A New Generating Set Search Algorithm for Partially Separable Functions

Lennart Frimannslund
Department of Informatics
University of Bergen
Bergen, Norway
Email: lennart@ii.uib.no

Trond Steihaug
Department of Informatics
University of Bergen
Bergen, Norway
Email: trond@ii.uib.no

***Abstract*—A new derivative-free optimization method for unconstrained optimization of partially separable functions is presented. Using average curvature information computed from sampled function values the method generates an average Hessian-like matrix and uses its eigenvectors as new search directions. For partially separable functions, many of the entries of this matrix will be identically zero. The method is able to exploit this property and as a consequence update its search directions more often than if sparsity is not taken into account. Numerical results show that this is a more effective method for functions with a topography which requires frequent updating of search directions for rapid convergence.**

The method is an important extension of a method for non-separable functions previously published by the authors. This new method allows for problems of larger dimension to be solved, and will in most cases be more efficient.

***Keywords*-Generating Set Search, Derivative-Free Optimization, Partial Separability, Sparsity.**

I. Introduction

Continuous optimization is an important area of study, with applications in statistical parameter estimation, economics, medicine, industry — simply put, anywhere a mathematical model can be used to represent some real-world process or system which is to be optimized. Mathematically, we can express such a problem as

$$\min_{x \in D \subseteq \mathbb{R}^n} f(x), \tag{1}$$

where f is the objective function, based on the model which is defined on the domain D. These models can range from simple analytic expressions to complex simulations. Well known optimization methods such as Newton's method use derivatives to iteratively find a solution. These derivatives must be provided, either through explicit formulas/computer code, or, for instance, automatic differentiation.

Suppose, however, that the objective function is produced by some sort of non-differentiable simulation, or that it involves expressions which can only be computed numerically, such as the solution to differential equations, integrals, and so on. In this case derivatives might not exist, or they may be unavailable if the numerically computed function is subject to some kind of adaptive discretization and truncation and therefore is non-differentiable, unlike the underlying mathematical function. In these cases derivative-based methods are not directly applicable, which leads to the need for methods that do not explicitly require derivatives. For an introduction to derivative free methods the reader is referred to [1].

Generating set search (GSS) methods are a subclass of derivative-free methods for unconstrained optimization. These methods can be extended to handle constraints, but we will focus on the unconstrained case when the domain D in the problem (1) is equal to $\mathbb{R}^n$. A comprehensive introduction to these methods can be found in [12]. In their most basic form these methods only use function values and do not collect any information such as average slope or average curvature information. Computing this information, however, can significantly speed up convergence, and this is done in the methods presented in [2], [3], [4].

In addition, information about the structure of the function known a priori can also be useful. Suppose that the objective function f can be written as a sum of element functions,

$$f = \sum_{i=1}^{m} f_i,$$

where each element function has the property that it is unaffected when we move along one or more of the coordinate directions. For example, we might have

$$f(x_1, x_2, x_3) = f_1(x_1, x_2) + f_2(x_2, x_3). \tag{2}$$

Then, the function is said to be partially separable [9] and we say that f_i has a large null space. If f is partially separable and twice continuously differentiable, then its Hessian matrix,

$$\nabla^2 f(x) = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \vdots & & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{bmatrix},$$

will be sparse. For the function (2) the Hessian element $\frac{\partial^2 f}{\partial x_1 \partial x_3}$ will be zero. If the function (2) is not twice continuously differentiable, then the matrix of the corresponding finite differences, that is, the matrix with

$$\Big[f(x_1+h,x_2,x_3+k)-f(x_1+h,x_2,x_3)$$

$$-f(x_1,x_2,x_3+k)+f(x_1,x_2,x_3)\Big]\Big/(hk)=0, \quad (3)$$

in position $(i,j)=(1,3)$ (and with similar expressions for all other (i,j)-pairs) will be sparse for any x, and any nonzero h and k, none of which have to be the same for each (i,j)-pair. The sparsity structure is the same as for the differentiable case, so that the expression (3) is identically zero. This result can be extended to any partially separable function, as proved in [5].

