International Journal of Statistics and Applications 2017, 7(2): 73-92
DOI: 10.5923/j.statistics.20170702.03

# Competitive Assessment of Two Variance Weighted Gradient (VWG) Methods with Some Standard Gradient and Non-Gradient Optimization Methods

**Otaru O. A. P., Iwundu M. P.***

Department of Mathematics and Statistics, University of Port Harcourt, Port Harcourt, Nigeria

**Abstract** A competitive assessment of the performance of two Variance Weighted Gradient (VWG) methods with some gradient and non-gradient optimization methods is considered for optimizing polynomial response surfaces. The variance weighted gradient methods could involve several response surfaces defined on joint feasible regions having same constraints or several response surfaces defined on disjoint feasible regions having different constraints. The gradient and non-gradient optimization methods include Quasi-Newton (GN), Genetic Algorithm (GA), Mesh Adaptive Search (MADS) and Generalized Pattern Search (GPS) methods. The variance weighted gradient methods perform considerably and comparatively well and require few iterative steps to convergence.

**Keywords** Response Surfaces, Optimization, Gradient Method, Non-Gradient Method, Weighted Gradients methods

## 1. Introduction

Gradient and non-gradient methods are popular techniques used in optimizing response functions. They have been very helpful in handling optimization of single objective functions. In the presence of multi-objective functions, the popularly used one-at-a-time optimization techniques become time consuming. In fact, the challenge in getting a good guess of starting point introduces cycling and sometime lack of convergence. The Gradient and non-gradient methods include Newton Method (NM), Guasi-Newton Method (GNM), Genetic Algorithm (GA), Mesh Adaptive Search (MADS) and Generalized Pattern Search (GPS) methods, etc. These methods could readily be seen in statistical software and thus, easy to use in handling optimization problems. As documented in Patil and Verma (2009), the Newton method was developed by Newton in 1669 and was later improved by Raphson in 1690 and hence popularly referred to as Newton-Raphson's iterative algorithm. NM is the commonly applied gradient-based method for optimizing polynomial function when the first and second derivative of the function is explicitly available. It assumes that the function can be locally approximated as a quadratic in the region around the optimum. Newton method requires computation of Hessian matrix and the convergence of the method is slow.

Davidon (1959) developed the Quasi Newton Method (QNM) which was later popularized by Fletcher and Powell (1963). The Quasi Newton Method can be used if the Hessian matrix is unavailable or too expensive to compute at all iterations. Thus the Quasi Newton Method helps to reduce computational rigor involved in the Newton method. Commonly used Quasi Newton algorithms are the SR1 formular, the BHHH method, the BFGS method and the L-BFGS method. Holland (1975) developed Genetic Algorithms (GAs) for linear and non-linear functions optimization. The algorithm solves both constrained and unconstrained optimization problems using natural procedure that imitates biological evolution.

Genetic Algorithms can be used to solve problems where standard optimization techniques do not apply as well as problems in which the objective function is discontinuous or non-linear. Pattern search algorithms can also be employed in getting optimal solutions. The generalized pattern search (GPS) for unconstrained optimization problems is due to Torczon (1997) and does not require information about the gradient or higher derivative to arrive at the optimal point. Audet and Dennis Jr (2006) developed the Mesh Adaptive Direct Search (MADS) as a class of algorithm that extends the generalized pattern search. Both algorithms compute a sequence of points that approach the optimal solutions. Abramson *et.al* (2009) applied the Mesh adaptive direct search in solving constrained mixed variable optimization problems in which variables may be continuous or categorical. The gradient-free class of Mesh adaptive direct search algorithm is called Mixed Variable MADS

* Corresponding author:
mary.iwundu@uniport.edu.ng (Iwundu M. P.)
Published online at http://journal.sapub.org/statistics

abbreviated MV-MADS. Audit *et.al* (2010) introduced MULTIMAD, a multi-objective Mesh adaptive direct search optimization algorithm. For more on the Gradient and non-gradient methods, see Audet and Dennis Jr (2009), Audet *et.al* (2014), Dolan *et.al* (2003), Goldberg (1989), Kolda *et.al* (2003), Lewin (1994a), Lewin (1994b), Lewis and Torczon (2000).

Iwundu *et.al* (2014) developed an iterative variance weighted gradient procedure that can simultaneously optimize multi-objective functions defined on the same region or having the same constraints. When the constraints are different and the regions are disjoint, a projection scheme that allows the projection of design points from one region to another is used. The performance of the variance weighted gradient method due to Iwundu *et al*. (2014) and its modified projection scheme technique shall be compared with the gradient-based QNM and non-gradient optimization methods, namely, GA, MADS and GPS. The basic algorithmic steps of the variance weighted gradient methods are presented in section 2.

## 2. Methodology

We present the algorithmic steps of the variance weighted gradient method for joint feasible region as well as for disjoint feasible region.

### 2.1. Variance Weighted Gradient (VWG) Method for Joint Feasible Regions with Same Constraints

Let $f_k(x)$ be an n-variate, p-parameter polynomial functions of degree m defined by constraints on the same feasible region, given by

$$f_k(x) = \underline{a}' \underline{x} + e; k = (1, M) \tag{1}$$

$$\underline{x} \in \widetilde{X} = \{\underline{c}_s' \underline{x} \leq, =, \geq b_s\};\ s = (1, S)$$

where $\underline{a}$ is a *p*-component vector of known coefficient, *e* is the random error component assumed normally and independently distributed with zero mean and constant variance, $\underline{c}_s$ is a component vector of known coefficients and $\underline{b}_s$ is a scalar for *s* number of constraints. The Variance Weighted Gradient (VWG) method for joint feasible regions with same constraints is defined by the following iterative steps:

i) Obtain N support points $\underline{x}_1, \underline{x}_2, \underline{x}_3, \cdots\cdots, \underline{x}_N$ from $\widetilde{X}$.

ii) Define the design measures $\xi_N^j$ which is made up of *N* support points such that

$$\xi_N^{(j)} = \begin{pmatrix} \frac{\underline{x}_1}{N} \\ \frac{\underline{x}_2}{N} \\ \vdots \\ \frac{\underline{x}_N}{N} \end{pmatrix} \ ;\ j = 0$$

where $N$ support points are spread evenly in $\widetilde{X}$

iii) From the support point *l* (1, N) that makes up the design measure, compute the starting points as.

$$\underline{\overline{x}}^* = (\underline{\overline{x}}_1, \underline{\overline{x}}_2, \cdots, \underline{\overline{x}}_n)'; \underline{\overline{x}}_i = \frac{\sum_{l=1}^{N} x_{il}}{N}$$

iv) Obtain the n-component gradient vector for the k$^{th}$ function.

$$\underline{g}_k = \left\{ \frac{\partial f_k(x)}{\partial x_i} \right\} = \begin{pmatrix} g_{k1}(x) \\ g_{k2}(x) \\ \vdots \\ g_{kn}(x) \end{pmatrix}$$

where

$g_{ki}(x) = \underline{q}' \underline{x} + e$ is an $(m-1)$ degree polynomial; $i = (1, n)$

$\underline{q}$ is a t-component vector of known coefficients

v) Compute the corresponding $k^{th}$ gradient vector, by substituting each *l* design point to the gradient function $\underline{g}_{ki}$ as

$$\underline{g}_{kl} = \begin{pmatrix} \underline{g}_1 \\ \underline{g}_2 \\ \vdots \\ \underline{g}_N \end{pmatrix}$$

vi) Using the gradient function and design measures obtain the corresponding design matrices $X_k$.

$$X_k(\xi_N^{(j)}) = \begin{pmatrix} x_{11} & x_{12} \cdots & x_{1t} \\ x_{21} & x_{22} \cdots & x_{2t} \\ \vdots & & \\ x_{N1} & x_{N2} \cdots & x_{Nt} \end{pmatrix} = \begin{pmatrix} \underline{\underline{x}}_1' \\ \underline{\underline{x}}_2' \\ \vdots \\ \underline{\underline{x}}_N' \end{pmatrix}$$

In order to form the design matrix $X_k$, a single polynomial that combines the respective gradient function $\underline{g}_{ki}(x)$ associated with each response function $f_k(x)$ is $\underline{g}_k(x)$.

vii) Compute the variances of each $l$ design point $\underline{x}_l$ for the $k^{th}$ function as $V_k$

$$V_k = \{ V_l = \underline{\underline{x}}_l' M_k^{-I} \underline{\underline{x}}_l \};\ l = (1, N);$$
$$M_k = X_k(\xi_N^{(j)})' X(\xi_N^{(j)})$$

viii) Obtain the direction vector for the $k^{th}$ function as

$$\underline{d}_k = \sum_{l=1}^{N} \theta_{kl} \underline{g}_{kl} \quad ;\ \theta_{kl} \in (0,1)$$

and the normalize direction vector $\underline{d}_k^* = \sum_{l=1}^{N_r} \theta_{kl}^* \underline{g}_{kl}$ such that $\underline{d}_k^{*\prime} \underline{d}_k = 1$.

ix) Compute the step-length $\rho_k^*$ as

$$\rho_k^* = \min_s \left\{ \frac{\underline{c}_{sk} \underline{\bar{x}}^* - b_{sk}}{\underline{c}_{sk} \underline{d}_k^*} \right\}$$

x) With $\underline{\bar{x}}^*$, $\rho_k^*$ and $\underline{d}_k^*$ make a move to

$$\underline{x}_{k,j}^* = \underline{\bar{x}}^* - \rho_k^* \underline{d}_k^*$$

xi) To make a next move set $j = j + 1$ and define the design measure as

$$\xi_{N+w}^{(j)} = \begin{pmatrix} \xi_N^{(j)} \\ \underline{x}_{1,j}^* \\ \underline{x}_{2,j}^* \\ \vdots \\ \underline{x}_{w,j}^* \end{pmatrix} \quad ;\ 1 \le w \le M$$

and repeat the process from step (iii) then obtain

$$\underline{x}_{k,j+1}^* = \underline{\bar{x}}^* - \rho_k^* \underline{d}_k^*$$

xii) If $f_k(\underline{x}_{k,j}^*) \le f_k(\underline{x}_{k,j+1}^*)$.

where $f_k(\underline{x}_{k,j}^*)$ is the $k^{th}$ function at the $j^{th}$ step and $f_k(\underline{x}_{k,j+1}^*)$ is the $k^{th}$ function at the next $(j+1)^{th}$ step.

then set the optimizers as

$$\underline{x}_{k,j+1}^* = \underline{x}_k^*$$

and STOP. Else, set $j = j + 1$ and repeat the process from (iii).

### 2.2. Variance Weighted Gradient (VWG) Method for Disjoint Feasible Regions with Different Constraints

Let

$$f_r(x) = \underline{a}' \underline{x} + e \qquad (2)$$

be the n-variate, p-parameter polynomial of degree m, defined on the $r^{th}$ feasible regions $\tilde{X}_r$ supported by s constraints. Such that

$$\underline{x} \in \tilde{X}_r = \{ \underline{c}_{sr}' \underline{x} \le, =, \ge b_{sr} \};$$
$$r = 1, 2, \cdots, R;\ s = 1, 2, \cdots, S$$

where $\underline{a}$ is a $p$-component vector of known coefficients, independent random variable $\underline{x} \in \tilde{X}_r$ and $e$ is the random error component assumed normally and independently distributed with zero mean and constant variance. While $\underline{c}_s$ is a component vector of know coefficients and $b_s$ is a scalar for $s$ number of constraints in the $r^{th}$ region.

The Variance Weighted Gradient (VWG) method for disjoint feasible regions, with different Constraints, is given by the following sequential steps:

i) From $\tilde{X}_r$ obtain the design measures $\xi_{rN_r}^j$ which are made up of support points from respective regions such that

$$\xi_{rN_r+j}^{(j)} = \begin{pmatrix} \dfrac{\underline{x}_{r1}}{N_r} \\ \dfrac{\underline{x}_{r2}}{N_r} \\ \vdots \\ \dfrac{\underline{x}_{rN_r}}{N_r} \end{pmatrix}$$

where $N_r$ support points are spread evenly in $\tilde{X}_r$

ii) From the support points that make up the design measure compute $R$ starting points as, the arithmetic mean vectors.

$$\underline{\bar{x}}_r^* = (\underline{\bar{x}}_{r1}, \underline{\bar{x}}_{r2}, \cdots, \underline{\bar{x}}_{rn})';\ \underline{\bar{x}}_{ri} = \frac{\sum_{l=1}^{N_r} x_{rli}}{N_r};\ \begin{pmatrix} i = (1, n) \\ l = (1, N_r) \\ r = (1, R) \end{pmatrix}$$

iii) Obtain the n-component gradient function for the $r^{th}$ region.

$$\underline{g}_r = \{\frac{\partial f_r(x)}{\partial x_i}\} = \begin{pmatrix} g_{r1}(x) \\ g_{r2}(x) \\ \vdots \\ g_{rn}(x) \end{pmatrix}$$

where

$g_{ri}(x) = \underline{q}' \underline{x} + e$ is an $(m-1)$ degree polynomial ; $i = (1, n)$

$\underline{q}$ is a t-component vector of known coefficients

iv) Compute the corresponding r gradient vectors, by substituting each design point defined on the r^th region to the gradient function $\underline{g}_r$ as

$$\underline{g}_{rl} = \begin{pmatrix} \underline{g}_{r1} \\ \underline{g}_{r2} \\ \vdots \\ \underline{g}_{rN_r} \end{pmatrix}$$

v) Using the gradient function and design measures obtain the corresponding design matrices $X_r$.

$$X_r(\xi_{rN_r+j}^{(j)}) = \begin{pmatrix} x_{r11} & x_{r12} \cdots & x_{r1t} \\ x_{r21} & x_{r22} \cdots & x_{r2t} \\ \vdots & & \\ x_{rN_r1} & x_{rN_r2} \cdots & x_{rN_rt} \end{pmatrix} = \begin{pmatrix} \underline{\underline{x}}_{r1}' \\ \underline{\underline{x}}_{r2}' \\ \vdots \\ \underline{\underline{x}}_{rN_r}' \end{pmatrix}$$

