Optim Eng (2008) 9: 119–142
DOI 10.1007/s11081-007-9023-1

Optimal sensor placement for enhancing sensitivity to change in stiffness for structural health monitoring

Josh M. Beal · Amit Shukla · Olga A. Brezhneva · Mark A. Abramson

Received: 14 April 2006 / Accepted: 27 June 2007 / Published online: 17 August 2007

Abstract This paper focuses on optimal sensor placement for structural health monitoring (SHM), in which the goal is to find an optimal configuration of sensors that will best predict structural damage. The problem is formulated as a bound constrained mixed variable programming (MVP) problem, in which the discrete variables are categorical; i.e., they may only take on values from a pre-defined list. The problem is particularly challenging because the objective function is computationally expensive to evaluate and first-order derivatives may not be available. The problem is solved numerically using the generalized mixed variable pattern search (MVPS) algorithm. Some new theoretical convergence results are proved, and numerical results are presented, which show the potential of our approach.

Keywords Mixed variable programming · Optimal sensor placement · Structural health monitoring

J.M. Beal
Department of Mathematics, Ohio University, Morton 321, Athens, OH 45701, USA

A. Shukla (✉)
Department of Mechanical and Manufacturing Engineering, 56 Engineering Building, School of Engineering & Applied Science, Miami University, Oxford, OH 45056, USA
e-mail: shuklaa@muohio.edu

O.A. Brezhneva
Department of Mathematics and Statistics, Miami University, 123 Bachelor Hall, Oxford, OH 45056, USA
e-mail: brezhnoa@muohio.edu

M.A. Abramson
Department of Mathematics and Statistics, Air Force Institute of Technology, 2950 Hobson Way, Wright-Patterson AFB, OH 45433, USA
e-mail: mark.abramson@afit.edu

1 Introduction

The area of health monitoring and non-destructive evaluation of structures has received much attention in the past two decades mainly because of its numerous applications, ranging from civil to transportation infrastructure. Since structural damage is a cumulative phenomenon, any local damage can eventually affect overall structural integrity. This motivates the need for continuous monitoring and tracking of damage, which is often done by integrating sensing and actuation systems *a priori* into the system design to create active and adaptive structural systems. This will hopefully lead to the development of self-healing systems.

The quality of structural health monitoring (SHM) essentially depends on the placement of the sensors. For example, if sensors are concentrated primarily in one area of the structure, then it becomes more likely that damage and other system changes go dangerously unnoticed in the areas with less sensor coverage. On the other hand, sensor placement is also guided by cost and data post-processing issues required for health monitoring decisions.

Sensor placement is typically done either *ad hoc*, or, at best, by experimentally testing a few possible combinations and choosing the one that performs the best, based on a few benchmark examples. In this case, sensor placement relies heavily on institutional knowledge and specific experience of those who place the sensors. Although certainly helpful, this approach can overlook more advantageous designs.

If sensor placement can be accurately modeled into an optimization problem that can be solved numerically, then the optimization process can account for this in finding a suitable design, so that sufficient coverage can be achieved. Given a limited number of sensors, the problem becomes one of determining the optimal number and locations of sensors, with respect to a suitable objective function, such as cost or some measure of sensor coverage.

Complicating the optimization process is that, in many cases (such as those described in Doebling et al. 1996), each objective function evaluation requires either a large finite element analysis or the acquisition of experimental data, both of which can be computationally expensive and time-consuming. To circumvent this issue, simple models are developed, which are then utilized to compare various objective function values to guide the optimization process.

Various researchers have developed highly multi-disciplinary tools and methods to detect, quantify and eventually predict damage phenomenon. A Los Alamos National Laboratory report (Doebling et al. 1996) contains a detailed overview of existing techniques and methods used to evaluate and monitor the health of structures and systems. The general optimal sensor placement problem has been studied in many different contexts, including noise attenuation and vibration control (Padula et al. 1998; Padula and Kincaid 1999), system identification (Kincaid and Padula 2002), radiation source failure detection (Chaudhuri and Ghosh 2000), positioning of actuators and sensors in a structure (Lopes et al. 2004), and placement of strain sensors on the flexible supporting structure for long reach manipulators (Mavroidis et al. 1997).

Problems specifically treating optimal sensor placement for damage detection have also been studied. Staszewski et al. (2000) study the problem of optimal sensor placement for impact detection and location in composite materials. Papadimitriou et

al. (2000) also discuss optimal placement strategies for structural damage identification. Specifically, the study describes the issues involved in constructing an optimal instrumentation. Guo et al. (2004) study the global optimization of sensor locations for SHM. Their work differs from the present one both in the problem formulation and in the numerical method employed to solve the problem. In Sect. 3 we will provide proper motivation for our choice of method.

The main goal of the work described here is to develop, implement, and apply a *black box* optimization algorithm specifically for optimal design and placement of sensors that will guide future work in SHM.

The paper is organized as follows. The problem definition and modeling details are presented in Sect. 2. In Sect. 3, we describe the mixed variable generalized pattern search algorithm and justify its use in numerically solving the sensor placement problem. This includes a new stronger convergence theory for mixed variable optimization problems having one continuous variable. Numerical results and concluding remarks are presented in Sects. 4 and 5, respectively.

2 Modeling the optimal sensor placement problem

Modal parameters, such as frequencies, mode shapes, and modal damping, are functions of the physical properties of the structure (e.g., mass, damping, and stiffness). Therefore, changes in physical properties will cause changes in the modal properties. Ideally, a robust damage detection scheme will be able to identify that damage has occurred at a very early stage, locate the damage within the sensor resolution being used, provide some estimate of the severity of the damage, and predict the remaining useful life of the structure. Current damage detection methodologies use an initial measurement of an undamaged structure as the baseline for future comparisons of measured response. Another important feature of damage identification methods, and specifically those methods which use prior models, is their ability to discriminate between model/data discrepancies caused by modeling errors and discrepancies that are a result of structural damage.

The observation that changes in structural properties cause changes in vibration frequencies was the impetus for using modal methods for damage identification and health monitoring. The somewhat low sensitivity of frequency shifts to damage requires either very precise measurements or large levels of damage.

Also, because modal frequencies are a global property of the structure, it is not clear that shifts in this parameter can be used to identify more than the mere existence of damage. In other words, frequencies generally cannot provide spatial information about structural changes. Damage detection methods also utilize the mode shape changes. Mode shapes can be partitioned using various schemes, and the change in transfer functions across the different partitioning techniques can be used to localize the structural damage. As observed in Doebling et al. (1996), graphical comparisons of relative changes in mode shapes proved to be the best way of detecting the damage location when only resonant frequencies and mode shapes were examined.

In this paper we study an $N = 20$ degree-of-freedom (DOF) *mass-spring-damper* system, illustrated in Fig. 1, that can be modeled by the system of linear second-order

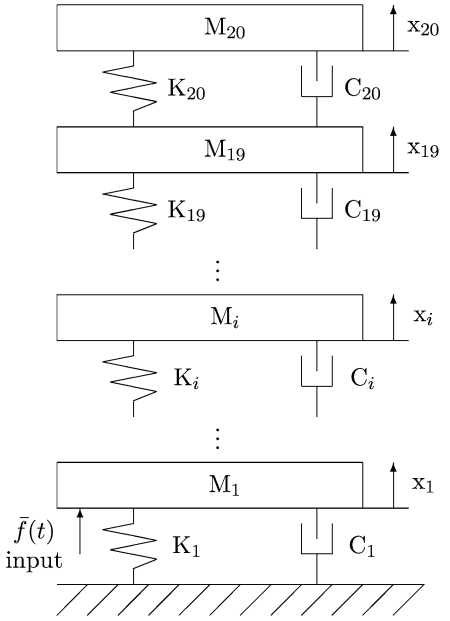

Fig. 1 Schematic of a 20-DOF Mass Spring Damper System

differential equations,

$$\mathbf{M}\ddot{\mathbf{x}} + \mathbf{C}\dot{\mathbf{x}} + \mathbf{K}\mathbf{x} = 0, \tag{1}$$

where $\mathbf{x} \in \mathbb{R}^N$ denotes the vertical positions of the masses, and where the mass $\mathbf{M}$, spring stiffness $\mathbf{K}$, and damping coefficient $\mathbf{C}$ are $N \times N$ matrices of system parameters. Nominal system parameter values are given in Sect. 4. We define *damage* as any change in the stiffness parameters associated with the system model.