In [15] a GSS method which exploits such structure is presented, which is applicable to the case where these element functions are individually available.

In this paper we present a GSS method which takes advantage of the structure of partially separable functions, without requiring the element functions (which may or may not be differentiable) to be available. It is an extension of the paper [4]. We use the concept of average curvature introduced in [4].

This paper is organized as follows: In section II we outline a basic framework for GSS, as well as the previous work of the authors on which the present paper is based. In Sections III and IV we present our main contribution, which is the framework for handling partially separable functions. Section V contains numerical results, and concluding remarks are given in Section VI.

II. Generating set search using curvature information

We restrict ourselves to a subset of GSS methods, namely sufficient decrease methods with $2n$ search directions, the positive and negative of n mutually orthogonal directions, of unit length. These directions will in general *not* be the coordinate directions. A simplified framework for the methods we consider is given in Figure 1. The univariate function ρ must be nondecreasing and satisfy $\lim_{x\downarrow 0}\frac{\rho(x)}{x}=0$. For simplicity, increasing the step length can be thought of as multiplying it by 2, and decreasing it as dividing by 2, although these rules may be more advanced. For the formal requirements on these rules, see [12]. Given mild requirements on the function f the step length δ will ultimately go to zero, and the common convergence criterion for all GSS methods is that δ is smaller than some tolerance.

As can be seen from the pseudo code in Figure 1, the set of search directions can be periodically updated. In [4], the authors present a method that computes average curvature information from previously sampled points, assembles this information in a Hessian-like matrix and uses the eigenvectors of this matrix as the search directions, which amounts to a rotation of the old search directions. Once this rotation has been performed, the process restarts, and new curvature information is computed, periodically resulting in new search directions. It is shown that the efficiency of the method can be greatly improved compared to just using the coordinate directions as the search directions throughout.

```
Given set of search directions $\mathcal{Q}$, step length $\delta$ and an
initial guess $x \leftarrow x_0$.
While $\delta$ is larger than some tolerance
    Repeat until $x$ has been updated or all $q \in \mathcal{Q}$ have
    been used:
        Get next search direction $q \in \mathcal{Q}$.
        If $f(x+\delta q) < f(x) - \rho(\delta)$
            Update $x$: $x \leftarrow x + \delta q$.
            Optionally increase $\delta$.
        End if
    End repeat
    If no search direction provided a better function
    value, decrease $\delta$.
    Optionally update $\mathcal{Q}$.
End while
```

Figure 1. Simplified framework for a sufficient decrease GSS method.

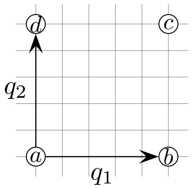

Figure 2. Location of sampled points used for curvature computation.

The computation of curvature information can be done in the following way, which is a slight modification of the methodology presented in [4]. Consider Figure 2, and assume that the current point is the point marked a, and that the next two search directions in the repeat-loop in the pseudo code are the directions shown, q_1 and q_2. When searching along two directions in a row, there are four possible outcomes. Success-success (both the search along q_1 and q_2 produce function values which satisfy the sufficient decrease condition), success-failure (the search along q_1 produces a sufficiently lower function value, but the search along q_2 does not), failure-success, and finally failure-failure. In all of these four cases, by computing the function value at a fourth point, the function values at four points in a rectangle can be obtained. The details are given in Table I. The function values at four such points a, b, c and d can be inserted into the formula

$$\frac{f(c)-f(b)-f(d)+f(a)}{\|b-a\|\ \|d-a\|}. \quad (4)$$

If the objective function is twice continuously differentiable, then (4) is equal to $q_1^T\nabla^2 f(\hat{x})q_2$, where $\hat{x}$ is some point within the rectangle $abcd$. If the function is not twice continuously differentiable, (4) captures the average curvature

Outcome	Notes
SS	The search along q_1 moves the current best point to b, and the search along q_2 moves the current best point to c. The function value at d must be computed separately.
SF	The search along q_1 moves the current best point to b, and the search along q_2 computes the function value at c, but does not move the current best point. The function value at d must be computed separately.
FS	The search along q_1 computes the function value at point b, but does not move the current best point. The search along q_2 computes the function value at point d. The function value at point c must be computed separately.
FF	Neither the search along q_1 nor q_2 update the current best point, but the function values at points b and d are obtained. The function value at point c must be computed separately.