In order to form the design matrix $X_r$, a single polynomial that combines the respective gradient function $\underline{g}_{ri}(x)$ associated with each response function $f_r(x)$ is $\underline{g}_r(x)$.

vi) Compute the variances of each $l$ design point $\underline{x}_{rl}$ defined on the r^th region as

$$V_{rl} = \underline{\underline{x}}_{rl}' M_r^{-1} \underline{\underline{x}}_{rl} \;\; ; M_r = X_r(\xi_{rN_r+j}^{(j)})' X_r(\xi_{rN_r+j}^{(j)})$$

vii) Obtain the direction vector in the r^th region as

$$\underline{d}_r = \sum_{l=1}^{N_r} \theta_{rl} \underline{g}_{rl} \;\; ; \theta_{rl} \in (0,1)$$

and the normalize direction vector $\underline{d}_r^* = \sum_{l=1}^{N_r} \theta_{rl}^* \underline{g}_{rl}$ such that $\underline{d}_r^{*\prime} \underline{d}_r = 1$.

viii) Compute the step-length $\rho_r^*$ as

$$\rho_r^* = \min_s \left\{ \frac{\underline{c}_{sr} \underline{\bar{x}}_r^* - b_{sr}}{\underline{c}_{sr} \underline{d}_r^*} \right\}$$

with $\underline{\bar{x}}_r^*$, $\rho_r^*$ and $\underline{d}_r^*$ make a move to

$$\underline{x}_{r,j}^* = \underline{\bar{x}}_r^* - \rho_r^* \underline{d}_r^*$$

using $\underline{x}_{r,j}^*$ evaluate the projection operator $P_r$ as

$$P_r = \underline{x}_r^* (\underline{x}_r^{*\prime} \underline{x}_r^*) \underline{x}_r^{*\prime}$$

and obtain the projector optimizers for each region $\underline{\underline{x}}_r^*$ as

$$\underline{\underline{x}}_1^* = P_1 \underline{x}_R^*$$
$$\underline{\underline{x}}_2^* = P_2 \underline{x}_{R-1}^*$$
$$\vdots$$
$$\underline{\underline{x}}_{R-1}^* = P_{R-1} \underline{x}_2^*$$
$$\underline{\underline{x}}_R^* = P_R \underline{x}_1^*$$

ix) To make a next move set $j = j + 1$ and define the design measure as

$$\xi_{rN_r+j}^{(j)} = \begin{pmatrix} \xi_{rN_r}^{(0)} \\ \underline{\underline{x}}_r^* \end{pmatrix}$$

and repeat the process from step (ii) then obtain

$$\underline{x}_{r,j+1}^* = \underline{\bar{x}}_r^* - \rho_r^* \underline{d}_r^*$$

x) If $f_r(\underline{x}_{r,j}^*) \le f_r(\underline{x}_{r,j+1}^*)$.

where $f_r(\underline{x}_{r,j}^*)$ is the r^th feasible region at the j^th step and $f_r(\underline{x}_{r,j+1,})$ is the r^th feasible region at the next j+1^th step.

then set the optimizers as

$$\underline{x}_{r,j+1}^* = \underline{x}_r^*$$

and STOP. Else, set $j = j + 1$ and repeat the process from (ii).

# 3. Results

In comparing the Variance Weighted Gradient (VWG) algorithms with standard gradient and non-gradient methods, we consider the optimization problems;

**Problem 1**

Minimize $f_1(x_1, x_2) = 2x_1^2 - 2x_1x_2 + 2x_2^2 - 6x_1 + 6$

subject to $\widetilde{X} = \left\{ \begin{matrix} x_1 + x_2 \le 2 \\ x_1, x_2 \ge 0 \end{matrix} \right\}$

(Ghadle and Pawar, 2015)

and

Maximize $f_2(x_1, x_2) = 2x_1 + 3x_2 - x_1^2 - x_2^2$ subject to

$$\widetilde{X} = \begin{Bmatrix} x_1 + x_2 \leq 2 \\ x_1, x_2 \geq 0 \end{Bmatrix}$$

(Hillier and Lieberman, 2001).

The solution to the minimization problem $f_1(x_1, x_2) = 2x_1^2 - 2x_1x_2 + 2x_2^2 - 6x_1 + 6$ using the Variance Weighted Gradient (VWG) method is $x_1 = 1.4960$, $x_2 = 0.5030$ and $f_1(x_1, x_2) = 0.5000$, from an initial starting point, $x_1 = 0.6600$, $x_2 = 0.6600$. Similarly, the solution to the maximization problem $f_2(x_1, x_2) = 2x_1 + 3x_2 - x_1^2 - x_2^2$ using the Variance Weighted Gradient (VWG) method is $x_1 = 0.7560$, $x_2 = 1.2430$ and $f_2(x_1, x_2) = 3.1249$, from an initial starting point, ($x_1 = 0.6600$, $x_2 = 0.6600$). QN, GA, MADS and GPS methods obtained the respective solutions {$x_1 = 1.5000$, $x_2 = 0.5000$, $f_1(x_1, x_2) = 0.5000$}, {$x_1 = 1.5040$, $x_2 = 0.4950$, $f_1(x_1, x_2) = 0.5001$}, {$x_1 = 1.5000$, $x_2 = 0.5000$, $f_1(x_1, x_2) = 0.5000$}, {$x_1 = 1.5000$, $x_2 = 0.5000$, $f_1(x_1, x_2) = 0.5000$} to the minimization problem $f_1(x_1, x_2) = 2x_1^2 - 2x_1x_2 + 2x_2^2 - 6x_1 + 6$ from respective initial guess value, ($x_1 = 0.0000$, $x_2 = 1.0000$), ($x_1 = 0.0000$, $x_2 = 1.0000$), ($x_1 = 0.0000$, $x_2 = 0.0000$), ($x_1 = 0.0000$, $x_2 = 0.0000$). Similarly, QN, GA, MADS and GPS methods obtained the respective solutions {$x_1 = 0.7500$, $x_2 = 1.2500$, $f_2(x_1, x_2) = 3.1250$}, {$x_1 = 0.7540$, $x_2 = 1.2500$, $f_2(x_1, x_2) = 3.1249$}, {$x_1 = 0.7570$, $x_2 = 1.2420$, $f_2(x_1, x_2) = 3.1244$}, {$x_1 = 0.7500$, $x_2 = 1.2500$, $f_2(x_1, x_2) = 3.1250$} to the maximization problem $f_2(x_1, x_2) = 2x_1 + 3x_2 - x_1^2 - x_2^2$ from respective initial guess value, ($x_1 = 0.0000$, $x_2 = 0.0000$), ($x_1 = 0.0000$, $x_2 = 0.0000$), ($x_1 = 0.0000$, $x_2 = 0.0000$), ($x_1 = 1.0000$, $x_2 = 0.0000$). The summary results are as tabulated in Tables 1 and 2.

**Problem 2**

Maximize $f_3(x_1, x_2) = 5x_1 - x_1^2 + 8x_2 - 2x_2^2$ subject to

$$\widetilde{X}_2 = \{3x_1 + 3x_2 \leq 6; x_1, x_2 \geq 0\}$$

(Hillier and Lieberman, 2001)

and

Maximize $f_4(x_1, x_2) = x_1 + 2x_2 - x_2^3$

subject to

$$\widetilde{X}_3 = \{x_1 + x_2 \leq 1; x_1, x_2 \geq 0\}$$

(Hillier and Lieberman, 2001)

The solution to the maximization problem $f_3(x_1, x_2) = 5x_1 - x_1^2 + 8x_2 - 2x_2^2$ using the Variance Weighted Gradient (VWG) method is $x_1 = 0.9800$, $x_2 = 1.5290$ and $f_3(x_1, x_2) = 11.4977$, from an initial starting point, $x_1 = 0.7500$, $x_2 = 0.7900$. Similarly, the solution to the maximization problem $f_4(x_1, x_2) = x_1 + 2x_2 - x_2^3$ using the Variance Weighted Gradient (VWG) method is $x_1 = 0.4280$, $x_2 = 0.5710$ and $f_4(x_1, x_2) = 1.3848$, from an initial starting point, ($x_1 = 0.2700$, $x_2 = 0.3300$). QN, GA, MADS and GPS methods obtained the respective solutions {$x_1 = 1.0000$, $x_2 = 1.5000$, $f_3(x_1, x_2) = 11.4999$}, {$x_1 = 0.9960$, $x_2 = 1.5040$, $f_3(x_1, x_2) = 11.4999$}, {$x_1 = 1.0000$, $x_2 = 1.5000$, $f_3(x_1, x_2) = 11.5000$}, {$x_1 = 1.0000$, $x_2 = 1.5000$, $f_3(x_1, x_2) = 11.5000$} from respective initial guess value, ($x_1 = 0.0000$, $x_2 = 1.0000$), ($x_1 = 0.0000$, $x_2 = 1.0000$), ($x_1 = 1.0000$, $x_2 = 0.0000$), ($x_1 = 0.0000$, $x_2 = 1.0000$) to the maximization problem $f_3(x_1, x_2) = 5x_1 - x_1^2 + 8x_2 - 2x_2^2$. In like manner, QN, GA, MADS and GPS methods obtained the respective solutions for {$x_1 = 0.4220$, $x_2 = 0.5770$, $f_4(x_1, x_2) = 1.3849$}, {$x_1 = 0.4220$, $x_2 = 0.5770$, $f_4(x_1, x_2) = 1.3849$}, {$x_1 = 0.4290$, $x_2 = 0.5700$, $f_4(x_1, x_2) = 1.3848$}, {$x_1 = 0.4220$, $x_2 = 0.5770$, $f_4(x_1, x_2) = 1.3849$} from respective initial guess value, ($x_1 = 0.0000$, $x_2 = 1.0000$), ($x_1 = 0.0000$, $x_2 = 0.0000$), ($x_1 = 0.5000$, $x_2 = 0.5000$), ($x_1 = 1.0000$, $x_2 = 0.0000$) to the maximization problem $f_4(x_1, x_2) = x_1 + 2x_2 - x_2^3$.

The summary results are as tabulated in Tables 3 and 4.

Solutions involving Quasi-Newton's Method (QNM), Genetic Algorithm (GA), Mesh Adaptive Search (MADS) and Generalized Pattern Search (GPS) algorithms were obtained with the aid of optimization tool and pattern tool in MATLAB version R2007b software. The MATLAB outputs are in Appendices A-D.

**Table 1.** Minimization of $f_1(x_1, x_2) = 2x_1^2 - 2x_1x_2 + 2x_2^2 - 6x_1 + 6$ Subject to $\tilde{X} = \{x_1, x_2 : x_1 + x_2 \leq 2; x_1, x_2 \geq 0\}$

| Methods | Initial Starting point ($x_1$, $x_2$) | | Number of Iterations | Optimizers ($x_1$, $x_2$) | | $f_1(x_1x_2)$ |
|---|---|---|---|---|---|---|
| Quasi-Newton's Method (QNM) | 0.0000 | 1.0000 | 4 | 1.5000 | 0.5000 | 0.5000 |
| Genetic Algorithm (GA) | 0.0000 | 1.0000 | 51 | 1.5040 | 0.4950 | 0.5001 |
| Mesh Adaptive Search (MADS) | 0.0000 | 0.0000 | 25 | 1.5000 | 0.5000 | 0.5000 |
| Generalized Pattern Search (GPS) | 0.0000 | 0.0000 | 26 | 1.5000 | 0.5000 | 0.5000 |
| Variance Weighted Gradient (VWG) | 0.6600 | 0.6600 | 5 | 1.4960 | 0.5030 | 0.5000 |

**Table 2.** Maximization of $f_2(x_1, x_2) = 2x_1 + 3x_2 - x_1^2 - x_2^2$ Subject to $\tilde{X} = \{x_1, x_2 : x_1 + x_2 \le 2 \; ; x_1, x_2 \ge 0\}$

| Methods | Initial Starting point ($x_1$, $x_2$) | | Number of Iterations | Optimizers ($x_1$, $x_2$) | | $f_1(x_1x_2)$ |
|---|---|---|---|---|---|---|
| Quasi-Newton's Method (QNM) | 0.0000 | 0.0000 | 3 | 0.7500 | 1.2500 | 3.1250 |
| Genetic Algorithm (GA) | 0.0000 | 0.0000 | 51 | 0.7540 | 1.2500 | 3.1249 |
| Mesh Adaptive Search (MADS) | 0.0000 | 0.0000 | 135 | 0.7570 | 1.2420 | 3.1244 |
| Generalized Pattern Search (GPS) | 1.0000 | 0.0000 | 24 | 0.7500 | 1.2500 | 3.1250 |
| Variance Weighted Gradient (VWG) | 0.6600 | 0.6600 | 3 | 0.7560 | 1.2430 | 3.1249 |

**Table 3.** Minimization of $f_3(x_1, x_2) = 5x_1 - x_1^2 + 8x_2 - 2x_2^2$ Subject to $\tilde{X}_2 = \{3x_1 + 3x_2 \le 6 \; ; x_1, x_2 \ge 0\}$

| Methods | Initial Starting point ($x_1$, $x_2$) | | Number of Iterations | Optimizers ($x_1$, $x_2$) | | $f_3(x_1x_2)$ |
|---|---|---|---|---|---|---|
| Quasi-Newton's Method (QNM) | 0.0000 | 1.0000 | 4 | 1.0000 | 1.5000 | 11.4999 |
| Genetic Algorithm (GA) | 0.0000 | 1.0000 | 51 | 0.9960 | 1.5040 | 11.4999 |
| Mesh Adaptive Search (MADS) | 1.0000 | 0.0000 | 23 | 1.0000 | 1.5000 | 11.5000 |
| Generalized Pattern Search (GPS) | 0.0000 | 1.0000 | 24 | 1.0000 | 1.5000 | 11.5000 |
| Variance Weighted Gradient (VWG) | 0.7500 | 0.7900 | 2 | 0.9800 | 1.5290 | 11.4977 |

**Table 4.** Minimization of $f_4(x_1, x_2) = x_1 + 2x_2 - x_2^3$ Subject to $\tilde{X}_3 = \{x_1 + x_2 \le 1 \; ; x_1, x_2 \ge 0\}$

| Methods | Initial Starting point ($x_1$, $x_2$) | | Number of Iterations | Optimizers ($x_1$, $x_2$) | | $f_1(x_1x_2)$ |
|---|---|---|---|---|---|---|
| Quasi-Newton's Method (QNM) | 0.0000 | 1.0000 | 5 | 0.4220 | 0.5770 | 1.3849 |
| Genetic Algorithm (GA) | 0.0000 | 0.0000 | 51 | 0.4220 | 0.5770 | 1.3848 |
| Mesh Adaptive Search (MADS) | 0.5000 | 0.5000 | 109 | 0.4290 | 0.5700 | 1.3848 |
| Generalized Pattern Search (GPS) | 1.0000 | 0.0000 | 40 | 0.4220 | 0.5770 | 1.3849 |
| Variance Weighted Gradient (VWG) | 0.2700 | 0.3300 | 3 | 0.4280 | 0.5710 | 1.3848 |

## 4. Discussion of Results

In minimizing $f_1(x_1, x_2) = 2x_1^2 - 2x_1x_2 + 2x_2^2 - 6x_1 + 6$ subject to $x_1 + x_2 \le 2$ and $x_1, x_2 \ge 0$, the Quasi-Newton Method (QNM) with the initial guess starting point (0.0000, 1.0000) locates the optimizer (1.5000, 0.5000) in 4 iterations with a response function value of 0.5000. Genetic Algorithm (GA) with initial guess starting points (0.0000, 1.0000) locates the optimizer (1.5040, 0.4950) in 51 iterations with a response function value of 0.5001. The Mesh Adaptive Search (MADS) and Generalized Pattern Search (GPS) methods, with initial guess starting point (0.0000, 0.0000), locate the same optimizer (1.5000, 0.5000) with a response function value of 0.5000 in 25 and 26 iterations, respectively. The Variance Weighted Gradient Method (VWG) obtained optimizers (1.4960, 0.5030) in 5 iterations, with a response function value of 0.5000 from an initial optimum starting point (0.66, 0.66).

