. H., Chen, S., & Zang, C. Y. (2008). Model and algorithm for inventory/routing decision in a three-echelon logistics system. *European Journal of Operational Research*, *191*(3), 627–635.

Zobolas, G. I., Tarantilis, C. D., & Ioannou, G. (2009a). Minimizing makespan in permutation flow shop scheduling problems using a hybrid metaheuristic algorithm. *Computers and Operations Research*, *36*(4), 1249–1267.

Zobolas, G. I., Tarantilis, C. D., & Ioannou, G. (2009b). A hybrid evolutionary algorithm for the job shop scheduling problem. *Journal of the Operational Research Society*, *60*(2), 221–235.