Table I
THE FOUR POSSIBLE OUTCOMES WHEN SEARCHING ALONG TWO CONSECUTIVE DIRECTIONS. S MEANS SUCCESS, F MEANS FAILURE.

in the rectangle.

The rectangle lies in the plane spanned by the search directions q_1 and q_2 since these were used consecutively. By successively reordering how the "get next search direction" statement considers the directions in $\mathcal{Q}$, one can obtain curvature information with respect to all the $n(n-1)/2$ possible different combinations of search directions, in a finite and uniformly bounded number of steps, which depends on n since there are $O(n^2)$ elements of curvature information which must be assembled. (For this reason, the method is not suitable for n larger than about 30, but exploiting structure can allow for much larger n, as will be explained in Section III.)

The information can be assembled in a matrix C_Q, so that C_Q, in the case of a twice continuously differentiable f, contains $q_i^T \nabla^2 f(\hat{x}) q_j$ in positions (i, j) and (j, i), which is curvature information with respect to the coordinate system defined by the n directions in $\mathcal{Q}$. (Note that the point $\hat{x}$ is different for each (i, j)-pair.) The diagonal elements of C_Q must be computed separately, for instance when the step length is reduced, since the preceding repeat-loop, combined with the current f-value then gives the function values at three equally spaced points on a straight line for all n search directions.

Once the matrix C_Q is complete, it is subjected to the rotation

$$C \leftarrow Q C_Q Q^T, \tag{5}$$

where Q is the matrix with the n unique search directions as its columns, ordered so that they correspond to the ordering of the elements in C_Q. C now contains curvature information with respect to the standard coordinate system. The search directions in $\mathcal{Q}$ are then replaced with the positive and negative of the eigenvectors of C.

III. Extension to separable functions

Suppose the function f is partially separable. As mentioned in the introduction, the Hessian will be sparse if f is twice continuously differentiable, and if the Hessian is not defined, the matrix of average curvature information will be sparse [5]. Let r be the number of nonzero elements in the lower diagonal of these curvature matrices. Then, even though the matrix C can be restricted to have this sparsity pattern, the matrix C_Q cannot be assumed to be sparse, since we cannot expect the finite differences (4) to be zero for arbitrary search directions $\mathcal{Q}$. However, sparsity can still be exploited.

Define the Kronecker product. Given two matrices $A \in \mathbb{R}^{m \times n}$ and B, then the Kronecker product $A \otimes B$ is given as

$$A \otimes B = \begin{bmatrix} A_{11}B & \cdots & A_{1n}B \\ \vdots & & \vdots \\ A_{m1}B & \cdots & A_{mn}B \end{bmatrix}. \tag{6}$$

The Kronecker product is useful in the present context because of the relation

$$AXB = C \Leftrightarrow (B^T \otimes A)\mathbf{vec}(X) = \mathbf{vec}(C). \tag{7}$$

Here $\mathbf{vec}(X)$ and $\mathbf{vec}(C)$ are vectors containing the entries of the matrices X and C stacked row-wise [11].

Using (6) and (7) the rotation (5) can be written implicitly as

$$(Q^T \otimes Q^T)\mathbf{vec}(C) = \mathbf{vec}(C_Q). \tag{8}$$

Since we impose a sparsity structure on C as well as symmetry, all the entries in the upper triangle, as well as all the zero entries of $\mathbf{vec}(C)$ can be removed from (8), resulting in the overdetermined equation system