In maximizing $f_2(x_1, x_2) = 2x_1 + 3x_2 - x_1^2 - x_2^2$ subject to $x_1 + x_2 \le 2$ and $x_1, x_2 \ge 0$, the Quasi-Newton Method (QNM) method, with the initial guess starting point (0.0000, 0.0000) locates the optimizer (0.7500, 1.2500) in 3 iterations with a response function value of 3.1250. The Genetic Algorithm (GA), with the initial guess starting point (0.0000, 0.0000) locates the optimizer (0.7540, 1.2500) in 51 iterations with a response function value of 3.1249. The Mesh Adaptive Search (MADS) method, with the initial guess starting point (0.0000, 0.0000) locates the optimizer (0.7570, 1.2420) in 135 iterations with a response function value of 3.1244. The Generalized Pattern Search (GPS) method, with the initial guess starting point (1.0000, 0.0000) locates the optimizer (0.7500, 1.2500) in 24 iterations with a response function value of 3.1250. The Variance Weighted Gradient Method (VWG) obtained optimizers (0.7560, 1.2430) in 3 iterations, with a response function value of 3.1249 from an initial optimum starting point (0.66, 0.66).

In maximizing $f_3(x_1, x_2) = 5x_1 - x_1^2 + 8x_2 - 2x_2^2$ subject to $3x_1 + 3x_2 \leq 6$ and $x_1, x_2 \geq 0$, the Quasi-Newton Method (QNM) with the initial guess starting point (0.0000, 1.0000) locates the optimizer (1.0000, 1.5000) in 4 iterations with a response function value of 11.4999. Genetic Algorithm (GA) with initial guess starting point (0.0000, 1.0000) locates the optimizer (0.9960, 1.5040) in 51 iterations, with response function value 11.4999. The Mesh Adaptive Search (MADS) method with initial guess starting point (1.0000, 0.0000) locates the optimizer (1.0000, 1.5000) in 23 iterations, with response function value 11.5000. The Generalized Pattern Search (GPS) method with initial guess starting point (0.0000, 1.0000) locates the optimizer (1.0000, 1.5000) in 24 iterations, with response function value of 11.5000. The Variance Weighted Gradient Method (VWG) obtained optimizers (0.9800, 1.5290) from an initial optimum starting point (0.7500, 0.7900) in 2 iterations, with response function value of 11.4977.

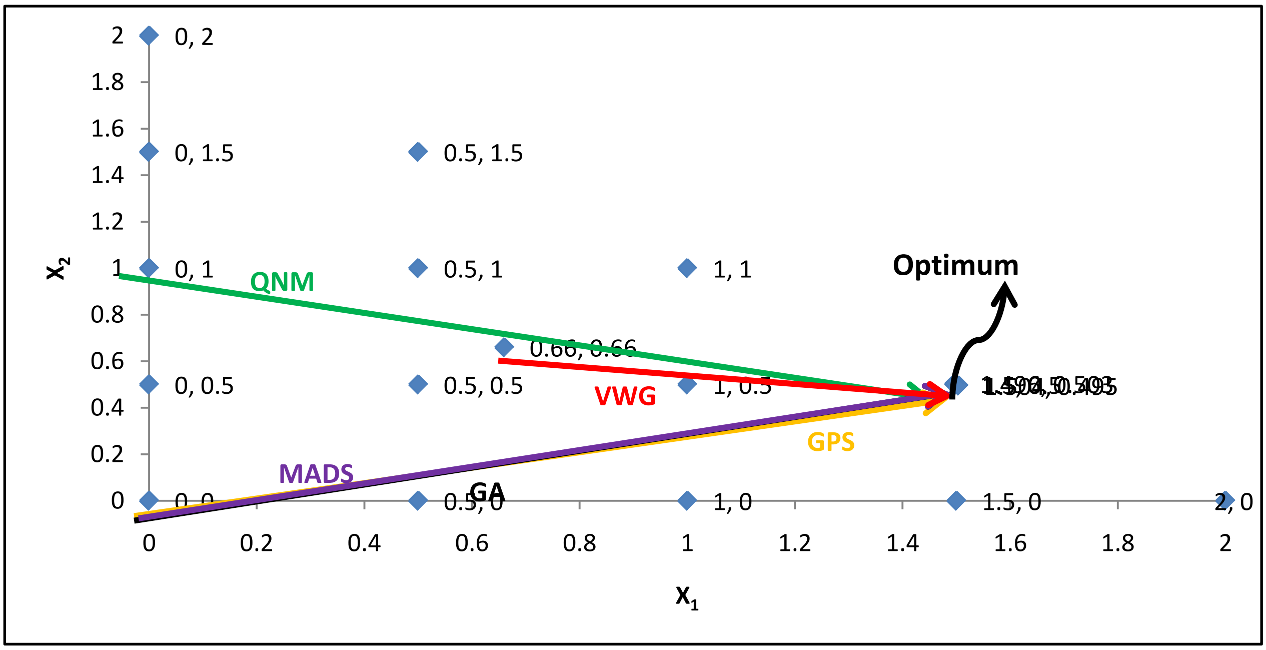


**Figure 1.** Optimum Point of $f_1(x_1, x_2) = 2x_1^2 - 2x_1x_2 + 2x_2^2 - 6x_1 + 6$ Subject to $\tilde{X} = \{x_1, x_2 : x_1 + x_2 \leq 2; x_1, x_2 \geq 0\}$

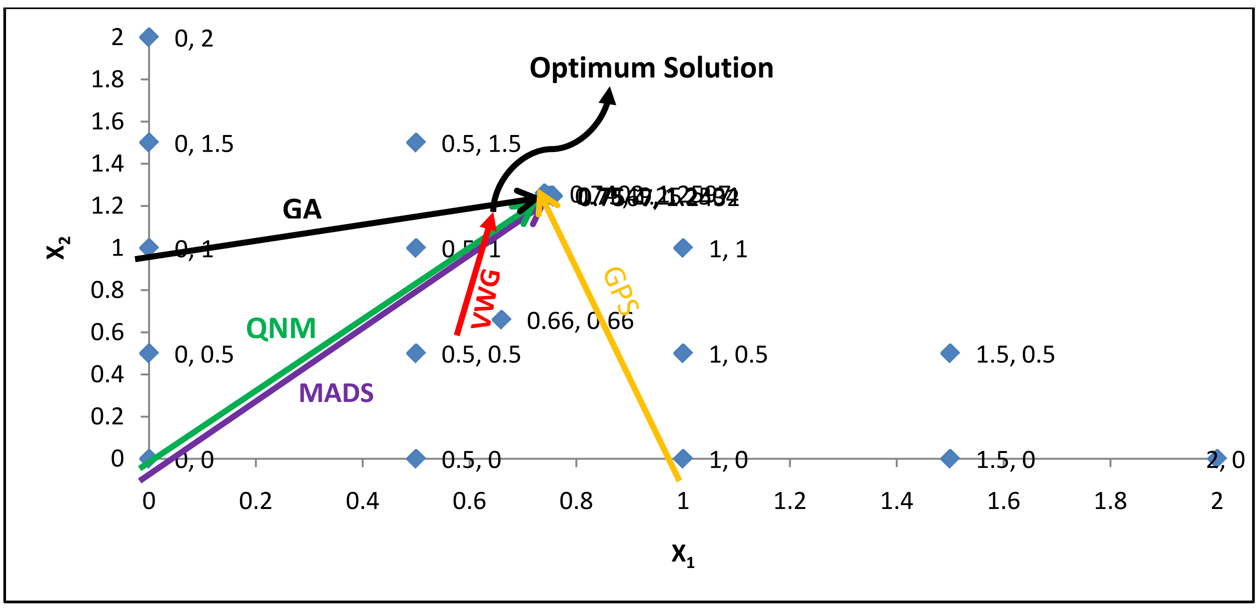


**Figure 2.** Optimum Point of $f_2(x_1, x_2) = 2x_1 + 3x_2 - x_1^2 - x_2^2$ Subject to $\tilde{X} = \{x_1, x_2 : x_1 + x_2 \leq 2; x_1, x_2 \geq 0\}$

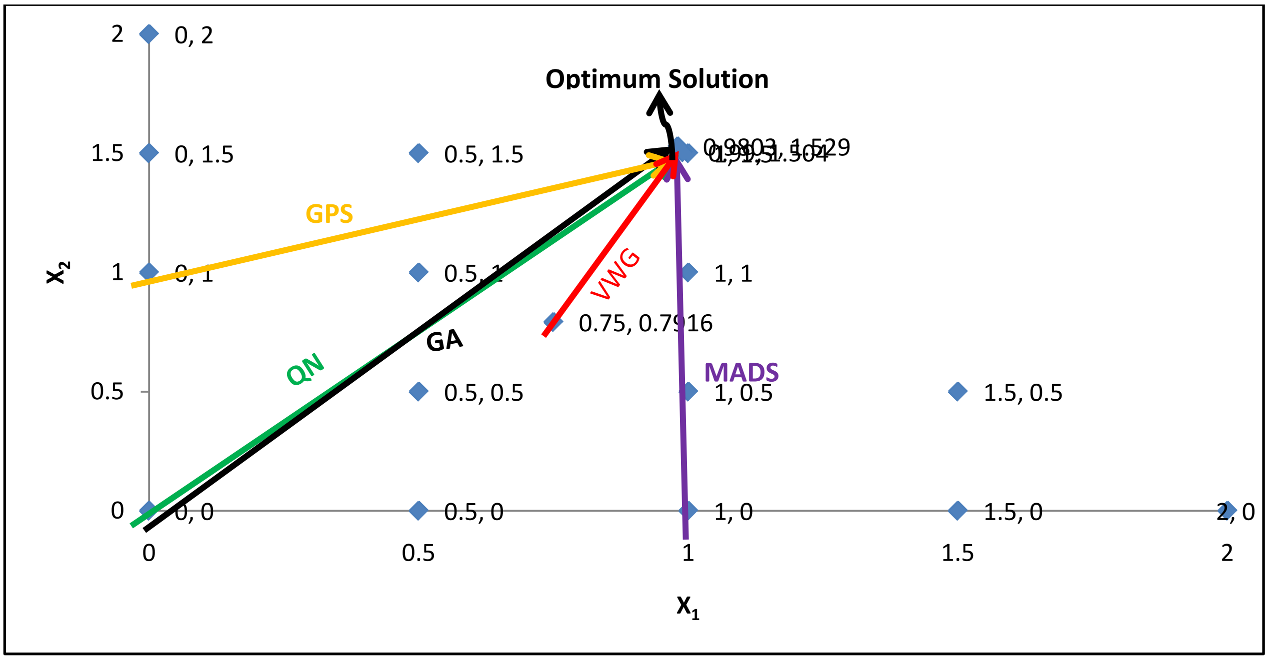


**Figure 3.** Optimum Point of $f_3(x_1,x_2) = 5x_1 - x_1^2 + 8x_2 - 2x_2^2$ Subject to $\tilde{X}_2 = \{3x_1 + 3x_2 \le 6\,; x_1, x_2 \ge 0\}$