Our problem involves positioning sensors on a fixed number $n < N$ of masses (1 DOF each), where N represents the total number of masses. Each mass is connected by a spring and damper, and we restrict the motion of the masses so that they may only move in the vertical direction.

Let $\bar{f}(t)$ be the *input* (force) and $\bar{a}(t)$ be the *output* (acceleration) to a general linear time-invariant system. We denote the Laplace transforms of $\bar{f}(t)$ and $\bar{a}(t)$ by $F(s)$ and $A(s)$, respectively. Then the output and input are related by the *transfer function*,

$$H(s) = \frac{A(s)}{F(s)}. \tag{2}$$

Thus, in describing how the output varies with the input, the transfer function provides information about the system dynamics. For the example considered in this paper, a transfer function can be calculated for each input and output pair located at any of the N degrees-of-freedom. In other cases where such models are not available, experimental data acquisition and post-processing can be employed to calculate the associated transfer function between the input and output locations. Furthermore, a transfer function can be easily calculated for continuous structures if the finite el-

ement model is known. However, in both of these cases, the optimization process would require large computational or experimental resources.

We also define a *sensor* as any data measurement available at the system under consideration. A system model could have degrees of freedom, some of which can be measured and some of which cannot. A *measurement degree of freedom* is one that can be measured with a sensor. We classify a model of the system as either full or reduced. The *full model* includes measurements available at *all* the relevant degrees of freedom. The *reduced model* includes only those degrees of freedom that can be measured. The reduced model is generally a subset of the full model guided by the observability of measured degrees-of-freedom.

2.1 Mixed variable problem formulation

The optimal sensor placement problem is often formulated (e.g., see Lopes et al. 2004) as a binary mixed integer nonlinear programming (MINLP) problem, where each possible sensor location is represented by a binary variable. Many traditional optimization methods, as well as heuristics have been employed in an attempt to solve this class of problems. In fact, a review of these methods as applied to sensor placement can be found in Guo et al. (2004).

However, one difficulty with the MINLP formulation is that, if the set of possible sensor locations is large, problem size becomes a serious issue. In this paper, we instead formulate the problem of interest as a mixed variable programming (MVP) problem. MVP problems differ from MINLP problems in that the discrete variables may be categorical, meaning that they must always take their values from a predefined list or set. Categorical variables can even be nonnumeric, such as color, shape, or type of material. For a sensor placement problem, rather than using a binary variable for each possible sensor location (which indicates whether or not a sensor is at that location), we define for each sensor a categorical variable containing the position of that sensor.

The objective function f used for SHM is related to the modal properties of the system under consideration when vibration-based data is used. Two examples, as discussed above, are changes in natural frequencies and changes in mode shapes. Another related measure which could be used is change in the overall transfer functions (within a frequency band). The evaluation of $f(p, M)$, where $p \in [0, 1]$ is a measurement of percentage change in the stiffness of the structure and M is the set of sensor locations, generally requires a finite element analysis or an experimental observation, and is thus very expensive to compute. Furthermore, percentage change in stiffness p is a result of damage induction. This is certainly known for damage phenomenon related to damage of joints and development of cracks in structures. In most cases, we do not know (or it is very expensive to evaluate) the derivatives of the objective function. Our approach is guided by real cases in which they would not be available. We also assume that the objective function is a nonsmooth function in terms of the damage induced.

It should be noted that, even though the transfer function is not a very sensitive measure of the change in stiffness, the goal of this work is to demonstrate the role of pattern search methods in the selection of optimal sensor placement locations. Any

other measure which is readily available (such as any higher order frequency domain or time domain metrics) via experimental analysis or numerical simulations can be incorporated into this framework without any loss of generality.

One feature of successful SHM technology in the future would be the ability to detect the onset of damage caused by crack initiation and its propagation at the earliest possible stage. This should be possible because damage of this kind leads to local changes in the inherent stiffness of the structure, which can be detected by sensors. Thus the optimal sensor configuration problem not only requires sensor placement with the best ability to detect damage, but also with the ability to detect damage at an early stage.

In this paper, the design variables not only include the sensor placement locations M, but also the percentage change in the stiffness p. From an engineering perspective, stiffness is an intrinsic property of the system, which changes as a result of damage. Hence, a small value of p corresponds to the small changes in stiffness that usually occur in the earlier stages of system damage. In modeling p as a variable in the optimization process, our goal is still to find the sensor locations that best detect the damage, but solutions with a lower value of p mean that the amount of damage is less; i.e., it is detected in the early stages.

An optimal solution with a large value of p is not necessarily desirable because it means that damage may not be detected until it is critical. Nevertheless, the optimal solution found as $\min f(p, M)$ shows us how well we can detect damage with a given number of sensors, compared to how well we could do it with sensors in all mass locations. We can use this formulation to then find sensor placements that yield good but perhaps suboptimal solutions that also have a small value of p, thus allowing us to detect the damage with good enough quality at an early stage.

From an optimization perspective, we could add a constraint on p to keep it sufficiently small, but we decided not to place any artificial limitations on p, since such a constraint would be application-specific. Moreover, running numerical simulations without tight bounds on the variable p allows us to collect statistical data that give insight into the system characteristics we are trying to analyze. If $\min f(p, M)$ is not small enough, then it means we need more sensors to detect the damage. If $\min f(p, M)$ is small, but p is not, then the number of sensors is not large enough to detect the damage at an early stage.

Transfer functions are defined for specific combinations of input and output locations. If the input location is held fixed, then the possible number of transfer functions is equal to the number of degrees-of-freedom for the discrete system. Thus, a sensor can be ideally placed at each mass to measure the corresponding transfer function. Unfortunately, this is not always possible due to lack of resources and/or the continuous nature of the systems under investigation. In the problem of sensor placement, damage is modeled as a loss of stiffness to the springs. Namely, we will simulate the damage by changing a particular spring constant j. Since we limit the number of sensors by $n < N$, our knowledge of damage to the system will be limited to the information we gather from n sensors (n transfer functions).

Let $M \subset \{1, 2, \ldots, N\}$ be a nonempty proper subset of masses where sensors could be placed, such that its cardinality is $|M| = n$. By definition, M cannot contain duplicate entries, which enforces the constraint that no more than one sensor can be

placed on a particular mass. Since the variable $p \in \mathbb{R}$ represents a percentage change, it follows that $p \in [0, 1]$.

In formulating the objective function, we wish to place n sensors that elicit the most information in a "best way", or equivalently, pick the "best" n transfer functions that contain the most information about the effect of change in stiffness on the modal properties. It seems appropriate that this "best" placement would minimize the difference between transfer function information gathered from the full model (which we assume we have initially) and that gathered from the reduced model.

For the damaged spring j, let k_{1j} and k_{2j} denote the initial and changed spring constants, respectively, and denote the corresponding ith transfer functions by $(H_i(s; p))_{k_{1j}}$ and $(H_i(s; p))_{k_{2j}}$, respectively, for $i = 1, 2, \ldots, N$. Aggregate relative changes are then measured as sums of the relative changes in transfer function values. The true relative change $T_V(p)$ is measured as a sum over all possible output locations, while the approximate relative change $A_V(p)$ is measured as a sum over only those output locations with sensors placed there; i.e.,

$$T_V(p) = \sum_{i=1}^{N} \left| \frac{(H_i(s; p))_{k_{1j}} - (H_i(s; p))_{k_{2j}}}{(H_i(s; p))_{k_{1j}}} \right|,$$

$$A_V(p, M) = \sum_{i \in M \neq \emptyset} \left| \frac{(H_i(s; p))_{k_{1j}} - (H_i(s; p))_{k_{2j}}}{(H_i(s; p))_{k_{1j}}} \right|.$$

Hence, the absolute approximation error is $|T_V(p) - A_V(p, M)|$.