$$(Q^T \otimes Q^T)P_c\overline{\mathbf{vec}}(C) = \mathbf{vec}(C_Q), \tag{9}$$

where the vector $\overline{\mathbf{vec}}(C)$ contains the r elements of C to be determined, and the $n^2 \times r$ 0-1 matrix P_c adds together the columns corresponding to upper and lower diagonal elements C_{ij} and C_{ji} for all off-diagonal elements, and deletes the columns corresponding to zero entries in C. For example, if C is to be tridiagonal and is of size 3×3, that is,

$$C = \begin{bmatrix} \times & \times & \\ \times & \times & \times \\ & \times & \times \end{bmatrix},$$

then it has one zero element and five nonzero elements in the lower triangle, so that P_c has size 9×5 and reads:

$$P_c = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \tag{10}$$

Since the equation system (9) is overdetermined, we can select r rows from the coefficient matrix and the right-hand side, resulting in the $r \times r$ equation system

$$P_{\text{row}}(Q^T \otimes Q^T)P_c\overline{\mathbf{vec}}(C) = P_{\text{row}}\mathbf{vec}(C_Q), \tag{11}$$

where P_{row} is an $r \times n^2$ 0-1 matrix which selects r rows. P_{row} will be the first r rows of a permuted $n^2 \times n^2$ identity matrix. The resulting equation system (11) will be significantly smaller than its counterpart (8) when a sparsity structure is imposed on C, and the corresponding effort required to compute the right-hand side is similarly smaller. If there are only $O(n)$ elements to be determined, then the number of steps needed to compute the entire right-hand side $P_{\text{row}}\mathbf{vec}(C_Q)$ does not depend on n, which does away with the practical limit on dimension discussed in the previous section.

Exactly which rows P_{row} should select in order to create a well-conditioned coefficient matrix is nontrivial, and is sometimes called the subset selection problem in the literature (see e.g., [7]). One suitable solution procedure is to determine these rows by computing a strong rank-revealing QR factorization of the transpose of $P_{\text{row}}(Q^T \otimes Q^T)$ and selecting the rows chosen by the theory and algorithms of Gu and Eisenstat, presented in [10]. An implementation of this selection procedure can be found in [14].

IV. Convergence theory

The method presented so far, being a sufficient decrease method with $2n$ search directions which are the positive and negative of n mutually orthogonal directions, adheres to the algorithmic framework and convergence theory of Lucidi and Sciandrone [13]. We can therefore state the following theorem, without proof.

Theorem 1: Suppose f is continuously differentiable, bounded below and the level set $\mathcal{L}(x) = \left\{ y \middle| f(y) \leq f(x) \right\}$ is compact. Then, the method converges to a first-order stationary point.

We now prove that if f is twice continuously differentiable, then the computed curvature matrix C converges to the true Hessian in the limit.

Define

$$A = P_{\text{row}}(Q^T \otimes Q^T)P_c.$$

Let f be twice continuously differentiable and the Hessian Lipschitz-continuous in the sense that

$$\|\nabla^2 f(x) - \nabla^2 f(y)\| \leq L\|x - y\|. \tag{12}$$

Define r pairs of vectors $p^{(k)}, q^{(k)}\ \ k = 1, \ldots, r$, all of unit length, such that the kth row of A is equal to

$$\left(p^{(k)T} \otimes q^{(k)T}\right)P_c. \tag{13}$$

This means some of these vectors will be equal, but the pairs will be unique. In addition let r points $x^k,\ k = 1, \ldots, r$, be such that element k of $P_{\text{row}}\mathbf{vec}(C_Q)$,

$$(P_{\text{row}}\mathbf{vec}(C_Q))_k = p^{(k)T}\nabla^2 f(x^k)q^{(k)}.$$

Let η be such that

$$\max_{i,j} \|x^i - x^j\| = \eta.$$

Let $\mathcal{N}$ be the neighborhood of points such that

$$\mathcal{N} = \left\{ x \middle| \|x - x^k\| \leq \eta,\ k = 1, \ldots, r \right\}.$$