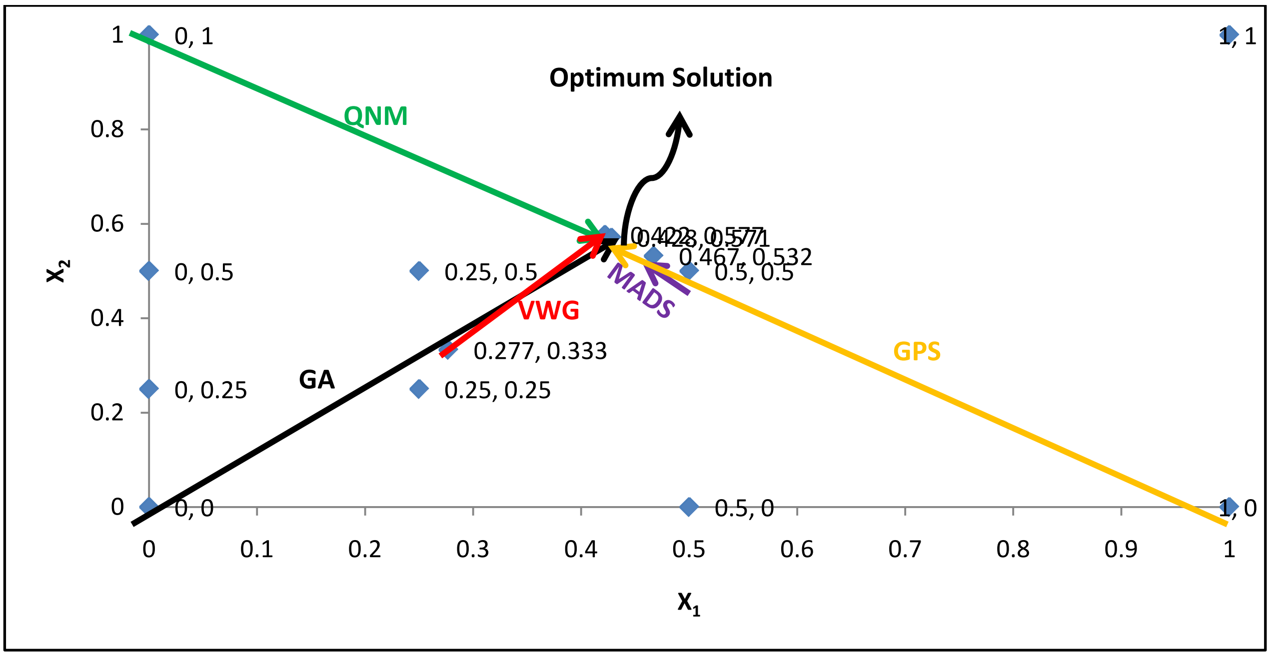


**Figure 4.** Optimum Point of $f_4(x_1,x_2) = x_1 + 2x_2 - x_2^3$ subject to $\tilde{X}_3 = \{x_1 + x_2 \le 1\,; x_1, x_2 \ge 0\}$

In maximizing $f_4(x_1,x_2) = x_1 + 2x_2 - x_2^3$ subject to $x_1 + x_2 \le 1$ and $x_1, x_2 \ge 0$, the Quasi-Newton Method (QNM) with the initial guess starting point (0.0000, 1.0000) locates the optimizer (0.4220, 0.5770) in 5 iterations with a response function value of 1.3849. Genetic Algorithm (GA) with the initial guess starting point (0.0000, 0.0000) locates the optimizer (0.4220, 0.5770) in 51 iterations with a response function value of 1.3849. The Mesh Adaptive Search (MADS) method with the initial guess starting point (0.5000, 0.5000) locates the optimizer (0.4290, 0.5770) in 109 iterations with a response function value of 1.3848. The Generalized Pattern Search (GPS) method with the initial guess starting point (1.0000, 0.0000) locates the optimizer (0.4220, 0.5770) in 40 iterations with a response function value of 1.3849. The Variance Weighted Gradient Method (VWG) obtained optimizers (0.4280, 0.5710) with the initial starting point (0.2700, 0.3300) in 3 iterations with a response function value of 1.3849.

The VWG method has the ability to obtain optimizers for polynomial response functions defined on joint and disjoint feasible regions simultaneously in either the first or second iterations. The comparative assessment shows that results obtained using the VWG simultaneous optimization are comparatively efficient in locating the optimizers of several response surfaces. The norm of the optimizers obtained using the VWG methods relative to the existing methods is very small, with the maximum recorded as 0.0352. Also, the absolute difference between the values of the response functions obtained using the new method relative to existing methods are approximately zero. The present study raises the possibility that optimizers of multifunction defined on joint feasible region can be added to the design points of the region and also the optimizers of multifunction defined on one feasible region can be projected to another feasible region in obtaining optimal solutions. We propose that further research should be undertaken to investigate multifunction polynomial response surfaces defined by constraints on different feasible regions and multifunction polynomial response surfaces for unconstraint.

## 5. Conclusions

The variance weighted gradient (VWG) methods are suitable for optimizing polynomial response surfaces defined on the same or different feasible experimental regions. For both cases, the starting point of search is not a guess point as commonly seen in many gradient and non-gradient algorithms. Although the two variance weighted gradient methods simultaneously optimize multiobjective functions, in handling problems involving different regions and different constraints, a projection scheme that allows the projection of design points from one design region to another is used. The projection scheme enhances fast convergence of the algorithm to the desired optima as measured by the number of iterative moves made. The results of this study establish that the variance weighted gradient methods are reliable optimization methods for optimizing polynomial response surfaces defined by constraints on joint and disjoint feasible regions, when compared with Quasi-Newton's Method, Genetic Algorithm, Mesh Adaptive Search and Generalized Pattern Search method. Interestingly, VWG algorithms required few numbers of iterative steps in obtaining the optimizers of the polynomial response functions.

## APPENDIX A

Minimization of $f_1(x_1, x_2) = 2x_1^2 - 2x_1x_2 + 2x_2^2 - 6x_1 + 6$

subject to

$$\widetilde{X} = \begin{Bmatrix} x_1 + x_2 \le 2 \\ x_1, x_2 \ge 0 \end{Bmatrix}$$

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%

**QUASI-NEWTON METHOD (QNM)**

Diagnostic Information

Number of variables: 2
Functions
 Objective: function 1
 Gradient: finite-differencing
 Hessian: finite-differencing (or
Quasi-Newton)
Constraints
 Nonlinear constraints: do not exist
 Number of linear inequality constraints: 1
 Number of linear equality constraints: 0
 Number of lower bound constraints: 0
 Number of upper bound constraints: 0
Algorithm selected
 medium-scale
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%
 End diagnostic information
 Max Line search Directional
First-order
 Iter F-count f(x) constraint steplength derivative
optimality Procedure
 0 3 6 -2
 1 8 3.5 -1.5 0.25 4 3
 2 11 0.514139 -2.22e-016 1 -1.33
1.17
 3 14 0.500197 0 1 0.00373
0.0481
 4 17 0.5 0 1 2.22e-010
3.88e-008
Optimization terminated: first-order optimality measure
less
 than options.TolFun and maximum constraint violation is
less
 than options.TolCon.
Active inequalities (to within options.TolCon = 1e-006):
 lower upper ineqlin ineqnonlin
 1
> >> [x,fval]=fmincon('p',[0,0],[],[],[1,1],2,[0,0],[inf,inf])
Warning: Large-scale (trust region) method does not
currently solve this type of
problem,
 using medium-scale (line search) instead.
> In fmincon at 317
Optimization terminated: first-order optimality measure
less
 than options.TolFun and maximum constraint violation is
less
 than options.TolCon.
No active inequalities.
x =
 1.5000 0.5000
fval =
 0.5000

**GENETIC ALGORITHM (GA)**

Diagnostic information.
 Fitness function = @function 1

```
 Number of variables = 2
 1 Inequality constraints
 0 Equality constraints
 1 Total number of linear constraints
Modified options:
 options.Display = 'diagnose'
 options.OutputFcns = { { @gatooloutput } }
End of diagnostic information.
```

| | | Best | Mean | Stall |
|---|---|---|---|---|
| Generation | f-count | f(x) | f(x) | Generations |
| 1 | 21 | 14.41 | 24.14 | 0 |
| 2 | 41 | 5.695 | 23.09 | 0 |
| 3 | 61 | 3.785 | 19.98 | 0 |
| 4 | 81 | 3.785 | 15.05 | 1 |
| 5 | 101 | 3.785 | 11.91 | 2 |
| 6 | 121 | 3.785 | 9.565 | 3 |
| 7 | 141 | 3.785 | 6.266 | 4 |
| 8 | 161 | 3.583 | 5.986 | 0 |
| 9 | 181 | 3.03 | 4.746 | 0 |
| 10 | 201 | 2.612 | 4.141 | 0 |
| 11 | 221 | 2.103 | 3.175 | 0 |
| 12 | 241 | 1.018 | 2.861 | 0 |
| 13 | 261 | 1.018 | 2.574 | 1 |
| 14 | 281 | 0.8612 | 2.062 | 0 |
| 15 | 301 | 0.8612 | 2.202 | 1 |
| 16 | 321 | 0.8612 | 1.716 | 2 |
| 17 | 341 | 0.8518 | 1.544 | 0 |
| 18 | 361 | 0.7691 | 1.468 | 0 |
| 19 | 381 | 0.7691 | 1.853 | 1 |
| 20 | 401 | 0.7691 | 1.763 | 2 |
| 21 | 421 | 0.5449 | 0.9965 | 0 |
| 22 | 441 | 0.5449 | 1.045 | 1 |
| 23 | 461 | 0.5317 | 0.9906 | 0 |
| 24 | 481 | 0.5317 | 0.9762 | 1 |
| 25 | 501 | 0.5317 | 0.9622 | 2 |
| 26 | 521 | 0.5304 | 0.7351 | 0 |
| 27 | 541 | 0.5304 | 0.7531 | 1 |
| 28 | 561 | 0.5304 | 0.6162 | 2 |
| 29 | 581 | 0.5006 | 0.5786 | 0 |
| 30 | 601 | 0.5006 | 0.5881 | 1 |

| | | Best | Mean | Stall |
|---|---|---|---|---|
| Generation | f-count | f(x) | f(x) | Generations |
| 31 | 621 | 0.5006 | 0.5687 | 2 |
| 32 | 641 | 0.5006 | 0.533 | 3 |
| 33 | 661 | 0.5004 | 0.5166 | 0 |
| 34 | 681 | 0.5004 | 0.5146 | 1 |
| 35 | 701 | 0.5004 | 0.5139 | 2 |
| 36 | 721 | 0.5003 | 0.509 | 0 |
| 37 | 741 | 0.5001 | 0.5088 | 0 |
| 38 | 761 | 0.5001 | 0.516 | 1 |
| 39 | 781 | 0.5001 | 0.5114 | 2 |
| 40 | 801 | 0.5001 | 0.5044 | 3 |
| 41 | 821 | 0.5001 | 0.5023 | 4 |
| 42 | 841 | 0.5001 | 0.5017 | 0 |
| 43 | 861 | 0.5001 | 0.5016 | 0 |
| 44 | 881 | 0.5001 | 0.5027 | 1 |
| 45 | 901 | 0.5001 | 0.5019 | 2 |
| 46 | 921 | 0.5001 | 0.5015 | 3 |
| 47 | 941 | 0.5001 | 0.5012 | 4 |
| 48 | 961 | 0.5001 | 0.5007 | 0 |
| 49 | 981 | 0.5001 | 0.5001 | 0 |
| 50 | 1001 | 0.5001 | 0.5003 | 0 |
| 51 | 1021 | 0.5001 | 0.5011 | 1 |

```
Optimization terminated: average change in the fitness value less than options.TolFun.
>>
x =
  1.50405   0.49595
fval =
  0.5001011
```

**GENERALISED PATTERN SEARCH (GPS)**

```
Diagnostic information.
 objective function = @function 1
 1 Inequality constraints
 0 Equality constraints
 1 Total number of linear constraints
 X0 = [ 0 0 ]
Modified options:
 options.PollMethod = 'GPSPositiveBasis2N'
 options.CompletePoll = 'on'
 options.SearchMethod = @GPSPositiveBasis2N
 options.CompleteSearch = 'on'
 options.Display = 'diagnose'
 options.OutputFcns = { { @psearchtooloutput } }
End of diagnostic information.
```

| Iter | f-count | f(x) | MeshSize | Method |
|---|---|---|---|---|
| 0 | 1 | 6 | 1 | |
| 1 | 5 | 2 | 2 | Successful Poll |
| 2 | 7 | 2 | 1 | Refine Mesh |
| 3 | 11 | 2 | 0.5 | Refine Mesh |
| 4 | 15 | 1.5 | 1 | Successful Poll |
| 5 | 17 | 1.5 | 0.5 | Refine Mesh |
| 6 | 21 | 0.5 | 1 | Successful Poll |
| 7 | 26 | 0.5 | 0.5 | Refine Mesh |
| 8 | 31 | 0.5 | 0.25 | Refine Mesh |
| 9 | 36 | 0.5 | 0.125 | Refine Mesh |
| 10 | 41 | 0.5 | 0.0625 | Refine Mesh |
| 11 | 46 | 0.5 | 0.03125 | Refine Mesh |
| 12 | 51 | 0.5 | 0.01563 | Refine Mesh |
| 13 | 56 | 0.5 | 0.007813 | Refine Mesh |
| 14 | 61 | 0.5 | 0.003906 | Refine Mesh |
| 15 | 66 | 0.5 | 0.001953 | Refine Mesh |
| 16 | 71 | 0.5 | 0.0009766 | Refine Mesh |
| 17 | 76 | 0.5 | 0.0004883 | Refine Mesh |
| 18 | 81 | 0.5 | 0.0002441 | Refine Mesh |
| 19 | 86 | 0.5 | 0.0001221 | Refine Mesh |
| 20 | 91 | 0.5 | 6.104e-005 | Refine Mesh |
| 21 | 96 | 0.5 | 3.052e-005 | Refine Mesh |
| 22 | 101 | 0.5 | 1.526e-005 | Refine Mesh |
| 23 | 106 | 0.5 | 7.629e-006 | Refine Mesh |
| 24 | 111 | 0.5 | 3.815e-006 | Refine Mesh |
| 25 | 116 | 0.5 | 1.907e-006 | Refine Mesh |

```
   26      121           0.5    9.537e-007      Refine Mesh
Optimization terminated: mesh size less than
options.TolMesh.
x =
   1.5    0.5
fval =
   0.5
```