For the system under consideration, assuming that $T_V(p) \neq 0$, we define the objective function f as the relative approximation error between the true and approximate relative changes; namely,

$$f(p, M) = \frac{|T_V(p) - A_V(p, M)|}{|T_V(p)|}. \tag{3}$$

The motivation for this formulation is in the observation that only a few transfer functions dominate the net effect of any particular change in stiffness due to damage, and we optimize the sensor placement to best capture those changes.

Our MVP formulation now becomes

$$\begin{aligned} \min_{p,M} \quad & f(p, M) = \frac{|T_V(p) - A_V(p, M)|}{|T_V(p)|}, \\ \text{subject to} \quad & M \subset \{1, 2, \ldots, N\}, \\ & 0 \leq p \leq 1. \end{aligned} \tag{4}$$

3 Mixed variable optimization of the sensor placement problem

Because of the mixed variable nature of our problem, traditional optimization approaches, such as those reviewed in Guo et al. (2004) are not applicable. Certainly, commonly used heuristics, such as genetic algorithms, could be applied. In fact, they

are often quite useful in practice. However, heuristics generally lack any formal convergence theory, which means that, while improvement in the objective function can be achieved, there are no guarantees of achieving proximity to an optimal point.

Thus, we turn our attention to algorithms specifically designed for this class of problems. To date, very few algorithms exist for solving MVP problems. In fact, other than two general frameworks, one derivative-free (Lucidi et al. 2005) and the other derivative-based (Lucidi and Piccialli 2004), the only current work in this area makes use of pattern search methods (Audet and Dennis 2000; Abramson 2002, 2004; Abramson et al. 2004; Kokkolaras et al. 2001; Sriver et al. 2005).

In order to fully describe the pattern search algorithm that we apply to this problem, we will consider a more general case of mixed variable problem that has more than one continuous variable and a finite number of bound and linear constraints (with respect to the continuous variables). Each configuration of sensors is represented as a trial point $x = (x^c, x^d) \in X = X^c \times X^d$, partitioned into its continuous component $x^c \in X^c \subset \mathbb{R}^{n^c}$ and discrete component $x^d \in X^d \subset \mathbb{Z}^{n^d}$, where n^c and n^d are the number of continuous and discrete variables, respectively. (Note that, for this section only, we have redefined the variable x here.) A more general sensor problem can then be represented mathematically as

$$\begin{aligned} &\min_{x \in X} \quad f(x), \\ &\text{subject to} \quad \ell \leq Ax^c \leq u, \end{aligned} \tag{5}$$

where the objective function $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ is nonlinear, $A \in \mathbb{R}^{m \times n^c}$, $\ell \in (\mathbb{R} \cup \{-\infty\})^m$, and $u \in (\mathbb{R} \cup \{+\infty\})^m$, with $\ell < u$.

3.1 Discrete neighbor sets and functions

For MVP problems, finding a global minimizer is very often an intractable problem because it would require exhaustive enumeration of all possible sets of discrete variable values. On the other hand, the concept of a local minimizer in a mixed variable domain is not well-defined because categorical variables do not necessarily have an inherent natural ordering. In order to define what we mean by a local minimum, we need a notion of a local neighborhood. We do this by means of a set-valued function $\mathcal{N} : X \to 2^X$, where X is the mixed variable domain defined above and 2^X denotes the power set of X (the set of all subsets of X). Given a point $x \in X$, the point $y \in X$ is a *discrete neighbor* of x if $y \in \mathcal{N}(x)$. We assume that, for all $x \in X$, $\mathcal{N}(x)$ is finite, and by convention, $x \in \mathcal{N}(x)$. We also assume a notion of continuity in the sets of neighbors in the sense that, for any sequence x_k and corresponding sequence of discrete neighbors $y_k \in \mathcal{N}(x_k)$ converging to $\hat{x}$ and $\hat{y}$, respectively, we have that $\hat{y} \in \mathcal{N}(\hat{x})$.

We now define a local minimum as follows (Audet and Dennis 2000; Abramson 2002; Abramson et al. 2004):

Definition 1 A point $x = (x^c, x^d) \in X$ is said to be a *local minimizer of f with respect to the set of neighbors $\mathcal{N}(x) \subset X$* if there exists an $\epsilon > 0$ such that $f(x) \leq$

$f(v)$ for all v in the set

$$X \cap \left(\bigcup_{y \in \mathcal{N}(x)} (B(y^c, \varepsilon) \times \{y^d\}) \right). \tag{6}$$

The notation $B(y^c, \varepsilon)$ refers to an open ball of radius $\varepsilon > 0$ (or ε-neighborhood), centered at y^c. Note that Definition 1 is stated not only in terms of ε-neighborhoods and discrete neighbors of x, but also in terms of ε-neighborhoods of every discrete neighbor of x.

As an illustrative example, Kokkolaras et al. (2001) studied the optimization of a thermal insulation system, consisting of hot and cold surfaces at the ends, with a series of insulators between them, each pair of which is separated by a heat intercept. The objective was to minimize the power required to keep one of the surfaces at the correct temperature. The discrete neighbor function was constructed so that, given a configuration of insulators and heat intercepts, a discrete neighbor is any design in which a single insulator is replaced with one of a different type, a single insulator and adjacent intercept are removed, or a single insulator and adjacent intercept are added by inserting them at any location in the system.

3.2 GPS algorithm for mixed variable optimization

The derivative-free class of pattern search algorithms was originally introduced by Torczon (1997) for unconstrained continuous optimization problems and extended by Lewis and Torczon to bound constrained (1999) and linearly constrained (2000) problems. In all these cases, convergence to a first-order stationary point was established, given sufficient smoothness of the objective function. Audet and Dennis (2003) applied the Clarke (1983) nonsmooth calculus to establish a hierarchy of convergence results that strengthens the previous theory. A thorough review of this class of methods is given in Kolda et al. (2003). Generalizations for nonlinearly constrained problems have been introduced by Lewis and Torczon (2002) and by Audet and Dennis (2004, 2006).

The class of pattern search algorithms for bound constrained MVP problems was introduced by Audet and Dennis (2000) (as an extension of Lewis and Torczon 1999). This algorithm was used to optimize the design of a thermal insulation system (Kokkolaras et al. 2001). The algorithm has been extended to linear (Abramson 2002) and nonlinear (Abramson 2002; Abramson et al. 2004) constraints, and was used to solve a modification of the problem in Kokkolaras et al. (2001), in which constraints on mass and thermal contraction were added (Abramson 2004). Sriver et al. (2005) have introduced a pattern search ranking and selection algorithm for linearly constrained stochastic MVP problems.

Pattern search algorithms generate a sequence of iterates with nonincreasing function values. Each iteration k is characterized by two phases: an optional SEARCH step, and a local POLL step. In both steps, trial points on a mesh are generated in an attempt to find a point with a lower function value than that of the current iterate $x_k = (x_k^c, x_k^d)$. Actually, the algorithm is not applied to f, but to f_X, defined by $f_X(x) = f(x)$ if $x \in X$ and $f_X(x) = \infty$ for $x \notin X$.

The mesh is constructed as the direct product of the discrete variable space X^d with the union of a finite number of lattices in the continuous variable space X^c, but translated from the current iterate; i.e.,

$$M_k = X^d \times \bigcup_{x^d \in X^d} \{x_k^c + \Delta_k D(x^d) z \in X^c : z \in \mathbb{Z}_+^{n_D}\}, \tag{7}$$

where the mesh size parameter $\Delta_k > 0$ controls the fineness of the mesh, and $D(x^d)$ denotes an $n \times n_D$ matrix, where $n_D = |D(x^d)| > n$, whose columns positively span the continuous variable space. Often, $D(x^d)$ is chosen as $[I, -I]$ or $[I, -e]$, where I is the identity matrix and e is the vector of ones.

The goal of the SEARCH step is to adequately sample the space in an attempt to quickly identify a promising region containing a good local minimizer. The only restriction on this step is that it evaluates a finite number of points on the mesh. How these points are selected is completely flexible. In fact, the SEARCH step may range from empty (not evaluating any points) to the construction and optimization of a less costly surrogate function at every iteration (see Booker et al. 1998, 1999, for example). Random sampling strategies and heuristics, such as Latin hypercubes, orthogonal arrays, and a few generations of a genetic algorithm, can also be used.