For convenience, let us restate (11), as

$$A\overline{\mathbf{vec}}(C) = P_{\text{row}}\mathbf{vec}(C_Q). \tag{14}$$

Lemma 2: Assume A is invertible. Let C be the symmetric $n \times n$ matrix constructed from the solution of (14). Then, there exists an $x \in \mathcal{N}$ such that

$$\|\nabla^2 f(x) - C\| \leq \|A^{-1}\|nL\eta.$$

Proof. Let us rewrite the contents of $P_{\text{row}}\mathbf{vec}(C_Q)$:

$$\begin{aligned} & (P_{\text{row}}\mathbf{vec}(C_Q))_k \\ = \ & p^{(k)T}\nabla^2 f(x^k)q^{(k)}. \\ = \ & p^{(k)T}\left(\nabla^2 f(x) + \nabla^2 f(x^k) - \nabla^2 f(x)\right)q^{(k)} \\ = \ & \left[p^{(k)T}\nabla^2 f(x)q^{(k)}\right] + \\ & \left[p^{(k)T}(\nabla^2 f(x^k) - \nabla^2 f(x))q^{(k)}\right]. \end{aligned} \tag{15}$$

Then, and in addition defining $h = \overline{\mathbf{vec}}(\nabla^2 f(x))$, equation (14) can be written as

$$A\overline{\mathbf{vec}}(C) = Ah + \epsilon. \tag{16}$$

Here $(Ah)_k$ is the expression in the first parenthesis of (15), and ϵ_k is the expression in the last parenthesis of (15). If we consider the norm of a single element in ϵ, this is

$$\begin{aligned} |\epsilon_k| & \leq \|p^{(k)}\|\|\nabla^2 f(x^k) - \nabla^2 f(x)\|\|q^{(k)}\| \\ & \leq L\eta, \end{aligned} \tag{17}$$

using (12) and the fact that p and q have unit length. When solving (14), we get

$$\overline{\mathbf{vec}}(C) = h + A^{-1}\epsilon.$$

If we consider a single element of $\overline{\mathbf{vec}}(C)$ and h we can write

$$|(\overline{\mathbf{vec}}(C))_k - h_k| \leq \|A^{-1}\||\epsilon_k|,$$

which can also be written

$$|C_{ij} - (\nabla^2 f(x))_{ij}| \leq \|A^{-1}\||\epsilon_k| \tag{18}$$

Using the property of the 2-norm that

$$\|A\|_2 \leq n \max_{i,j} |a_{ij}|,$$

as well as (17) we can extend (18) to

$$\|C - \nabla^2 f(x)\| \leq \|A^{-1}\| nL\eta,$$

which completes the proof. □

We must now prove that there always exists a matrix A with rank r, and that the term $\|A^{-1}\|$ is uniformly bounded. Since A is made up of the rows of the matrix $(Q^T \otimes Q^T)P_c$, there will be a choice of rows which imply full rank if the matrix $(Q^T \otimes Q^T)P_c$ has rank r.

Lemma 3: For any orthogonal matrix Q and any sparsity structure to be imposed on C, the matrix $(Q^T \otimes Q^T)P_c$ has full rank r, and its smallest singular value σ_r satisfies $\sigma_r \geq 1$.

Proof. Since Q is orthogonal, so is Q^T, and also $(Q^T \otimes Q^T)$. For any sparsity structure, right-multiplying $(Q^T \otimes Q^T)$ with P_c either adds together two columns, or deletes columns. Consequently, the columns of the resulting matrix $(Q^T \otimes Q^T)P_c$ are orthogonal (which implies full rank), and have either length one or length $\sqrt{2}$. It then follows that the singular values are equal to the length of the column vectors, either 1 or $\sqrt{2}$. □

Lemma 4: P_{row} can be chosen such that for a given n, the smallest singular value of A is uniformly bounded below, and consequently that $\|A^{-1}\|$ is uniformly bounded.