**MESH ADAPTIVE SEARCH (MADS)**

```
Diagnostic information.
  objective function = @ function 1
  1 Inequality constraints
  0 Equality constraints
  1 Total number of linear constraints
  X0 = [ 0 0 ]
Modified options:
  options.PollMethod = 'MADSPositiveBasis2N'
  options.CompletePoll = 'on'
  options.SearchMethod = @MADSPositiveBasis2N
  options.CompleteSearch = 'on'
  options.Display = 'diagnose'
  options.OutputFcns = { { @psearchtooloutput } }
End of diagnostic information.
```

| Iter | f-count | f(x) | MeshSize | Method |
|---|---|---|---|---|
| 0 | 1 | 6 | 1 | |
| 1 | 9 | 2 | 1 | Successful Poll |
| 2 | 12 | 2 | 0.25 | Refine Mesh |
| 3 | 20 | 1.5 | 1 | Successful Poll |
| 4 | 22 | 1.5 | 0.25 | Refine Mesh |
| 5 | 30 | 0.5 | 1 | Successful Poll |
| 6 | 33 | 0.5 | 0.25 | Refine Mesh |
| 7 | 36 | 0.5 | 0.0625 | Refine Mesh |
| 8 | 39 | 0.5 | 0.01563 | Refine Mesh |
| 9 | 42 | 0.5 | 0.003906 | Refine Mesh |
| 10 | 45 | 0.5 | 0.0009766 | Refine Mesh |
| 11 | 48 | 0.5 | 0.0002441 | Refine Mesh |
| 12 | 51 | 0.5 | 6.104e-005 | Refine Mesh |
| 13 | 54 | 0.5 | 1.526e-005 | Refine Mesh |
| 14 | 57 | 0.5 | 3.815e-006 | Refine Mesh |
| 15 | 60 | 0.5 | 9.537e-007 | Refine Mesh |
| 16 | 63 | 0.5 | 2.384e-007 | Refine Mesh |
| 17 | 66 | 0.5 | 5.96e-008 | Refine Mesh |
| 18 | 69 | 0.5 | 1.49e-008 | Refine Mesh |
| 19 | 72 | 0.5 | 3.725e-009 | Refine Mesh |
| 20 | 75 | 0.5 | 9.313e-010 | Refine Mesh |
| 21 | 78 | 0.5 | 2.328e-010 | Refine Mesh |
| 22 | 81 | 0.5 | 5.821e-011 | Refine Mesh |
| 23 | 84 | 0.5 | 1.455e-011 | Refine Mesh |
| 24 | 87 | 0.5 | 3.638e-012 | Refine Mesh |
| 25 | 90 | 0.5 | 9.095e-013 | Refine Mesh |

```
Optimization terminated: mesh size less than
'sqrt(TolMesh)'.
x =
   1.5    0.5
fval =
   0.5
```

# Appendix B

Maximization of $f_2(x_1, x_2) = 2x_1 + 3x_2 - x_1^2 - x_2^2$

subject to

$$\widetilde{X} = \begin{Bmatrix} x_1 + x_2 \le 2 \\ x_1, x_2 \ge 0 \end{Bmatrix}$$

**QUASI-NEWTON METHOD (QNM)**

```
  Diagnostic Information
Number of variables: 2
Functions
  Objective:                      function 2
  Gradient:                        finite-differencing
  Hessian:                        finite-differencing (or
Quasi-Newton)
Constraints
  Nonlinear constraints:             do not exist
  Number of linear inequality constraints:    1
  Number of linear equality constraints:      0
  Number of lower bound constraints:         0
  Number of upper bound constraints:         0
Algorithm selected
    medium-scale
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%
  End diagnostic information
                                Max      Line search Directional
First-order
  Iter F-count       f(x)    constraint    steplength    derivative
optimality Procedure
    0      3           -1           -1
    1      6           -2            0           1           4           2
    2      9       -3.12188          0           1       0.125
0.316
    3      12       -3.125    -2.22e-016          1    1.89e-009
6.18e-008
Optimization terminated: first-order optimality measure
less
  than options.TolFun and maximum constraint violation is
less
  than options.TolCon.
Active inequalities (to within options.TolCon = 1e-006):
  lower      upper      ineqlin     ineqnonlin
                          1
                        1
>> [x,fval]=fmincon('p',[0,0],[],[],[1,1],2,[0,0],[inf,inf])
Warning: Large-scale (trust region) method does not
currently solve this type of
problem,
  using medium-scale (line search) instead.
> In fmincon at 317
Optimization terminated: first-order optimality measure
less
  than options.TolFun and maximum constraint violation is
less
```

than options.TolCon.
No active inequalities.
x =
0.7500 1.2500
fval =
-3.1250

**GENETIC ALGORITHM (GA)**

Diagnostic information.
Fitness function = @ function 2
Number of variables = 2
1 Inequality constraints
0 Equality constraints
1 Total number of linear constraints
Modified options:
options.Display = 'diagnose'
options.OutputFcns = { { @gatooloutput } }
End of diagnostic information.

| Generation | f-count | Best f(x) | Mean f(x) | Stall Generations |
|---|---|---|---|---|
| 1 | 21 | -2.692 | -0.8409 | 0 |
| 2 | 41 | -2.692 | -0.8272 | 1 |
| 3 | 61 | -2.692 | -1.311 | 2 |
| 4 | 81 | -2.965 | -1.587 | 0 |
| 5 | 101 | -2.965 | -2.215 | 1 |
| 6 | 121 | -2.965 | -2.399 | 2 |
| 7 | 141 | -3.031 | -2.527 | 0 |
| 8 | 161 | -3.031 | -2.697 | 1 |
| 9 | 181 | -3.056 | -2.87 | 0 |
| 10 | 201 | -3.056 | -2.944 | 1 |
| 11 | 221 | -3.056 | -2.97 | 2 |
| 12 | 241 | -3.056 | -3.006 | 3 |
| 13 | 261 | -3.056 | -3.024 | 4 |
| 14 | 281 | -3.072 | -3.046 | 0 |
| 15 | 301 | -3.09 | -3.061 | 0 |
| 16 | 321 | -3.092 | -3.064 | 0 |
| 17 | 341 | -3.109 | -3.062 | 0 |
| 18 | 361 | -3.109 | -3.037 | 1 |
| 19 | 381 | -3.109 | -3.056 | 2 |
| 20 | 401 | -3.11 | -3.069 | 0 |
| 21 | 421 | -3.11 | -3.077 | 1 |
| 22 | 441 | -3.11 | -3.083 | 2 |
| 23 | 461 | -3.11 | -3.089 | 3 |
| 24 | 481 | -3.125 | -3.108 | 0 |
| 25 | 501 | -3.125 | -3.104 | 1 |
| 26 | 521 | -3.125 | -3.11 | 2 |
| 27 | 541 | -3.125 | -3.112 | 3 |
| 28 | 561 | -3.125 | -3.117 | 0 |
| 29 | 581 | -3.125 | -3.12 | 0 |
| 30 | 601 | -3.125 | -3.119 | 0 |

| Generation | f-count | Best f(x) | Mean f(x) | Stall Generations |
|---|---|---|---|---|
| 31 | 621 | -3.125 | -3.121 | 1 |
| 32 | 641 | -3.125 | -3.124 | 0 |
| 33 | 661 | -3.125 | -3.121 | 1 |
| 34 | 681 | -3.125 | -3.121 | 2 |
| 35 | 701 | -3.125 | -3.123 | 0 |
| 36 | 721 | -3.125 | -3.124 | 0 |
| 37 | 741 | -3.125 | -3.124 | 1 |
| 38 | 761 | -3.125 | -3.123 | 0 |
| 39 | 781 | -3.125 | -3.123 | 1 |
| 40 | 801 | -3.125 | -3.121 | 2 |
| 41 | 821 | -3.125 | -3.122 | 0 |
| 42 | 841 | -3.125 | -3.123 | 1 |
| 43 | 861 | -3.125 | -3.123 | 2 |
| 44 | 881 | -3.125 | -3.123 | 3 |
| 45 | 901 | -3.125 | -3.123 | 4 |
| 46 | 921 | -3.125 | -3.124 | 0 |
| 47 | 941 | -3.125 | -3.124 | 1 |
| 48 | 961 | -3.125 | -3.125 | 2 |
| 49 | 981 | -3.125 | -3.125 | 0 |
| 50 | 1001 | -3.125 | -3.125 | 0 |
| 51 | 1021 | -3.125 | -3.125 | 0 |

Optimization terminated: average change in the fitness value less than options.TolFun.
x =
0.754948 1.25042
fval =
-3.124948

**GENERALISED PATTERN SEARCH (GPS)**

Diagnostic information.
objective function = @ function 2
1 Inequality constraints
0 Equality constraints
1 Total number of linear constraints
X0 = [ 1 0 ]
Modified options:
options.PollMethod = 'GPSPositiveBasis2N'
options.CompletePoll = 'on'
options.SearchMethod = @GPSPositiveBasis2N
options.CompleteSearch = 'on'
options.Display = 'diagnose'
options.OutputFcns = { { @psearchtooloutput } }
End of diagnostic information.

| Iter | f-count | f(x) | MeshSize | Method |
|---|---|---|---|---|
| 0 | 1 | -1 | 1 | |
| 1 | 5 | -3 | 2 | Successful Poll |
| 2 | 8 | -3 | 1 | Refine Mesh |
| 3 | 13 | -3 | 0.5 | Refine Mesh |
| 4 | 18 | -3.125 | 1 | Successful Poll |
| 5 | 21 | -3.125 | 0.5 | Refine Mesh |
| 6 | 26 | -3.125 | 0.25 | Refine Mesh |
| 7 | 31 | -3.125 | 0.125 | Refine Mesh |
| 8 | 36 | -3.125 | 0.0625 | Refine Mesh |
| 9 | 41 | -3.125 | 0.03125 | Refine Mesh |
| 10 | 46 | -3.125 | 0.01563 | Refine Mesh |
| 11 | 51 | -3.125 | 0.007813 | Refine Mesh |
| 12 | 56 | -3.125 | 0.003906 | Refine Mesh |
| 13 | 61 | -3.125 | 0.001953 | Refine Mesh |
| 14 | 66 | -3.125 | 0.0009766 | Refine Mesh |
| 15 | 71 | -3.125 | 0.0004883 | Refine Mesh |
| 16 | 76 | -3.125 | 0.0002441 | Refine Mesh |
| 17 | 81 | -3.125 | 0.0001221 | Refine Mesh |

| | | | | |
|---|---|---|---|---|
| 18 | 86 | -3.125 | 6.104e-005 | Refine Mesh |
| 19 | 91 | -3.125 | 3.052e-005 | Refine Mesh |
| 20 | 96 | -3.125 | 1.526e-005 | Refine Mesh |
| 21 | 101 | -3.125 | 7.629e-006 | Refine Mesh |
| 22 | 106 | -3.125 | 3.815e-006 | Refine Mesh |
| 23 | 111 | -3.125 | 1.907e-006 | Refine Mesh |
| 24 | 116 | -3.125 | 9.537e-007 | Refine Mesh |

Optimization terminated: mesh size less than options.TolMesh.

>> x =

0.75 1.25

fval =

-3.125

**MESH ADAPTIVE SEARCH (MADS)**

Diagnostic information.

objective function = @ function 2

1 Inequality constraints

0 Equality constraints

1 Total number of linear constraints

X0 = [ 0.5 0.5 ]

Modified options:

options.PollMethod = 'MADSPositiveBasisNp1'

options.SearchMethod = @MADSPositiveBasisNp1

options.CompleteSearch = 'on'

options.Display = 'diagnose'

options.OutputFcns = { { @psearchtooloutput } }

End of diagnostic information.