If the SEARCH step fails to find a better point, the POLL step is performed, in which the adjacent points on the mesh are evaluated. This step is necessary to ensure convergence of a subsequence of iterates to a limit point satisfying certain necessary conditions for optimality. For MVP problems, the POLL step also includes evaluating the discrete neighbors of the current iterate x_k. Thus the POLL step includes evaluation of the points in the union of the continuous mesh neighbors $P(x_k)$ of x_k with the user-defined set of discrete neighbor points $\mathcal{N}(x_k)$ of x_k. The continuous mesh neighbors of a point $x \in X$ can be expressed as

$$P_k(x) = \{(x^c + \Delta_k d, x^d) \in X : d \in D_k(x)\}, \tag{8}$$

where $D_k(x) \subseteq D(x^d)$ is the set of poll directions at iteration k, which also positively spans the continuous space.

In the case of bound or linear constraints, directions D must be chosen so that they include all possible tangent cone generators, and at each iteration, the choice of $D_k(x_k)$ must be chosen to conform to the geometry of the nearby constraints (Audet and Dennis 2003). Lewis and Torczon (2000) provide an algorithm for computing these directions using simple linear algebra tools. However, for problems with only one continuous variable, these conditions are automatically satisfied by any positive spanning set.

If neither the SEARCH or POLL succeeds in finding an improved mesh point, then an EXTENDED POLL step is performed about any discrete neighbor whose objective function value is sufficiently close to that of the incumbent. Specifically, given a fixed scalar $\xi > 0$ and an *extended poll trigger* $\xi_k \geq \xi$, an EXTENDED POLL is performed with respect to $y_k \in \mathcal{N}(x_k)$ if the condition, $f_X(x_k) \leq f_X(y_k) \leq f_X(x_k) + \xi_k$, is met. For any such y_k, we begin a sequence of polls about the points $\{y_k^j\}_{j=1}^{J_k}$, beginning with $y_k^0 = y_k$ and ending when either $f(y_k^{J_k} + \Delta_k(d, 0)) < f(x_k)$ for some

$d \in D_k(y_k^{J_k})$, or when $f_X(y_k^{J_k}) \leq f_X(y_k^{J_k} + \Delta_k(d, 0))$ for all $d \in D_k(y_k^{J_k})$. The notation $(d, 0)$ indicates change in the direction d with respect to the continuous variables while holding the discrete variables constant. The number J_k simply represents how many EXTENDED POLL steps are executed, and is not known in advance. However, under very mild assumptions, J_k is always finite (Audet and Dennis 2000).

In practice, the extended poll trigger ξ_k is typically set as a percentage of the objective function value, but bounded away from zero (e.g., $\xi_k = \max\{\xi, 0.05|f(x)|\}$). A relatively high choice of ξ_k tends to make the solution more global (since it results in extended polls around more discrete neighbor points), but at a higher computational cost.

The set of extended poll points evaluated for a particular discrete neighbor $y \in \mathcal{N}(x_k)$ contains a subset of the points $\{P_k(y_k^j)\}_{j=1}^{J_k}$. More precisely, at iteration k, the set of points evaluated in the EXTENDED POLL step (or *extended poll set*) is given by

$$\mathcal{X}_k(\xi_k) = \bigcup_{y_k \in \mathcal{N}_k^{\xi_k}} \left(\bigcup_{j=1}^{J_k} P_k(y_k^j) \right), \tag{9}$$

where $\mathcal{N}_k^{\xi_k} = \{y \in \mathcal{N}(x_k) : f_X(x_k) \leq f_X(y_k) \leq f_X(x_k) + \xi_k\}$.

The algorithm is opportunistic, in that it can move to a new iterate as soon as improvement in the objective function is found, in which case, the iteration ends and the mesh is either retained or coarsened. On the other hand, if the SEARCH, POLL, or EXTENDED POLL steps all fail to find an improved mesh point, then the current iterate x_k is declared to be a *mesh local optimizer* and retained as the current iterate, and the mesh is refined.

Rules for updating the mesh are as follows. Given a fixed rational number $\tau > 1$ and two integers $w^- \leq 1$ and $w^+ \geq 0$, the mesh size parameter Δ_k^m is updated according to the rule,

$$\Delta_{k+1}^m = \tau^{w_k} \Delta_k^m, \tag{10}$$

where

$$w_k \in \begin{cases} \{0, 1, \ldots, w^+\} & \text{if improved mesh point is found,} \\ \{w^-, w^- + 1, \ldots, -1\} & \text{otherwise.} \end{cases} \tag{11}$$

The MVPS algorithm is presented formally in Fig. 2.

3.3 Convergence analysis

Convergence results for bound constrained MVP problems were established in Audet and Dennis (2000) under the assumption that the function f is continuously differentiable (with respect to the continuous variables) near limit points of certain subsequences of the algorithm. This analysis was extended in Abramson (2002) to linearly constrained MVP problems in which the local smoothness of f can be relaxed.

In this section, we establish new results for MVP problems with one continuous variable, which act as corollaries to the theorems of Audet and Dennis (2000) and

Generalized Mixed Variable Pattern Search – MVPS

Initialization: Let $x_0 \in X$ satisfy $f_X(x_0) < \infty$. Set $\Delta_0 > 0$ and $\xi > 0$.

For $k = 0, 1, 2, \ldots$, perform the following:

1. Set extended poll trigger $\xi_k \geq \xi$.
2. SEARCH **step:** Employ some finite strategy seeking an improved mesh point; i.e., $x_{k+1} \in M_k$ such that $f_X(x_{k+1}) < f_X(x_k)$.
3. POLL **step:** If the SEARCH step does not find an improved mesh point, evaluate f at points in $P_k(x_k) \cup \mathcal{N}(x_k)$ until an improved mesh point x_{k+1} is found (or until all points are exhausted).
4. EXTENDED POLL **step:** If the SEARCH and POLL steps do not find an improved mesh point, evaluate f at points in $\mathcal{X}_k(\xi_k)$ (see (9)) until an improved mesh point x_{k+1} is found (or until all points are exhausted).
5. **Update:** If SEARCH, POLL, or EXTENDED POLL finds an improved mesh point,
   Update x_{k+1}, and set $\Delta_{k+1} \geq \Delta_k$ according to (10)–(11);
   Otherwise, set $x_{k+1} = x_k$, and set $\Delta_{k+1} < \Delta_k$ according to (10)–(11).

Fig. 2 MVPS Algorithm

Abramson (2002) (and also Abramson 2005), and which apply specifically to the problem posed in (4). In particular, the first-order and second-order directional optimality results in Audet and Dennis (2000) and Abramson (2005), respectively, apply to *all* directions in $\mathbb{R}$ because there are only two normalized directions in $\mathbb{R}$, thus enabling stronger results under the same hypotheses. This theory not only applies to MVP problems, but also to single variable NLP problems, for which there are plenty of applications—most notably, line searches.

The analysis in the remainder of this section requires that the following assumptions hold:

A1. An initial feasible point $x_0 \in X$ satisfying $f(x_0) < \infty$ is available.
A2. All MVPS iterates $\{x_k\}$ lie in a compact set.

Audet and Dennis (2000) proved the existence of a convergent *refining subsequence* $\{x_k\}_{k \in K}$ (for some set of indices K), with an associated positive spanning set of *refining directions* such that, under assumptions A1–A2, satisfies $\lim_{k \in K} \Delta_k = 0$ and $\hat{x} = \lim_{k \in K} x_k$. Furthermore, under the previously assumed notion of continuity of $\mathcal{N}$, for each subsequence of discrete neighbors $\{y_k\}_{k \in K}$ satisfying $y_k \in \mathcal{N}(x_k)$, there exist limit points $\hat{y} = \lim_{k \in K} y_k$ and $\hat{z} = \lim_{k \in K} z_k$, where $\hat{y} \in \mathcal{N}(\hat{x})$ and $z_k = y_k^{J_k}$ is the EXTENDED POLL endpoint corresponding to y_k at iteration k. This notation is used throughout the remainder of this section.