Proof. This result follows from the theory and methods of Gu and Eisenstat [10], which guarantee that the rows of A (or equivalently the columns of A^T, as is done in [10]) can be selected from the rows of $(Q^T \otimes Q^T)P_c$ in such a way that the smallest singular value of A is larger than or equal to the smallest singular value of $(Q^T \otimes Q^T)P_c$, divided by a low order polynomial in n and r. Since n and r are given and the smallest singular value of $(Q^T \otimes Q^T)P_c$ is always larger than or equal to 1, the result follows. □

Finally, we show that η goes to zero as the GSS method converges to a stationary point.

Lemma 5: Assume that the step length expansion factor is uniformly bounded by, say, M. Then, as the step length δ go to zero, so does η.

Proof. That the step length δ goes to zero is an integral part of the convergence theory of GSS methods and is proved in e.g. [12]. η is the diameter of neighborhood of points $\mathcal{N}$. Since all the points in $\mathcal{N}$ lie within the rectangles of points used in the formula (4), it follows that η must be smaller than maximum possible distance between the first and the last corner point used for computing C. Suppose, that when the computation of C is started the step length is $\delta_{\max}$, and that the maximum possible number of step length increases before C is computed is t. Then we have

$$\eta \leq \sum_{k=0}^{t} \delta_{\max} M^{k-1}.$$

The only variable in this expression is $\delta_{\max}$, and we know it goes to zero as the method converges. Consequently, so must η. □

This allows us to state the following theorem:

Theorem 6: Assume that f is twice continuously differentiable, bounded below and that the level sets $\mathcal{L}(x)$ are compact. Then, as the method converges, C converges to the true Hessian.

The proof follows from the preceding Lemmas. This result, together with the preliminary numerical results in [6] allows us to conjecture that the method actually converges to second-order stationary points.

V. Numerical Results

For the sake of brevity, there are many common implementation details for GSS methods which have been omitted in this paper. For instance, it is possible to have individual step lengths (e.g., n step lengths, one for each positive-negative search direction pair), to compute an approximate gradient and performing Newton-like steps, have variations on how step length(s) can be increased and decreased, choose ρ in several ways, and so on. These all affect the numerical performance of the method. The purpose of the present paper is, however, to show the benefits of exploiting sparsity when computing curvature information in the context of GSS methods. For this reason, it is the relative increase in performance when exploiting sparsity that is important in our numerical experiments, which used, among other things, n individual step lengths. The results are shown in Table II. The table reads, from left to right, the function name, and the dimension n. The functions are all differentiable, so the column r indicates the number of nonzero elements in the Hessian matrix. Then follow the number of function evaluations required to reduce the objective function value from the recommended initial solution to 10^{-5}, first for the method exploiting sparsity, and finally for the method not exploiting sparsity. The functions all have an optimal objective function value $f^* = 0$.

As one can see, one sometimes can get significant savings when exploiting sparsity, for example for the extended Rosenbrock function, CRAGGLVY, MOREBV and TQUARTIC. The reason for this is that the new method is able to rotate its search directions more often, which adapts them to the local topography of the objective function.

If we look at the extended Rosenbrock function there are several advantages to exploiting sparsity. Firstly there are $3n/2$ nonzero elements in the Hessian, which means

Function	n	r	Sparsity	No sparsity
BRYBND	10	49	936	1100
	50	329	4111	3774
CHNROSNB	10	19	2103	2971
	25	49	7400	15451
	50	99	26385	52574
CRAGGLVY	4	5	118	481
DECONVU	61	767	3232	15790
DQRTIC	10	9	335	471
	50	49	2991	3774
Ext. Rosenbr.	16	24	3369	6407
	32	48	6945	16577
	64	96	13889	50635
FREUROTH	10	19	912	1226
LIARWHD	36	71	3602	5257
MOREBV	10	27	363	521
	50	147	1320	5769
SBRYBND	10	49	747	736
SPARSQUR	100	1232	2878	2988
TQUARTIC	50	99	9022	14176
TRIDIA	20	39	1065	1453
	30	59	1662	2791
	50	99	2843	5621

Table II
NUMBER OF FUNCTION FUNCTION EVALUATIONS REQUIRED TO REDUCE THE OBJECTIVE FUNCTION VALUE TO 10^{-5}, STARTING AT THE RECOMMENDED INITIAL SOLUTION, FOR SELECTED FUNCTIONS FROM THE CUTER TEST SET [8].