| Iter | f-count | f(x) | MeshSize | Method |
|---|---|---|---|---|
| 0 | 1 | -2 | 1 | |
| 1 | 5 | -2.75 | 1 | Successful Poll |
| 2 | 11 | -3 | 1 | Successful Poll |
| 3 | 13 | -3 | 0.25 | Refine Mesh |
| 4 | 16 | -3 | 0.0625 | Refine Mesh |
| 5 | 20 | -3.07422 | 0.25 | Successful Poll |
| 6 | 21 | -3.07422 | 0.0625 | Refine Mesh |
| 7 | 22 | -3.07422 | 0.01563 | Refine Mesh |
| 8 | 25 | -3.08179 | 0.0625 | Successful Poll |
| 9 | 27 | -3.08179 | 0.01563 | Refine Mesh |
| 10 | 29 | -3.08179 | 0.003906 | Refine Mesh |
| 11 | 33 | -3.11709 | 0.01563 | Successful Poll |
| 12 | 35 | -3.11709 | 0.003906 | Refine Mesh |
| 13 | 36 | -3.11709 | 0.0009766 | Refine Mesh |
| 14 | 37 | -3.11709 | 0.0002441 | Refine Mesh |
| 15 | 39 | -3.11709 | 6.104e-005 | Refine Mesh |
| 16 | 41 | -3.11709 | 1.526e-005 | Refine Mesh |
| 17 | 45 | -3.11919 | 6.104e-005 | Successful Poll |
| 18 | 46 | -3.11919 | 1.526e-005 | Refine Mesh |
| 19 | 49 | -3.11935 | 6.104e-005 | Successful Poll |
| 20 | 54 | -3.12017 | 0.0002441 | Successful Poll |
| 21 | 55 | -3.12017 | 6.104e-005 | Refine Mesh |
| 22 | 57 | -3.12017 | 1.526e-005 | Refine Mesh |
| 23 | 60 | -3.12037 | 6.104e-005 | Successful Poll |
| 24 | 63 | -3.12037 | 1.526e-005 | Refine Mesh |
| 25 | 66 | -3.12037 | 3.815e-006 | Refine Mesh |
| 26 | 68 | -3.12037 | 9.537e-007 | Refine Mesh |
| 27 | 70 | -3.12037 | 2.384e-007 | Refine Mesh |
| 28 | 73 | -3.12037 | 5.96e-008 | Refine Mesh |
| 29 | 76 | -3.12037 | 1.49e-008 | Refine Mesh |
| 30 | 78 | -3.12037 | 3.725e-009 | Refine Mesh |
| Iter | f-count | f(x) | MeshSize | Method |
| 31 | 82 | -3.12038 | 1.49e-008 | Successful Poll |
| 32 | 88 | -3.12039 | 5.96e-008 | Successful Poll |
| 33 | 95 | -3.12039 | 2.384e-007 | Successful Poll |
| 34 | 100 | -3.12039 | 9.537e-007 | Successful Poll |
| 35 | 107 | -3.1204 | 3.815e-006 | Successful Poll |
| 36 | 112 | -3.1206 | 1.526e-005 | Successful Poll |
| 37 | 117 | -3.12061 | 6.104e-005 | Successful Poll |
| 38 | 122 | -3.12064 | 0.0002441 | Successful Poll |
| 39 | 129 | -3.12076 | 0.0009766 | Successful Poll |
| 40 | 132 | -3.12076 | 0.0002441 | Refine Mesh |
| 41 | 134 | -3.12076 | 6.104e-005 | Refine Mesh |
| 42 | 139 | -3.12125 | 0.0002441 | Successful Poll |
| 43 | 141 | -3.12125 | 6.104e-005 | Refine Mesh |
| 44 | 143 | -3.12125 | 1.526e-005 | Refine Mesh |
| 45 | 144 | -3.12125 | 3.815e-006 | Refine Mesh |
| 46 | 145 | -3.12125 | 9.537e-007 | Refine Mesh |
| 47 | 150 | -3.12181 | 3.815e-006 | Successful Poll |
| 48 | 155 | -3.12182 | 1.526e-005 | Successful Poll |
| 49 | 162 | -3.12182 | 6.104e-005 | Successful Poll |
| 50 | 169 | -3.12185 | 0.0002441 | Successful Poll |
| 51 | 176 | -3.12198 | 0.0009766 | Successful Poll |
| 52 | 178 | -3.12198 | 0.0002441 | Refine Mesh |
| 53 | 181 | -3.12198 | 6.104e-005 | Refine Mesh |
| 54 | 188 | -3.12201 | 0.0002441 | Successful Poll |
| 55 | 190 | -3.12201 | 6.104e-005 | Refine Mesh |
| 56 | 197 | -3.12204 | 0.0002441 | Successful Poll |
| 57 | 199 | -3.12204 | 6.104e-005 | Refine Mesh |
| 58 | 206 | -3.12207 | 0.0002441 | Successful Poll |
| 59 | 209 | -3.12207 | 6.104e-005 | Refine Mesh |
| 60 | 212 | -3.12207 | 1.526e-005 | Refine Mesh |
| Iter | f-count | f(x) | MeshSize | Method |
| 61 | 219 | -3.12207 | 6.104e-005 | Successful Poll |
| 62 | 222 | -3.12207 | 1.526e-005 | Refine Mesh |
| 63 | 225 | -3.12207 | 3.815e-006 | Refine Mesh |
| 64 | 232 | -3.12208 | 1.526e-005 | Successful Poll |
| 65 | 234 | -3.12208 | 3.815e-006 | Refine Mesh |
| 66 | 239 | -3.12212 | 1.526e-005 | Successful Poll |
| 67 | 245 | -3.12214 | 6.104e-005 | Successful Poll |
| 68 | 248 | -3.12258 | 0.0002441 | Successful Poll |
| 69 | 250 | -3.12258 | 6.104e-005 | Refine Mesh |
| 70 | 252 | -3.12258 | 1.526e-005 | Refine Mesh |
| 71 | 253 | -3.12258 | 3.815e-006 | Refine Mesh |
| 72 | 259 | -3.12346 | 1.526e-005 | Successful Poll |
| 73 | 264 | -3.12347 | 6.104e-005 | Successful Poll |
| 74 | 269 | -3.1235 | 0.0002441 | Successful Poll |
| 75 | 276 | -3.12362 | 0.0009766 | Successful Poll |
| 76 | 280 | -3.12435 | 0.003906 | Successful Poll |
| 77 | 282 | -3.12435 | 0.0009766 | Refine Mesh |
| 78 | 283 | -3.12435 | 0.0002441 | Refine Mesh |
| 79 | 285 | -3.12435 | 6.104e-005 | Refine Mesh |
| 80 | 287 | -3.12435 | 1.526e-005 | Refine Mesh |
| 81 | 288 | -3.12435 | 3.815e-006 | Refine Mesh |

| | | | | |
|---|---|---|---|---|
| 82 | 289 | -3.12435 | 9.537e-007 | Refine Mesh |
| 83 | 291 | -3.12435 | 2.384e-007 | Refine Mesh |
| 84 | 296 | -3.12468 | 9.537e-007 | Successful Poll |
| 85 | 303 | -3.12468 | 3.815e-006 | Successful Poll |
| 86 | 308 | -3.12468 | 1.526e-005 | Successful Poll |
| 87 | 313 | -3.12469 | 6.104e-005 | Successful Poll |
| 88 | 318 | -3.12472 | 0.0002441 | Successful Poll |
| 89 | 323 | -3.12484 | 0.0009766 | Successful Poll |
| 90 | 326 | -3.12484 | 0.0002441 | Refine Mesh |
| Iter | f-count | f(x) | MeshSize | Method |
| 91 | 329 | -3.12484 | 6.104e-005 | Refine Mesh |
| 92 | 336 | -3.12487 | 0.0002441 | Successful Poll |
| 93 | 338 | -3.12487 | 6.104e-005 | Refine Mesh |
| 94 | 341 | -3.12487 | 1.526e-005 | Refine Mesh |
| 95 | 348 | -3.12488 | 6.104e-005 | Successful Poll |
| 96 | 350 | -3.12488 | 1.526e-005 | Refine Mesh |
| 97 | 357 | -3.12489 | 6.104e-005 | Successful Poll |
| 98 | 359 | -3.12489 | 1.526e-005 | Refine Mesh |
| 99 | 366 | -3.12489 | 6.104e-005 | Successful Poll |
| 100 | 369 | -3.12489 | 1.526e-005 | Refine Mesh |
| 101 | 372 | -3.12489 | 3.815e-006 | Refine Mesh |
| 102 | 374 | -3.12489 | 9.537e-007 | Refine Mesh |
| 103 | 381 | -3.1249 | 3.815e-006 | Successful Poll |
| 104 | 384 | -3.1249 | 9.537e-007 | Refine Mesh |
| 105 | 391 | -3.1249 | 3.815e-006 | Successful Poll |
| 106 | 394 | -3.1249 | 9.537e-007 | Refine Mesh |
| 107 | 401 | -3.1249 | 3.815e-006 | Successful Poll |
| 108 | 403 | -3.1249 | 9.537e-007 | Refine Mesh |
| 109 | 406 | -3.1249 | 2.384e-007 | Refine Mesh |
| 110 | 413 | -3.1249 | 9.537e-007 | Successful Poll |
| 111 | 416 | -3.1249 | 2.384e-007 | Refine Mesh |
| 112 | 419 | -3.1249 | 5.96e-008 | Refine Mesh |
| 113 | 424 | -3.1249 | 2.384e-007 | Successful Poll |
| 114 | 427 | -3.1249 | 5.96e-008 | Refine Mesh |
| 115 | 432 | -3.1249 | 2.384e-007 | Successful Poll |
| 116 | 434 | -3.1249 | 5.96e-008 | Refine Mesh |
| 117 | 439 | -3.1249 | 2.384e-007 | Successful Poll |
| 118 | 442 | -3.1249 | 5.96e-008 | Refine Mesh |
| 119 | 444 | -3.1249 | 1.49e-008 | Refine Mesh |
| 120 | 449 | -3.1249 | 5.96e-008 | Successful Poll |
| Iter | f-count | f(x) | MeshSize | Method |
| 121 | 452 | -3.1249 | 1.49e-008 | Refine Mesh |
| 122 | 459 | -3.1249 | 5.96e-008 | Successful Poll |
| 123 | 462 | -3.1249 | 1.49e-008 | Refine Mesh |
| 124 | 464 | -3.1249 | 3.725e-009 | Refine Mesh |
| 125 | 471 | -3.1249 | 1.49e-008 | Successful Poll |
| 126 | 474 | -3.1249 | 3.725e-009 | Refine Mesh |
| 127 | 481 | -3.1249 | 1.49e-008 | Successful Poll |
| 128 | 483 | -3.1249 | 3.725e-009 | Refine Mesh |
| 129 | 485 | -3.1249 | 9.313e-010 | Refine Mesh |
| 130 | 487 | -3.1249 | 2.328e-010 | Refine Mesh |
| 131 | 490 | -3.1249 | 5.821e-011 | Refine Mesh |
| 132 | 493 | -3.1249 | 1.455e-011 | Refine Mesh |
| 133 | 496 | -3.1249 | 3.638e-012 | Refine Mesh |
| 134 | 499 | -3.1249 | 9.095e-013 | Refine Mesh |
| 135 | 502 | -3.1249 | 2.274e-013 | Refine Mesh |

Optimization terminated: mesh size less than 'numberOfVariables*sqrt(TolMesh)'.>>
x =
 0.7572 1.242
fval =
 -3.124896

# Appendix C

Maximization of $f_3(x_1,x_2)=5x_1-x_1^2+8x_2-2x_2^2$
subject to
$\widetilde{X}_2=\{3x_1+3x_2\le 6\,;x_1,x_2\ge 0\}$

**QUASI-NEWTON METHOD (QNM)**
Diagnostic Information
Number of variables: 2
Functions
 Objective: function 3
 Gradient: finite-differencing
 Hessian: finite-differencing (or Quasi-Newton)
Constraints
 Nonlinear constraints: do not exist
 Number of linear inequality constraints: 1
 Number of linear equality constraints: 0
 Number of lower bound constraints: 0
 Number of upper bound constraints: 0
Algorithm selected
 medium-scale

End diagnostic information

| Iter | F-count | f(x) | Max constraint | Line search steplength | Directional derivative | First-order optimality | Procedure |
|---|---|---|---|---|---|---|---|
| 0 | 3 | -10 | -1 | | | | |
| 1 | 7 | -10.7352 | -0.5 | 0.5 | 1.14 | 1.54 | |
| 2 | 10 | -11.4998 | 0 | 1 | -0.484 | 0.224 | |
| 3 | 13 | -11.5 | 0 | 1 | 2.1e-005 | 0.0043 | |
| 4 | 16 | -11.5 | 0 | 1 | 7.22e-011 | 1.18e-007 | Hessian modified |

Optimization terminated: first-order optimality measure less
 than options.TolFun and maximum constraint violation is less
 than options.TolCon.
Active inequalities (to within options.TolCon = 1e-006):
 lower upper ineqlin ineqnonlin
 1
>> [x,fval]=fmincon('p',[0,0],[],[],[3,2],6,[0,0],[inf,inf])
Warning: Large-scale (trust region) method does not currently solve this type of problem,
 using medium-scale (line search) instead.

> In fmincon at 317
Optimization terminated: first-order optimality measure less
than options.TolFun and maximum constraint violation is less
than options.TolCon.
No active inequalities.
x =
1.0000 1.5000
fval =
-11.4999999
>>

**GENETIC ALGORITHM (GA)**

Diagnostic information.
Fitness function = @ function 3
Number of variables = 2
1 Inequality constraints
0 Equality constraints
1 Total number of linear constraints
Modified options:
options.Display = 'diagnose'
options.OutputFcns = { { @gatooloutput } }
End of diagnostic information.

| | | Best | Mean | Stall |
|---|---|---|---|---|
| Generation | f-count | f(x) | f(x) | Generations |
| 1 | 21 | -5.729 | -0.1121 | 0 |
| 2 | 41 | -5.729 | -1.021 | 1 |
| 3 | 61 | -8.877 | -2.789 | 0 |
| 4 | 81 | -8.877 | -4.598 | 1 |
| 5 | 101 | -10.76 | -5.827 | 0 |
| 6 | 121 | -10.76 | -6.866 | 1 |
| 7 | 141 | -11.27 | -8.525 | 0 |
| 8 | 161 | -11.27 | -8.673 | 1 |
| 9 | 181 | -11.27 | -9.114 | 2 |
| 10 | 201 | -11.27 | -10.29 | 3 |
| 11 | 221 | -11.44 | -10.69 | 0 |
| 12 | 241 | -11.44 | -10.88 | 1 |
| 13 | 261 | -11.44 | -11.06 | 2 |
| 14 | 281 | -11.44 | -11.19 | 3 |
| 15 | 301 | -11.44 | -11.22 | 4 |
| 16 | 321 | -11.47 | -11.29 | 0 |
| 17 | 341 | -11.47 | -11.35 | 1 |
| 18 | 361 | -11.47 | -11.42 | 2 |
| 19 | 381 | -11.5 | -11.46 | 0 |
| 20 | 401 | -11.5 | -11.47 | 1 |
| 21 | 421 | -11.5 | -11.49 | 2 |
| 22 | 441 | -11.5 | -11.49 | 3 |
| 23 | 461 | -11.5 | -11.49 | 4 |
| 24 | 481 | -11.5 | -11.5 | 0 |
| 25 | 501 | -11.5 | -11.5 | 1 |
| 26 | 521 | -11.5 | -11.5 | 2 |
| 27 | 541 | -11.5 | -11.5 | 3 |
| 28 | 561 | -11.5 | -11.5 | 4 |
| 29 | 581 | -11.5 | -11.5 | 0 |
| 30 | 601 | -11.5 | -11.5 | 0 |

| | | Best | Mean | Stall |
|---|---|---|---|---|
| Generation | f-count | f(x) | f(x) | Generations |
| 31 | 621 | -11.5 | -11.5 | 1 |
| 32 | 641 | -11.5 | -11.5 | 0 |
| 33 | 661 | -11.5 | -11.5 | 1 |
| 34 | 681 | -11.5 | -11.5 | 0 |
| 35 | 701 | -11.5 | -11.5 | 0 |
| 36 | 721 | -11.5 | -11.5 | 0 |
| 37 | 741 | -11.5 | -11.5 | 0 |
| 38 | 761 | -11.5 | -11.5 | 0 |
| 39 | 781 | -11.5 | -11.5 | 0 |
| 40 | 801 | -11.5 | -11.49 | 1 |
| 41 | 821 | -11.5 | -11.49 | 2 |
| 42 | 841 | -11.5 | -11.49 | 3 |
| 43 | 861 | -11.5 | -11.49 | 4 |
| 44 | 881 | -11.5 | -11.5 | 5 |
| 45 | 901 | -11.5 | -11.5 | 6 |
| 46 | 921 | -11.5 | -11.5 | 0 |
| 47 | 941 | -11.5 | -11.5 | 1 |
| 48 | 961 | -11.5 | -11.5 | 2 |
| 49 | 981 | -11.5 | -11.5 | 3 |
| 50 | 1001 | -11.5 | -11.5 | 4 |
| 51 | 1021 | -11.5 | -11.5 | 5 |