The following result from Abramson (2002) establishes local optimality with respect to the set of discrete neighbors.

Theorem 2 *If f is lower semi-continuous at $\hat{x}$ and continuous at $\hat{y}$ with respect to the continuous variables, then $f(\hat{x}) \leq f(\hat{y})$.*

The next two results show that the limit points $\hat{x}$ and $\hat{z}$ satisfy a nonsmooth first-order necessary condition for optimality with respect to the continuous variables. They are based on the Clarke (1983) generalized directional derivative, which is defined for a Lipschitz continuous function g at $x \in \mathbb{R}^n$ and in the direction $d \in \mathbb{R}^n$ by

$$g^\circ(x; d) \equiv \limsup_{y \to x,\ t \downarrow 0} \frac{g(y + td) - g(y)}{t}.$$

We note that Theorem 3 is a corollary to Theorems 3.7 of Audet and Dennis (2003) and 5.14 of Abramson (2002), while Theorem 4 is a corollary to Theorem 5.15 in Abramson (2002).

Theorem 3 *If f is Lipschitz continuous near $\hat{x} \in X$ with respect to the continuous variables, then $f^\circ(\hat{x}; (v, 0)) \geq 0$ for all feasible directions $v \in \mathbb{R}$.*

Proof In $\mathbb{R}$, a set of refining directions must necessarily include a set $\{d_1, -d_2 : d_1, d_2 > 0\}$. Let $v \in \mathbb{R}$ be a feasible direction. If $v = 0$, then the result holds trivially. Assume that $v > 0$. Then $d_1 > 0$ is a feasible direction at $\hat{x}$, which means that $x_k + \Delta_k(d_1, 0)$ is feasible for all sufficiently large $k \in K$. Then by the positive homogeneity of f° (Clarke 1983), it follows that

$$\begin{aligned} f^\circ(\hat{x}; (v, 0)) &= f^\circ\left(\hat{x}; \frac{v}{d_1}(d_1, 0)\right) \\ &= \frac{v}{d_1} f^\circ(\hat{x}; (d_1, 0)) \\ &= \frac{v}{d_1} \limsup_{y \to \hat{x},\ t \downarrow 0} \frac{f(y + t(d_1, 0)) - f(y)}{t} \\ &\geq \frac{v}{d_1} \lim_{k \in K} \frac{f(x_k + \Delta_k(d_1, 0)) - f(x_k)}{\Delta_k} \geq 0. \end{aligned}$$

The last step holds because $k \in K$ means that each term in the numerator of the final expression must be nonnegative. A similar argument can be made for $v < 0$ by using $-d_2$ instead of d_1. □

Theorem 4 *Suppose that, for $\hat{y} \in \mathcal{N}(\hat{x})$, $f(\hat{y}) < f(\hat{x}) + \xi$, where $\xi > 0$ is a lower bound on the extended poll trigger ξ_k for all k. If f is Lipschitz continuous near $\hat{z} \in X$ with respect to the continuous variables, then $f^\circ(\hat{z}; (v, 0)) \geq 0$ for all feasible directions $v \in \mathbb{R}$.*

Proof The proof is identical to that of Theorem 3, but with x_k and $\hat{x}$ replaced with z_k and $\hat{z}$, respectively. □

The classical second-order necessary condition for optimality requires that the Hessian be positive semidefinite on the null space of the binding constraints. However, in one dimension, if *any* constraint is binding, then the null space of the binding constraints is the zero subspace, in which case, the second-order necessary condition holds automatically because the zero vector is specifically excluded from the definition of positive definiteness. The next two results establish that $\hat{x}$ and $\hat{z}$ satisfy a second-order nonsmooth necessary condition for optimality with respect to the continuous variables when the limit point does not lie on the boundary of a linear constraint. These results act as corollaries to Theorem 5.15 in Abramson (2005), from which we denote by $f^{\circ\circ}(x; (d_1, 0), (d_2, 0))$ the Clarke (second-order) generalized directional derivative in the direction d_2 of the directional derivative $f'(x; (d_1, 0))$ of f at x in the fixed direction d_1 with respect to the continuous variables. That is, if $h(x) = f'(x; (d_1, 0))$, then $f^{\circ\circ}(x; (d_1, 0), (d_2, 0)) = h^{\circ}(x; (d_2, 0))$.

Theorem 5 *If f is continuously differentiable with Lipschitz continuous derivatives near $\hat{x}$ with respect to the continuous variables and either $\hat{x}^c \in \text{int}(X^c)$ or $X^c = \mathbb{R}$, then $f^{\circ\circ}(\hat{x}; (v, 0), (v, 0)) \geq 0$ for all $v \in \mathbb{R}$.*

Proof First, for $\lambda \geq 0$, the positive homogeneity of $f'(x; \cdot)$ yields

$$\begin{aligned}
&f^{\circ\circ}(\hat{x}; \lambda(w_1, 0), \lambda(w_2, 0)) \\
&\quad = \limsup_{y \to \hat{x},\ t \downarrow 0} \frac{f'(y + \lambda t(w_2, 0); \lambda(w_1, 0)) - f'(y; \lambda(w_1, 0))}{t} \\
&\quad = \lambda \limsup_{y \to \hat{x},\ t \downarrow 0} \frac{f'(y + \lambda t(w_2, 0); (w_1, 0)) - f'(y; (w_1, 0))}{t} \\
&\quad = \lambda \limsup_{y \to \hat{x},\ t \downarrow 0} \frac{\lambda[f'(y + (\lambda t)(w_2, 0); (w_1, 0)) - f'(y; (w_1, 0))]}{\lambda t} \\
&\quad = \lambda^2 \limsup_{y \to \hat{x},\ t \downarrow 0} \frac{f'(y + (\lambda t)(w_2, 0); (w_1, 0)) - f'(y; (w_1, 0))}{\lambda t}.
\end{aligned}$$

It follows that

$$f^{\circ\circ}(\hat{x}; \lambda(w_1, 0), \lambda(w_2, 0)) = \lambda^2 f^{\circ\circ}(\hat{x}; (w_1, 0), (w_2, 0)). \tag{12}$$

Let $D \supseteq \{d_1, -d_2 : d_1, d_2 > 0\}$ and $\beta = \frac{d_1}{d_2} > 0$. Under the assumptions, all directions $v \in \mathbb{R}$ are feasible. Since $f^{\circ\circ}(\hat{x}; (-v, 0), (-v, 0)) = f^{\circ\circ}(\hat{x}; (v, 0), (v, 0))$, we only need test the case where $v > 0$, in which case, $v = \alpha d_1$ for some $\alpha > 0$. Then it follows from (12) and the definition of $f^{\circ\circ}$ that

$$\begin{aligned}
&f^{\circ\circ}(\hat{x}; (v, 0), (v, 0)) \\
&\quad = f^{\circ\circ}(\hat{x}; \alpha(d_1, 0), \alpha(d_1, 0)) \\
&\quad = \alpha^2 f^{\circ\circ}(\hat{x}; (d_1, 0), (d_1, 0))
\end{aligned}$$

$$
\begin{aligned}
&= \alpha^2 \limsup_{y \to \hat{x},\ t \downarrow 0} \frac{f'(y + t(d_1, 0); (d_1, 0)) - f'(y; (d_1, 0))}{t} \\
&= \alpha^2 \limsup_{y \to \hat{x},\ t \downarrow 0} \left[\lim_{s \to 0} \frac{f(y + t(d_1, 0)) - f(y + t(d_1, 0) - s(d_1, 0))}{st} \right. \\
&\quad \left. - \lim_{s \to 0} \frac{f(y) - f(y - s(d_1, 0))}{st} \right] \\
&= \alpha^2 \left[\limsup_{y \to \hat{x},\ t \downarrow 0} \left[\lim_{s \to 0} \frac{f(y + t(d_1, 0)) - f(y + (t - s)(d_1, 0))}{ts} \right] \right. \\
&\quad \left. + \beta \limsup_{y \to \hat{x},\ t \downarrow 0} \left[\lim_{s \to 0} \frac{f(y - (s\beta)(d_2, 0)) - f(y)}{t(\beta s)} \right] \right] \\
&= \alpha^2 \left[\limsup_{y \to \hat{x},\ t \downarrow 0} \frac{f(y + t(d_1, 0)) - f(y)}{t^2} \right. \\
&\quad \left. + \beta \limsup_{y \to \hat{x},\ t \downarrow 0} \frac{f(y + t(-d_2, 0)) - f(y)}{t^2} \right] \\
&\geq \alpha^2 \left[\lim_{k \in K} \frac{f(x_k + \Delta_k(d_1, 0)) - f(x_k)}{\Delta_k^2} + \beta \lim_{k \in K} \frac{f(x_k + \Delta_k(-d_2, 0)) - f(x_k)}{\Delta_k^2} \right] \\
&\geq 0,
\end{aligned}
$$

since, for all sufficiently large $k \in K$, the refining directions d_1 and $-d_2$ satisfy $f(x_k) \leq f(x_k + \Delta_k(d_1, 0))$ and $f(x_k) \leq f(x_k + \Delta_k(-d_2, 0))$, respectively. □