that in relative terms, C can be computed increasingly cheaply as n grows. Secondly, the Hessian is block diagonal, which implies that it has element functions which can be optimized independently. As a consequence the eigenvectors have a block structure as well, which, since there are n step lengths, actually means that the method exploiting sparsity automatically optimizes the element functions independently of each other. This is reflected in the fact that the number of function evaluations needed to obtain a solution grows more or less linearly with n, as opposed to when not exploiting sparsity, where the growth in function evaluations is almost quadratic.

If the topography is such that frequent updating of the search directions is not important, then the results are more similar for the two algorithms.

VI. CONCLUSION

We have presented a GSS algorithm which exploits the partial separability of the objective function. The method is provably convergent to first-order stationary points, and based on its theoretical and numerical properties we conjecture that it is convergent to second-order stationary points. Numerical results indicate that exploiting separability can lead to significant improvement in convergence, in many cases.

ACKNOWLEDGEMENTS

The authors would like to thank Marielba Rojas for suggesting the use of rank-revealing QR factorizations. The first author gratefully acknowledges partial funding from The Norwegian Research Council, Gassco and Statoil under contract 175967/S30.

REFERENCES

[1] A. R. Conn, K. Scheinberg, and L. N. Vicente. *Introduction to Derivative-Free Optimization*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2009.

[2] I. D. Coope and C. J. Price. A direct search conjugate directions algorithm for unconstrained minimization. *The ANZIAM Journal*, 42(E):C478–C498, 2000.

[3] A. L. Custódio and L. N. Vicente. Using sampling and simplex derivatives in pattern search methods. *SIAM Journal on Optimization*, 18(2):537–555, 2007.

[4] L. Frimannslund and T. Steihaug. A generating set search method using curvature information. *Computational Optimization and Applications*, 38(1):105–121, 2007.

[5] L. Frimannslund and T. Steihaug. Sparsity of the average curvature information matrix. *PAMM, Proc. Appl. Math. Mech.*, 7:1062101–1062102, 2007.

[6] L. Frimannslund and T. Steihaug. Convergence basins for some derivative-free optimization methods on problems with saddle points. Technical Report 394, Department of Informatics, University of Bergen, Norway, 2010.

[7] G. H. Golub and C. F. van Loan. *Matrix Computations*. The Johns Hopkins University Press, 3rd edition, 1996.

[8] N. I. M. Gould, D. Orban, and Ph. L. Toint. CUTEr (and SifDec), a constrained and unconstrained testing environment, revisited. Technical Report RAL–TR–2002–009, Computational Science and Engineering Department, Rutherford Appleton Laboratory, 2002.

[9] A. Griewank and Ph. L. Toint. On the unconstrained optimization of partially separable functions. In M. Powell, editor, *Nonlinear Optimization 1981*, pages 301–312. 1982.

[10] M. Gu and S. C. Eisenstat. Efficient algorithms for computing a strong rank-revealing QR factorization. *SIAM J. Sci. Comput.*, 17(4):848–869, 1996.

[11] R. A. Horn and C. R. Johnson. *Topics in matrix analysis*. Cambridge University Press, Cambridge, United Kingdom, 1991.

[12] T. G. Kolda, R. M. Lewis, and V. Torczon. Optimization by direct search: New perspectives on some classical and modern methods. *SIAM Review*, 45(3):385–482, 2003.

[13] S. Lucidi and M. Sciandrone. On the global convergence of derivative-free methods for unconstrained optimization. *SIAM Journal on Optimization*, 13(1):97–116, 2002.

[14] S. R. Pope. *Parameter Identification in Lumped Compartment Cardiorespiratory Models*. PhD thesis, North Carolina State University, Raleigh, North Carolina, USA, 2009.

[15] C. P. Price and Ph. L. Toint. Exploiting problem structure in pattern search methods for unconstrained optimization. *Optimization Methods and Software*, 21(3):479–491, 2006.