Optimization terminated: average change in the fitness value less than options.TolFun.
>> x =
0.9969 1.50457
fval =
-11.4999999
>>

**GENERALISED PATTERN SEARCH (GPS)**

Diagnostic information.
objective function = @ function 3
1 Inequality constraints
0 Equality constraints
1 Total number of linear constraints
X0 = [ 0 1 ]
Modified options:
options.PollMethod = 'GPSPositiveBasis2N'
options.CompletePoll = 'on'
options.SearchMethod = @GPSPositiveBasis2N
options.CompleteSearch = 'on'
options.Display = 'diagnose'
options.OutputFcns = { { @psearchtooloutput } }
End of diagnostic information.

| Iter | f-count | f(x) | MeshSize | Method |
|---|---|---|---|---|
| 0 | 1 | -6 | 1 | |
| 1 | 5 | -10 | 2 | Successful Poll |
| 2 | 7 | -10 | 1 | Refine Mesh |
| 3 | 9 | -10 | 0.5 | Refine Mesh |
| 4 | 12 | -11.5 | 1 | Successful Poll |
| 5 | 15 | -11.5 | 0.5 | Refine Mesh |
| 6 | 20 | -11.5 | 0.25 | Refine Mesh |
| 7 | 25 | -11.5 | 0.125 | Refine Mesh |
| 8 | 30 | -11.5 | 0.0625 | Refine Mesh |

```
  9    35    -11.5   0.03125     Refine Mesh
 10    40    -11.5   0.01563     Refine Mesh
 11    45    -11.5   0.007813    Refine Mesh
 12    50    -11.5   0.003906    Refine Mesh
 13    55    -11.5   0.001953    Refine Mesh
 14    60    -11.5   0.0009766   Refine Mesh
 15    65    -11.5   0.0004883   Refine Mesh
 16    70    -11.5   0.0002441   Refine Mesh
 17    75    -11.5   0.0001221   Refine Mesh
 18    80    -11.5   6.104e-005  Refine Mesh
 19    85    -11.5   3.052e-005  Refine Mesh
 20    90    -11.5   1.526e-005  Refine Mesh
 21    95    -11.5   7.629e-006  Refine Mesh
 22   100    -11.5   3.815e-006  Refine Mesh
 23   105    -11.5   1.907e-006  Refine Mesh
 24   110    -11.5   9.537e-007  Refine Mesh
Optimization terminated: mesh size less than
options.TolMesh.
>> x =
   1   1.5
fval =
  -11.5
>>
```

**MESH ADAPTIVE SEARCH (MADS)**

```
Diagnostic information.
  objective function = @ function 3
  1 Inequality constraints
  0 Equality constraints
  1 Total number of linear constraints
  X0 = [ 1 0 ]
Modified options:
  options.PollMethod = 'MADSPositiveBasis2N'
  options.CompletePoll = 'on'
  options.SearchMethod = @MADSPositiveBasis2N
  options.CompleteSearch = 'on'
  options.Display = 'diagnose'
  options.OutputFcns = { { @psearchtooloutput } }
End of diagnostic information.
Iter   f-count    f(x)    MeshSize    Method
  0     1          -4      1
  1     9         -10      1          Successful Poll
  2    12         -10      0.25       Refine Mesh
  3    18       -11.5      1          Successful Poll
  4    21       -11.5      0.25       Refine Mesh
  5    24       -11.5      0.0625     Refine Mesh
  6    27       -11.5      0.01563    Refine Mesh
  7    30       -11.5      0.003906   Refine Mesh
  8    33       -11.5      0.0009766  Refine Mesh
  9    36       -11.5      0.0002441  Refine Mesh
 10    39       -11.5      6.104e-005 Refine Mesh
 11    42       -11.5      1.526e-005 Refine Mesh
 12    45       -11.5      3.815e-006 Refine Mesh
 13    48       -11.5      9.537e-007 Refine Mesh
 14    51       -11.5      2.384e-007 Refine Mesh
 15    54       -11.5      5.96e-008  Refine Mesh
 16    57       -11.5      1.49e-008  Refine Mesh
 17    60       -11.5      3.725e-009 Refine Mesh
 18    63       -11.5      9.313e-010 Refine Mesh
 19    66       -11.5      2.328e-010 Refine Mesh
 20    69       -11.5      5.821e-011 Refine Mesh
 21    72       -11.5      1.455e-011 Refine Mesh
 22    75       -11.5      3.638e-012 Refine Mesh
 23    78       -11.5      9.095e-013 Refine Mesh
Optimization terminated: mesh size less than
'sqrt(TolMesh)'.
>> x =
   1   1.5
fval =
  -11.5
>>
```

# Appendix D

Maximization of $f_4(x_1, x_2) = x_1 + 2x_2 - x_2^3$
subject to
$\widetilde{X}_3 = \{x_1 + x_2 \le 1 \; ; x_1, x_2 \ge 0\}$

**QUASI-NEWTON METHOD (QNM)**

```
  Diagnostic Information
Number of variables: 2
Functions
 Objective:                  function
 Gradient:                   finite-differencing
 Hessian:                    finite-differencing (or
Quasi-Newton)
Constraints
 Nonlinear constraints:            do not exist
 Number of linear inequality constraints:   1
 Number of linear equality constraints:     0
 Number of lower bound constraints:         0
 Number of upper bound constraints:         0
Algorithm selected
  medium-scale

End diagnostic information
                          Max     Line search Directional
First-order
 Iter F-count     f(x)    constraint   steplength   derivative
optimality Procedure
  0     3        0          -1
  1     6       -1           0         1         1        2
  2     9    -1.375    -1.11e-016         1      0.125
0.75
  3    12    -1.38409        0          1     -0.00412
0.0741
  4    15    -1.3849         0          1      0.00013
0.00554
  5    18    -1.3849   -1.11e-016       1   1.73e-007
0.000106
Optimization terminated: magnitude of directional
derivative in search
```

```
  direction less than 2*options.TolFun and maximum
constraint violation
  is less than options.TolCon.
Active inequalities (to within options.TolCon = 1e-006):
  lower      upper      ineqlin    ineqnonlin
                           1
>> [x,fval]=fmincon('p',[1,0],[],[],[1,1],1,[0,0],[inf,inf])
Warning: Large-scale (trust region) method does not
currently solve this type of
problem,
  using medium-scale (line search) instead.
> In fmincon at 317
Optimization terminated: magnitude of directional
derivative in search
  direction less than 2*options.TolFun and maximum
constraint violation
  is less than options.TolCon.
No active inequalities.
x =
    0.4226    0.5774
fval =
    -1.3849
>>
```

**GENETIC ALGORITHM (GA)**

```
Diagnostic information.
  Fitness function = @ function 4
  Number of variables = 2
  1 Inequality constraints
  0 Equality constraints
  1 Total number of linear constraints
Modified options:
  options.Display = 'diagnose'
  options.OutputFcns = { { @gatooloutput } }
End of diagnostic information.
```

| Generation | f-count | Best f(x) | Mean f(x) | Stall Generations |
|---|---|---|---|---|
| 1 | 21 | -1.085 | -0.7051 | 0 |
| 2 | 41 | -1.361 | -0.8173 | 0 |
| 3 | 61 | -1.361 | -0.9189 | 1 |
| 4 | 81 | -1.364 | -0.8839 | 0 |
| 5 | 101 | -1.364 | -0.7675 | 1 |
| 6 | 121 | -1.38 | -0.9082 | 0 |
| 7 | 141 | -1.385 | -0.9588 | 0 |
| 8 | 161 | -1.385 | -0.9239 | 1 |
| 9 | 181 | -1.385 | -1.026 | 2 |
| 10 | 201 | -1.385 | -1.15 | 3 |
| 11 | 221 | -1.385 | -1.178 | 4 |
| 12 | 241 | -1.385 | -1.249 | 5 |
| 13 | 261 | -1.385 | -1.293 | 6 |
| 14 | 281 | -1.385 | -1.305 | 7 |
| 15 | 301 | -1.385 | -1.353 | 8 |
| 16 | 321 | -1.385 | -1.367 | 9 |
| 17 | 341 | -1.385 | -1.377 | 0 |
| 18 | 361 | -1.385 | -1.38 | 0 |
| 19 | 381 | -1.385 | -1.378 | 1 |
| 20 | 401 | -1.385 | -1.381 | 2 |
| 21 | 421 | -1.385 | -1.382 | 3 |
| 22 | 441 | -1.385 | -1.384 | 0 |
| 23 | 461 | -1.385 | -1.383 | 1 |
| 24 | 481 | -1.385 | -1.385 | 0 |
| 25 | 501 | -1.385 | -1.385 | 1 |
| 26 | 521 | -1.385 | -1.385 | 2 |
| 27 | 541 | -1.385 | -1.385 | 0 |
| 28 | 561 | -1.385 | -1.385 | 1 |
| 29 | 581 | -1.385 | -1.385 | 2 |
| 30 | 601 | -1.385 | -1.385 | 0 |

| Generation | f-count | Best f(x) | Mean f(x) | Stall Generations |
|---|---|---|---|---|
| 31 | 621 | -1.385 | -1.385 | 1 |
| 32 | 641 | -1.385 | -1.385 | 2 |
| 33 | 661 | -1.385 | -1.385 | 0 |
| 34 | 681 | -1.385 | -1.385 | 0 |
| 35 | 701 | -1.385 | -1.385 | 0 |
| 36 | 721 | -1.385 | -1.385 | 1 |
| 37 | 741 | -1.385 | -1.385 | 0 |
| 38 | 761 | -1.385 | -1.385 | 1 |
| 39 | 781 | -1.385 | -1.385 | 0 |
| 40 | 801 | -1.385 | -1.385 | 1 |
| 41 | 821 | -1.385 | -1.385 | 2 |
| 42 | 841 | -1.385 | -1.385 | 3 |
| 43 | 861 | -1.385 | -1.385 | 0 |
| 44 | 881 | -1.385 | -1.385 | 1 |
| 45 | 901 | -1.385 | -1.385 | 0 |
| 46 | 921 | -1.385 | -1.385 | 1 |
| 47 | 941 | -1.385 | -1.385 | 2 |
| 48 | 961 | -1.385 | -1.385 | 3 |
| 49 | 981 | -1.385 | -1.385 | 4 |
| 50 | 1001 | -1.385 | -1.385 | 0 |
| 51 | 1021 | -1.385 | -1.385 | 1 |

```
Optimization terminated: average change in the fitness
value less than options.TolFun.
>> x =
    0.42222    0.57778
fval =
    -1.384899
>>
```

**GENERALISED PATTERN SEARCH (GPS)**

```
Diagnostic information.
  objective function = @ function 4
  1 Inequality constraints
  0 Equality constraints
  1 Total number of linear constraints
  X0 = [ 1 0 ]
Modified options:
  options.PollMethod = 'GPSPositiveBasis2N'
  options.CompletePoll = 'on'
  options.SearchMethod = @GPSPositiveBasis2N
  options.CompleteSearch = 'on'
  options.Display = 'diagnose'
  options.OutputFcns = { { @psearchtooloutput } }
End of diagnostic information.
```

| Iter | f-count | f(x) | MeshSize | Method |
|---|---|---|---|---|
| 0 | 1 | -1 | 1 | |
| 1 | 4 | -1.375 | 2 | Successful Poll |
| 2 | 7 | -1.375 | 1 | Refine Mesh |
| 3 | 12 | -1.375 | 0.5 | Refine Mesh |
| 4 | 17 | -1.375 | 0.25 | Refine Mesh |
| 5 | 22 | -1.38086 | 0.5 | Successful Poll |
| 6 | 27 | -1.38086 | 0.25 | Refine Mesh |
| 7 | 32 | -1.38086 | 0.125 | Refine Mesh |
| 8 | 37 | -1.38452 | 0.25 | Successful Poll |
| 9 | 42 | -1.38452 | 0.125 | Refine Mesh |
| 10 | 47 | -1.38452 | 0.0625 | Refine Mesh |
| 11 | 52 | -1.38452 | 0.03125 | Refine Mesh |
| 12 | 57 | -1.3849 | 0.0625 | Successful Poll |
| 13 | 62 | -1.3849 | 0.03125 | Refine Mesh |
| 14 | 67 | -1.3849 | 0.01563 | Refine Mesh |
| 15 | 72 | -1.3849 | 0.007813 | Refine Mesh |
| 16 | 77 | -1.3849 | 0.003906 | Refine Mesh |
| 17 | 82 | -1.3849 | 0.001953 | Refine Mesh |
| 18 | 87 | -1.3849 | 0.003906 | Successful Poll |
| 19 | 92 | -1.3849 | 0.001953 | Refine Mesh |
| 20 | 97 | -1.3849 | 0.0009766 | Refine Mesh |
| 21 | 102 | -1.3849 | 0.0004883 | Refine Mesh |
| 22 | 107 | -1.3849 | 0.0009766 | Successful Poll |
| 23 | 112 | -1.3849 | 0.0004883 | Refine Mesh |
| 24 | 117 | -1.3849 | 0.0002441 | Refine Mesh |
| 25 | 122 | -1.3849 | 0.0001221 | Refine Mesh |
| 26 | 127 | -1.3849 | 0.0002441 | Successful Poll |
| 27 | 132 | -1.3849 | 0.0001221 | Refine Mesh |
| 28 | 137 | -1.3849 | 6.104e-005 | Refine Mesh |
| 29 | 142 | -1.3849 | 0.0001221 | Successful Poll |
| 30 | 147 | -1.3849 | 6.104e-005 | Refine Mesh |
| Iter | f-count | f(x) | MeshSize | Method |
| 31 | 152 | -1.3849 | 3.052e-005 | Refine Mesh |
| 32 | 157 | -1.3849 | 6.104e-005 | Successful Poll |
| 33 | 162 | -1.3849 | 3.052e-005 | Refine Mesh |
| 34 | 167 | -1.3849 | 1.526e-005 | Refine Mesh |
| 35 | 172 | -1.3849 | 7.629e-006 | Refine Mesh |
| 36 | 177 | -1.3849 | 1.526e-005 | Successful Poll |
| 37 | 182 | -1.3849 | 7.629e-006 | Refine Mesh |
| 38 | 187 | -1.3849 | 3.815e-006 | Refine Mesh |
| 39 | 192 | -1.3849 | 1.907e-006 | Refine Mesh |
| 40 | 197 | -1.3849 | 9.537e-007 | Refine Mesh |