Theorem 6 *Suppose that, for $\hat{y} \in \mathcal{N}(\hat{x})$, $f(\hat{y}) < f(\hat{x}) + \xi$, where $\xi > 0$ is a lower bound on the extended poll trigger ξ_k for all k. If f is continuously differentiable with Lipschitz continuous derivatives near $\hat{z}$ with respect to the continuous variables and either $\hat{z}^c \in \text{int}(X^c)$ or $X^c = \mathbb{R}$, then $f^{\circ\circ}(\hat{z}; (v, 0), (v, 0)) \geq 0$ for all $v \in \mathbb{R}$.*

Proof The proof is identical to that of Theorem 5, but with x_k and $\hat{x}$ replaced with z_k and $\hat{z}$, respectively. □

Finally, we conclude with the following theorem, which establishes reasonable conditions under which convergence of the algorithm to a local minimizer is ensured. In this theorem, the notation f_c'' denotes the second derivative of f with respect to the continuous variables (while holding discrete variables fixed).

Theorem 7 *Suppose that f_c'' is continuous near, and nonzero at, $\hat{x}$ and all $\hat{z}$ and $\hat{y} \in \mathcal{N}(\hat{x})$ satisfying $f(\hat{z}) = f(\hat{x})$ and $f(\hat{y}) = f(\hat{x})$, respectively. If the number of* EXTENDED POLL *steps is uniformly bounded, then $\hat{x}$ is a local minimizer of f on X.*

Proof We need to show that $\hat{x}$ is a local minimizer according to Definition 1. First, Theorem 3 and the smoothness of f ensure that $f'(\hat{x}; (v, 0)) = f^{\circ}(\hat{x}; (v, 0)) \geq 0$

for all feasible $v \in \mathbb{R}$. Furthermore, Theorem 5 and the assumption that $f_c''(\hat{x}) \neq 0$ ensure that $f_c''(\hat{x}) > 0$. These two results mean that $\hat{x}$ satisfies the classical sufficient conditions for local optimality with respect to the continuous variables.

Theorems 4 and 6 establish the same result for any $\hat{z}$ corresponding to a discrete neighbor $\hat{y}$ for which $f(\hat{y}) \leq f(\hat{x}) + \xi$, where $\xi > 0$ is a lower bound on the EXTENDED POLL triggers. Thus any such $\hat{z}$ is also a local minimizer of f with respect to the continuous variables.

Theorem 2 ensures that $f(\hat{x}) \leq f(\hat{y})$. If $f(\hat{x}) < f(\hat{y})$, as is the case when $f(\hat{y}) > f(\hat{x}) + \xi$, then there exists $\varepsilon > 0$ such that $f(\hat{x}) < f(w)$ for all $w \in X$ satisfying $w^d = \hat{y}^d$ and $w^c \in B(\hat{y}^c, \varepsilon)$.

Thus the only case left to show is when $f(\hat{x}) = f(\hat{y})$. In this case, the EXTENDED POLL step yields $y_k^{j+1} = y_k^j + \Delta_k(d^j, 0)$ for some $d^j \in D(y_k)$, and it follows that for all $k \in K$, $z_k^c = y_k^c + \Delta_k \sum_{j=1}^{J_k} d^j$. Since $\Delta_k \to 0$ (in K) and J_k is uniformly bounded (by assumption), it follows that $z_k - y_k \to 0$ (in K). Thus $\hat{y} = \hat{z}$, and we have that $\hat{y}$ is a local minimizer with respect to the continuous variables, and the entire result is proved. □

4 Numerical results

The MVPS algorithm is implemented in the NOMADm MATLAB® software (Abramson 2007) and was used to numerically solve several test cases. For each mass, the associated weight and damping constants were assumed to be 20 and 0.01 (unspecified) units, respectively, while spring constants for each mass are specified in Table 1.

Recall from (4) and the discussion that precedes it that each trial point is of the form $x = (p, M) \in X = X^c \times X^d$, where $p \in X^c = [0, 1]$ and $M \subset X^d = \{1, 2, \ldots, N\}$ contains the index numbers of the sensor locations. Thus no elements of M may repeat. This is equivalent to setting $X^d = \{[y_1, y_2, \ldots, y_n] : y_i \neq y_j \text{ for } i \neq j,\ i, j = 1, 2, \ldots, n\}$.

Given a current iterate $x_k = (p, M)$, the set of discrete neighbors $\mathcal{N}(x_k)$ consists of all points obtained by holding p constant while adding or subtracting 1 to any of

Table 1 Spring constant for each of 20 masses

Location	Constant	Location	Constant
1	2000	11	1350
2	1900	12	900
3	1800	13	800
4	2100	14	700
5	1600	15	1675
6	1500	16	500
7	1700	17	400
8	1950	18	1300
9	1200	19	200
10	1100	20	100

the elements of M (with the restriction that elements of M cannot be repeated); i.e., $\mathcal{N}(x) = \{(p, M \pm e_j) \in X : j = 1, 2, \ldots, n\}$, where $e_j \in \mathbb{R}^n$ is the standard unit vector in the jth coordinate direction, $j = 1, 2, \ldots, n$. Any local solution to the problem given in (4) would then be with respect to this choice of $\mathcal{N}(x)$. For example, the set $\mathcal{N}(x)$ of discrete neighbors of $x = (0.2, [10, 8, 2, 11])$ consists of the following points in X:

$$(0.2, [9, 8, 2, 11]), (0.2, [10, 7, 2, 11]), (0.2, [10, 9, 2, 11]),$$
$$(0.2, [10, 8, 1, 11]), (0.2, [10, 8, 3, 11]), (0.2, [10, 8, 2, 12]).$$

We divide our results into two main subsections, one for placing 4 sensors, and one for 3 and 5 sensors. A third subsection provides some general remarks on the optimization runs.

Each set of experiments was to optimally place the specified number of sensors, given that damage occurred at a specified mass j. In each case, the NOMADm software (Abramson 2007) was run 100 times, with randomly selected initial sensor locations and p initially set to 0.01. Constraints on the sensor locations (no more than one sensor per mass, and none at mass $j - 1$ or j) were enforced initially and in the construction of the set of discrete neighbors.

In presenting our results, we provide two tables for each scenario. The first summarizes the optimal values found for the 100 runs, while the second is a frequency table for the most commonly found sensor locations; that is, it lists the most commonly found sensor locations and, for each one listed, what percentage of the 100 runs found that particular location in the optimal solution.

4.1 Results for 4 sensors

We first considered the optimal placement of 4 sensors with damage occurring, in turn, at locations $j = 10$, 2, and 19. It may seem tempting to exhaustively enumerate the different combinations of sensor locations and solve a one-dimensional nonlinear optimization problem during each run. However, this would actually require $(20)(19)(18)(17)/4! = 4845$ runs, which is more than 48 times the number of runs used for these experiments.

Damage at mass 10

The results are summarized below. Table 2 contains information on optimal values found, while Table 3 is a frequency table for the 5 most commonly found sensor locations.

Table 2 Numerical results for 4 sensors, damage at mass 10

	Final $f(p, M)$	Final p
Mean	0.1218	0.3951
Mode	0.0767	0.31
Minimum	0.0206	0.0002344
St Dev	0.1123	0.2645

Table 3 Frequency results for 4 sensors, damage at mass 10

Location	% Occurrence
19	78
20	50
18	50
1	46
8	43

Table 4 Numerical results for 4 sensors, damage at mass 2

	Final $f(p, M)$	Final p
Mean	0.374	0.0125
Mode	0.228	0.0002344
Minimum	0.0636	0.0001123
St Dev	0.1967	0.08277

Table 5 Frequency results for 4 sensors, damage at mass 2

Location	% Occurrence
18	98
19	73
4	59
17	48

In 10% of the runs, the optimal solution included the first 4 masses (19, 20, 18, 1) from Table 3, and in 15% of the runs the 4 optimal masses were included in those found in Table 3. The optimal solution over all the runs consisted of placing sensors at masses [1, 2, 4, 8] and $p = 0.6352$, which yielded an objective function value of 0.0206.

This optimal objective function value is very low, meaning that the four sensors are able to capture a lot of information on the changes occurring in the system. However, the optimal solution was achieved when the spring stiffness has changed by 63%. Hence, the sensor configuration of [1, 2, 4, 8] is not desirable from a practical perspective. Moreover, the high median and mode values of p and relatively high standard deviation of p indicate that finding a good placement of 4 sensors that detects system damage at early stages is difficult.

On the other hand, the most common solutions found during the 100 runs (i.e., those which locate the sensors at masses [20, 19, 18, 1]), yield a much lower value of p, most being close to the minimum value found of $p = 0.0002$ (see Table 2), even though these solutions are sub-optimal with respect to the objective function value.

Damage at mass 2

Next, we investigated the outcome of sensor placement if the damage occurs near one of the endpoints, instead of mass 10. Tables 4 and 5 summarize the results for 100 runs with damage at mass 2. In addition to the locations specified in Table 5,

Table 6 Numerical results for 4 sensors, damage at mass 19

	Final $f(p, M)$	Final p
Mean	0.1296	0.3458
Mode	0.0048	0.358
Minimum	0.0043	0.01
St Dev	0.2018	0.2079

Table 7 Frequency results for 4 sensors, damage at mass 19

Location	% Occurrence
9	63
14	55
3	38
10	37

sensors were optimally placed at other masses 22% of the time or less. Placement at all four masses (4, 17, 18, 19) together occurred in 6% of the runs, while placements that included at least 3 of 4 of these locations accounted for 52% of the runs. The best solution found across all runs consisted of sensors placed at masses [17, 18, 19, 20] with $p = 0.0001123$, yielding a minimum objective function value of 0.0636. The small mean and mode values for p indicate that capturing system damage at an early stage is much easier than for the previous scenario.

This sensor placement configuration of [17, 18, 19, 20] is excellent because the objective function value of 0.0636 is optimal and results in a very low value of $p = 0.000123$. This means that it can capture change in spring stiffness very quickly (after just a 0.01% change) and accurately (6.36% difference between results for 4 sensors and 20 sensors).

Damage at mass 19

Finally, if the damage occurs at mass 19, similar results are obtained, as shown in Tables 6 and 7. In this case, placement of sensors at all four of the masses [3, 9, 10, 14] simultaneously did not occur, while placement of sensors on at least three of any four simultaneously occurred in 30% of the trials. Despite the low objective function value of 0.0043, the result is not good enough in practice because $p = 0.358$ is quite large. The relatively high mean, mode, and standard deviation of p mean that it is difficult to find a suitable sensor placement that can capture damage at mass 19 quickly. Our recommendation in this case is either to increase the number of sensors, or to look at suboptimal configurations with a relatively low value of p.

4.2 Results for 3 and 5 sensors

We now present similar results for the 3 and 5 sensor cases, in which damage occurs at mass 10. For the 3-sensor case, Tables 8 and 9 summarize the results under the same experimental conditions as in Sect. 4.1. Optimal placements at masses

Table 8 Numerical results for 3 sensors, damage at mass 10

	Final $f(p, M)$	Final p
Mean	0.3816	0.2911
Mode	0.7869	0.3100
Minimum	0.0217	0.0001123
St Dev	0.3174	0.2426

Table 9 Frequency results for 3 sensors, damage at mass 10

Location	% Occurrence
19	61
20	47
18	36
1	35
8	33

Table 10 Numerical results for 5 sensors, damage at mass 10

	Final $f(p, M)$	Final p
Mean	0.09001	0.4366
Mode	0.0206	0.31
Minimum	0.0201	0.0001123
St Dev	0.06458	0.2498

[18, 19, 20] occurred 15% of the time, while any 3 of the 5 masses were attained 35% of the time. The objective function value of 0.0217 was achieved by three different configurations: sensors [1, 8, 19] with $p = 0.6525$, sensors [7, 19, 20] with $p = 0.00023438$, and sensors [1, 4, 8] with $p = 0.6346$. The most common configuration of [18, 19, 20] yields an objective function value of 0.1731 with $p = 0.000123$. In this case, p is very close to optimal, but f is not (though it is lower than the mean value over all the runs). Furthermore, the high standard deviation (0.31) indicates that the probability of getting a value of f close to its minimum value is relatively small. The low objective function value also suggests that using only three sensors can yield enough information about system damage, but the mean, mode, and standard deviation for the 4-sensor configuration are all smaller.

For the 5-sensor case, Tables 10 and 11 summarize the results. Over 40% of the trials yielded optimal configurations with all sensor locations coming from the list in Table 11. The objective function value of 0.0201 was achieved by three different configurations: sensors [20, 4, 1, 8, 2] with $p = 0.6354$, sensors [2, 8, 14, 4, 1] with $p = 0.6355$, and sensors [4, 17, 1, 2, 8] with $p = 0.6355$.

Similar to the other scenarios, various optimal configurations can be found with low objective function values, but each with a different p-value. The more desirable configurations in both the 3-sensor and 5-sensor case are again those obtained most frequently, each of which does not have quite as good an objective function value, but has a much lower p-value. Not surprisingly, the 5-sensor configuration provides

Table 11 Frequency results for 5 sensors, damage at mass 10

Location	% Occurrence
20	77
17	66
1	55
18	53
8	50
19	40

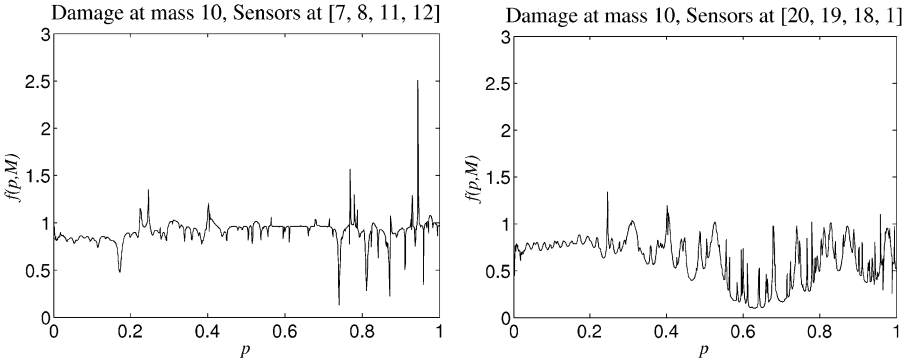

Fig. 3 Objective function profile for damage at mass 10

a slightly better optimal objective function value than 4-sensors can, and it is able to detect damage quickly ($p = 0.01\%$).

4.3 Additional remarks

Since the strong results obtained in Sect. 3.3 indicate that convergence to a local minimizer is almost guaranteed, these results show that the optimal sensor location problem that we have studied has many local minima. Indeed, this is verified by Fig. 3, in which plots of the objective function are given for 2 different fixed sets of four sensor locations when damage occurs at mass 10. The first is with sensor locations as close as possible to the damage, and the second with the optimal locations found by the NOMADm algorithm.