Optimization terminated: mesh size less than options.TolMesh.

```
>> x =
   0.42265    0.57735
fval =
   -1.384900
>>
```

**MESH ADAPTIVE SEARCH (MADS)**

```
Diagnostic information.
  objective function = @ function 4
  1 Inequality constraints
  0 Equality constraints
  1 Total number of linear constraints
  X0 = [ 0.5 0.5 ]
Modified options:
  options.PollMethod = 'MADSPositiveBasisNp1'
  options.SearchMethod = @MADSPositiveBasisNp1
  options.CompleteSearch = 'on'
  options.Display = 'diagnose'
  options.OutputFcns = { { @psearchtooloutput } }
End of diagnostic information.
```

| Iter | f-count | f(x) | MeshSize | Method |
|---|---|---|---|---|
| 0 | 1 | -1.375 | 1 | |
| 1 | 4 | -1.375 | 0.25 | Refine Mesh |
| 2 | 6 | -1.375 | 0.0625 | Refine Mesh |
| 3 | 9 | -1.375 | 0.01563 | Refine Mesh |
| 4 | 12 | -1.375 | 0.003906 | Refine Mesh |
| 5 | 15 | -1.375 | 0.0009766 | Refine Mesh |
| 6 | 17 | -1.375 | 0.0002441 | Refine Mesh |
| 7 | 20 | -1.375 | 6.104e-005 | Refine Mesh |
| 8 | 23 | -1.375 | 1.526e-005 | Refine Mesh |
| 9 | 26 | -1.375 | 3.815e-006 | Refine Mesh |
| 10 | 28 | -1.375 | 9.537e-007 | Refine Mesh |
| 11 | 31 | -1.375 | 2.384e-007 | Refine Mesh |
| 12 | 33 | -1.375 | 5.96e-008 | Refine Mesh |
| 13 | 35 | -1.375 | 1.49e-008 | Refine Mesh |
| 14 | 38 | -1.375 | 3.725e-009 | Refine Mesh |
| 15 | 41 | -1.375 | 9.313e-010 | Refine Mesh |
| 16 | 44 | -1.375 | 2.328e-010 | Refine Mesh |
| 17 | 47 | -1.375 | 5.821e-011 | Refine Mesh |
| 18 | 51 | -1.375 | 2.328e-010 | Successful Poll |
| 19 | 58 | -1.375 | 9.313e-010 | Successful Poll |
| 20 | 65 | -1.375 | 3.725e-009 | Successful Poll |
| 21 | 71 | -1.37501 | 1.49e-008 | Successful Poll |
| 22 | 78 | -1.37501 | 5.96e-008 | Successful Poll |
| 23 | 85 | -1.37501 | 2.384e-007 | Successful Poll |
| 24 | 92 | -1.37501 | 9.537e-007 | Successful Poll |
| 25 | 97 | -1.37513 | 3.815e-006 | Successful Poll |
| 26 | 104 | -1.37514 | 1.526e-005 | Successful Poll |
| 27 | 111 | -1.37516 | 6.104e-005 | Successful Poll |
| 28 | 118 | -1.37522 | 0.0002441 | Successful Poll |
| 29 | 121 | -1.37522 | 6.104e-005 | Refine Mesh |
| 30 | 123 | -1.37522 | 1.526e-005 | Refine Mesh |

| Iter | f-count | f(x) | MeshSize | Method |
|---|---|---|---|---|
| 31 | 130 | -1.37524 | 6.104e-005 | Successful Poll |
| 32 | 132 | -1.37524 | 1.526e-005 | Refine Mesh |
| 33 | 135 | -1.37524 | 3.815e-006 | Refine Mesh |
| 34 | 142 | -1.37525 | 1.526e-005 | Successful Poll |
| 35 | 145 | -1.37525 | 3.815e-006 | Refine Mesh |
| 36 | 148 | -1.37525 | 9.537e-007 | Refine Mesh |
| 37 | 155 | -1.37525 | 3.815e-006 | Successful Poll |
| 38 | 158 | -1.37525 | 9.537e-007 | Refine Mesh |
| 39 | 161 | -1.37525 | 2.384e-007 | Refine Mesh |
| 40 | 164 | -1.37525 | 5.96e-008 | Refine Mesh |
| 41 | 171 | -1.37525 | 2.384e-007 | Successful Poll |
| 42 | 174 | -1.37525 | 5.96e-008 | Refine Mesh |
| 43 | 179 | -1.37525 | 2.384e-007 | Successful Poll |
| 44 | 182 | -1.37525 | 5.96e-008 | Refine Mesh |

| | | | | |
|---|---|---|---|---|
| 45 | 189 | -1.37525 | 2.384e-007 | Successful Poll |
| 46 | 193 | -1.37531 | 9.537e-007 | Successful Poll |
| 47 | 200 | -1.37531 | 3.815e-006 | Successful Poll |
| 48 | 207 | -1.37532 | 1.526e-005 | Successful Poll |
| 49 | 214 | -1.37534 | 6.104e-005 | Successful Poll |
| 50 | 219 | -1.37672 | 0.0002441 | Successful Poll |
| 51 | 226 | -1.37699 | 0.0009766 | Successful Poll |
| 52 | 229 | -1.37699 | 0.0002441 | Refine Mesh |
| 53 | 232 | -1.37699 | 6.104e-005 | Refine Mesh |
| 54 | 237 | -1.37706 | 0.0002441 | Successful Poll |
| 55 | 239 | -1.37706 | 6.104e-005 | Refine Mesh |
| 56 | 244 | -1.37713 | 0.0002441 | Successful Poll |
| 57 | 248 | -1.37807 | 0.0009766 | Successful Poll |
| 58 | 250 | -1.37807 | 0.0002441 | Refine Mesh |
| 59 | 253 | -1.37933 | 0.0009766 | Successful Poll |
| 60 | 255 | -1.37933 | 0.0002441 | Refine Mesh |
| Iter | f-count | f(x) | MeshSize | Method |
| 61 | 257 | -1.37933 | 6.104e-005 | Refine Mesh |
| 62 | 258 | -1.37933 | 1.526e-005 | Refine Mesh |
| 63 | 262 | -1.38128 | 6.104e-005 | Successful Poll |
| 64 | 269 | -1.38134 | 0.0002441 | Successful Poll |
| 65 | 272 | -1.38134 | 6.104e-005 | Refine Mesh |
| 66 | 279 | -1.38141 | 0.0002441 | Successful Poll |
| 67 | 282 | -1.38141 | 6.104e-005 | Refine Mesh |
| 68 | 289 | -1.38148 | 0.0002441 | Successful Poll |
| 69 | 291 | -1.38148 | 6.104e-005 | Refine Mesh |
| 70 | 298 | -1.38154 | 0.0002441 | Successful Poll |
| 71 | 300 | -1.38154 | 6.104e-005 | Refine Mesh |
| 72 | 303 | -1.38154 | 1.526e-005 | Refine Mesh |
| 73 | 308 | -1.38156 | 6.104e-005 | Successful Poll |
| 74 | 312 | -1.3822 | 0.0002441 | Successful Poll |
| 75 | 319 | -1.38245 | 0.0009766 | Successful Poll |
| 76 | 323 | -1.38272 | 0.003906 | Successful Poll |
| 77 | 325 | -1.38272 | 0.0009766 | Refine Mesh |
| 78 | 326 | -1.38272 | 0.0002441 | Refine Mesh |
| 79 | 327 | -1.38272 | 6.104e-005 | Refine Mesh |
| 80 | 329 | -1.38272 | 1.526e-005 | Refine Mesh |
| 81 | 334 | -1.38467 | 6.104e-005 | Successful Poll |
| 82 | 341 | -1.38473 | 0.0002441 | Successful Poll |
| 83 | 344 | -1.38473 | 6.104e-005 | Refine Mesh |
| 84 | 351 | -1.3848 | 0.0002441 | Successful Poll |
| 85 | 354 | -1.3848 | 6.104e-005 | Refine Mesh |
| 86 | 357 | -1.3848 | 1.526e-005 | Refine Mesh |
| 87 | 362 | -1.38481 | 6.104e-005 | Successful Poll |
| 88 | 365 | -1.38481 | 1.526e-005 | Refine Mesh |
| 89 | 368 | -1.38481 | 3.815e-006 | Refine Mesh |
| 90 | 371 | -1.38481 | 9.537e-007 | Refine Mesh |
| Iter | f-count | f(x) | MeshSize | Method |
| 91 | 378 | -1.38481 | 3.815e-006 | Successful Poll |
| 92 | 381 | -1.38481 | 9.537e-007 | Refine Mesh |
| 93 | 388 | -1.38481 | 3.815e-006 | Successful Poll |
| 94 | 391 | -1.38481 | 9.537e-007 | Refine Mesh |
| 95 | 398 | -1.38482 | 3.815e-006 | Successful Poll |
| 96 | 401 | -1.38482 | 9.537e-007 | Refine Mesh |
| 97 | 404 | -1.38482 | 2.384e-007 | Refine Mesh |
| 98 | 407 | -1.38482 | 5.96e-008 | Refine Mesh |
| 99 | 409 | -1.38482 | 1.49e-008 | Refine Mesh |
| 100 | 412 | -1.38482 | 3.725e-009 | Refine Mesh |
| 101 | 415 | -1.38482 | 9.313e-010 | Refine Mesh |
| 102 | 418 | -1.38482 | 2.328e-010 | Refine Mesh |
| 103 | 421 | -1.38482 | 5.821e-011 | Refine Mesh |
| 104 | 426 | -1.38482 | 2.328e-010 | Successful Poll |
| 105 | 429 | -1.38482 | 5.821e-011 | Refine Mesh |
| 106 | 432 | -1.38482 | 1.455e-011 | Refine Mesh |
| 107 | 435 | -1.38482 | 3.638e-012 | Refine Mesh |
| 108 | 438 | -1.38482 | 9.095e-013 | Refine Mesh |
| 109 | 441 | -1.38482 | 2.274e-013 | Refine Mesh |

```
Optimization terminated: mesh size less than
'numberOfVariables*sqrt(TolMesh)'.
>> x =
   0.4296    0.5703
fval =
   -1.384815
>>
```

# REFERENCES


[1] Abramson, M. A., Audet, C., Chrissis, J.W. & Walston, J.G. (2009). Mesh adaptive direct search algorithms for mixed variable optimization. Optimization Letters, 3 (1), 35-47.

[2] Audet, C. & Dennis Jr, J. E., (2006). Mesh adaptive direct search algorithms for constrained optimization. *SIAM Journal on Optimization,* 17, 188-217.

[3] Audet, C. & Dennis Jr., J. E. (2009). A progressive barrier for derivative free non linear programming. *SIAM Journal on Optimization*, 20 (1), 445-472.

[4] Audet, C., Lanni, A., Le Digabel, S. & Tribes, C. (2014). Reducing the number of function evaluations in mesh adaptive direct search algorithm. *SIAM Journal on Optimization*, 24 (2), 621-642.

[5] Audet, C., Savard, G. & Zghal, W. (2010). A mesh adaptive direct search algorithm for multi-objective optimization. *European Journal of Operational Research*, 204 (3), 545-556.

[6] Davidon, W. C. (1959). Variable metric method for minimization. *Research and Development Report* ANL-5990 (Ref.) U.S. Atomic Energy Commission, Argonne National Laboratories.

[7] Dolan, E.D., Lewis, R.M., & Torczon, V. (2003). On the local convergence of pattern search. *SIAM Journal on Optimization*, 14 (2), 567-583.

[8] Fletcher, R & Powell, M.J.D (1963). A rapidly convergent descent method for minimization. *Comput. J.* 6, 163-168.

[9] Ghadle, K. P. & Pawar, T. S (2015). New approach for Wolfe's modified simplex method to solve quadratic programming problems. International Journal of Research in.

[10] Engineering and Technology, 4 (1): 371-376. eISSN: 2319-1163| pISSN:2321-7308. http://www.ijret.org.

[11] Goldberg, D. E. (1989). Genetic algorithms in search.

*Optimization & Machine Learning*. Reading, MA: Addison-Wesley.

[12] Hillier, F. S. & Lieberman, G. J. (2001). Introduction to Operations Research. Seventh.

[13] Edition, *McGraw-Hill Series in Industrial Engineering and Management Science*. McGraw-Hill, New York.

[14] Holland, J. H. (1975). Adaptation in natural and artificial systems. Ann Arbor. MI: University of Michigan Press, USA.

[15] Iwundu, M.P., Otaru, O.P., & Onukogu, I.B. (2014). Simultaneous Optimization of Response Surfaces: Using Variance Weighted Gradient. *Journal of Knowledge Management, Economics and Information Technology,* 4 (1.1), 168-174.

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

[17] Lewin, D. R (1994). A genetic algorithm for MIMO feedback control system design. *Advanced Control of Chemical Process,* 101-106.

[18] Lewin, D. R (1994b). Multivariable feed forward control design using disturbance cost map and a genetic algorithm. *Computers & Chemical Engineering*, 28 (12), 1477-1489.

[19] Lewis, R.M. & Torczon, V. (2000). Pattern search methods for linearly constrained minimization. *SIAM Journal on Optimization*, 10 (3), 917-941.

[20] Patil, P.B., and Verma, U.P., (2009). Numerical Computational Methods. Revised Edition, Narosa Publishing House, Chennai Mumbai Kolkata. New Delhi. ISBN 978-81-7319 951-6.

[21] Torczon, V., (1997). On the convergence of pattern search algorithms. *SIAM J. Optimization,* 7, 1.