Even when based on simple transfer function estimates, the objective function not only has many local solutions, but is also highly nonlinear. Further complexity is expected for structural (geometric and material) nonlinearities. The results presented here were based on the linear system (1) and demonstrate the complex interrelationship between the number of sensors, their locations, and the ability to detect damage at an early stage (as defined by the percentage change p in stiffness).

In most cases, the most frequently found locally optimal solution is more desirable than the global one because it generally has a much lower p-value, meaning that damage is detected sooner. Indeed, this has been the challenge of SHM techniques so far, as most sensor configurations are not able to detect the onset of damage. The

most frequently found solution is also the most robust, in that it is less sensitive to different initial points.

With regard to the actual solutions, ideal sensor placement is not proximal to the site of damage, but rather, it is a nonlinear function of modal parameters, which often places sensors near masses adjacent and attached to springs with the least stiffness, provided that the damage is somewhat removed from that location. Because of this, we were able to identify some good sensor configurations that are not intuitively obvious, thus providing deeper insight into the problem.

5 Concluding remarks

We have introduced a new approach to solving sensor placement problems in the area of structural health monitoring, in which the problem is formulated as a mixed variable optimization problem and numerically solved by applying the mixed variable generalized pattern search algorithm. Furthermore, we have introduced new convergence results for pattern search algorithms applied to mixed variable problems having only one continuous variable, but any finite number of discrete variables. These results are stronger than previous pattern search results, but only valid for problems with one continuous variable. The approach introduced here is sufficiently general, so as to be applicable to other classes of sensor placement problems.

Our approach also allows us to evaluate options and tradeoffs in SHM. It would work best if potential sites for damage onset, such as locations of stress concentrations, are known with some degree of confidence. If we know *a priori* how many sensors are available, our optimization runs can be performed a statistically significant number of times (100 runs in this paper, using different random starting points), and the most frequent sensor locations found can be used as the recommended solution, provided that the corresponding percentage damage parameter p lies below a specified desired level, which is set by the user, based on engineering judgment.

Since we prefer solutions with low values of both $f(p, M)$ and p, the most logical next step in our study of sensor placement problems is to extend our formulation to multi-objective optimization, in which we attempt to simultaneously minimize both of these as objectives. In fact, we would eventually like to minimize a third function—the number of sensors needed—and consider more realistic problems that have nonlinear interactions between the masses, and noise and uncertainty associated with both the structure and sensors. Finally, we would like to study what effects multiple damage sites and varying likelihood of sites for onset damage have on the optimal sensor configuration.

Acknowledgements This work was done while the first author was a graduate student at Miami University, under the direction of the second and third authors. The authors wish to thank anonymous reviewers, whose comments significantly improved the quality of the manuscript.

The views expressed in this document are those of the authors and do not reflect the official policy or position of the United States Air Force, Department of Defense, or United States Government.

References

Abramson MA (2002) Pattern search algorithms for mixed variable general constrained optimization problems. PhD thesis, Department of Computational and Applied Mathematics, Rice University

Abramson MA (2004) Mixed variable optimization of a load-bearing thermal insulation system using a filter pattern search algorithm. Optim Eng 5(2):157–177

Abramson MA (2005) Second-order behavior of pattern search. SIAM J Optim 16(2):315–330

Abramson MA (2007) NOMADm optimization software. http://www.afit.edu/en/ENC/Faculty/MAbramson/NOMADm.html

Abramson MA, Audet C, Dennis JE Jr (2004) Filter pattern search algorithms for mixed variable constrained optimization problems. Pac J Optim (in press). Also appears as Technical Report TR04-09, Department of Computational and Applied Mathematics, Rice University, Houston, Texas

Audet C, Dennis JE Jr (2000) Pattern search algorithms for mixed variable programming. SIAM J Optim 11(3):573–594

Audet C, Dennis JE Jr (2003) Analysis of generalized pattern searches. SIAM J Optim 13(3):889–903

Audet C, Dennis JE Jr (2004) A pattern search filter method for nonlinear programming without derivatives. SIAM J Optim 14(4):980–1010

Audet C, Dennis JE Jr (2006) Mesh adaptive direct search algorithms for constrained optimization. SIAM J Optim 17(2):188–217

Booker AJ, Dennis JE Jr, Frank PD, Serafini DB, Torczon V (1998) Optimization using surrogate objectives on a helicopter test example. In: Borggaard J, Burns J, Cliff E, Schreck S (eds) Optimal design and control. Birkhäuser, Cambridge, pp 49–58

Booker AJ, Dennis JE Jr, Frank PD, Serafini DB, Torczon V, Trosset MW (1999) A rigorous framework for optimization of expensive functions by surrogates. Struct Optim 17(1):1–13

Chaudhuri S, Ghosh RK (2000) On optimal sensor placement with hypercube cutting planes. Technical Report CS497, Indian Institute of Technology, Kanpur, India

Clarke FH (1983) Optimization and nonsmooth analysis. Wiley, New York. Reissued in 1990 by SIAM Publications, as vol. 5 in the series Classics in Applied Mathematics

Doebling SW, Farrar CR, Prime MB, Shevitz DW (1996) Damage identification and health monitoring of structural and mechanical systems from changes in their vibration characteristics: a literature review. Technical Report LA-13070-MS, Los Alamos National Laboratory, Los Alamos, New Mexico

Guo HY, Zhang L, Zhang LL, Zhou JX (2004) Optimal placement of sensors for structural health monitoring using improved genetic algorithms. Smart Mater Struct 13:528–534

Kincaid RK, Padula SL (2002) D-optimal designs for sensor/actuator locations. Comput Oper Res 29(6):701–713

Kokkolaras M, Audet C, Dennis JE Jr (2001) Mixed variable optimization of the number and composition of heat intercepts in a thermal insulation system. Optim Eng 2(1):5–29

Kolda TG, Lewis RM, Torczon V (2003) Optimization by direct search: new perspectives on some classical and modern methods. SIAM Rev 45(3):385–482

Lewis RM, Torczon V (1999) Pattern search algorithms for bound constrained minimization. SIAM J Optim 9(4):1082–1099

Lewis RM, Torczon V (2000) Pattern search methods for linearly constrained minimization. SIAM J Optim 10(3):917–941

Lewis RM, Torczon V (2002) A globally convergent augmented Lagrangian pattern search algorithm for optimization with general constraints and simple bounds. SIAM J Optim 12(4):1075–1089

Lopes V Jr, Steffen V Jr, Inman DJ (2004) Optimal placement of piezoelectric sensor/actuators for smart structures vibration control. In: Udwadia F, Weber H, Leitman G (eds) Dynamical systems and control (stability and control), vol 22 (Part II). Chapman & Hall/CRC, Boca Raton, pp 221–236

Lucidi S, Piccialli V (2004) A derivative-based algorithm for a particular class of mixed variable optimization problems. Optim Methods Softw 17(3,4):317–387

Lucidi S, Piccialli V, Sciandrone M (2005) An algorithm model for mixed variable programming. SIAM J Optim 15(4):1057–1084

Mavroidis C, Dubowsky S, Thomas K (1997) Optimal sensor placement in motion control of flexibly supported long reach manipulators. Trans ASME, J Dyn Syst Meas Cont 119(4):718–726

Padula SL, Kincaid RK (1999) Optimization strategies for sensor and actuator placement. Technical Report LA-13070-MS, NASA-Langley Research Center, Hampton, Virginia

Padula SL, Palumbo DL, Kincaid RK (1998) Optimal sensor/actuator locations for active structural acoustic control. AIAA-98-1865

Papadimitriou C, Katafygiotis L, Karamanos S (2000) Optimal sensor placement strategies for structural damage identification. In: Proc. 8th ASCE specialty conference on probabilistic mechanics and structural reliability

Sriver TA, Chrissis JW, Abramson MA (2005) Pattern search ranking and selection algorithms for mixed variable simulation-based optimization. Preprint

Staszewski WJ, Worden K, Wardle R, Tomlinson GR (2000) Fail-safe sensor distributions for impact detection in composite materials. Smart Mater Struct 9(3):298–303

Torczon V (1997) On the convergence of pattern search algorithms. SIAM J Optim 7(1):1–25