Multilevel Nelder Mead's Simplex Method

G.N.Sashi Kumar and V.K. Suri
Computational Studies Section, M.D.D., Ch.T.G.,
Bhabha Atomic Research Centre,
Mumbai - 400 085, India, Email: gnsk@barc.gov.in
Telephone : +91-22-2559 3611, Fax: +91-22-2550 5151

***Abstract*—Nelder-Mead's simplex (NMS) algorithm has been the work horse for numerous problems in various fields of engineering since its introduction in 1965 [1, 2]. Being a multi dimensional direct search method its performance is commendable for unconstrained local optimization problems. Despite its popularity NMS fails to converge for higher dimensional problems. In this paper a modification to the basic algorithm is proposed using multilevel approach (is named as MNMS). Also modifications to the parameters used in MNMS have been proposed with the help of sensitivity analysis. MNMS is able to capture the optimum for higher dimensional problems with (i) higher success rate and (ii) faster convergence. Numerical experiments with higher dimensional problems have demonstrated the superiority of MNMS over a class of algorithms based on NMS.**

Keywords—Multilevel; Nelder Mead Simplex; MNMS; Higher dimensions; Dakota.

I. INTRODUCTION

In an undergrad school, the class of optimization starts with Nelder Mead's Simplex(NMS) [3]. The influence of this algorithm is so high that all the simulation packages offering optimization routines contain Nelder Mead's implementation. Non requirement of gradient information and ease of implementation makes Nelder Mead's simplex method a choice for practitioners in many fields. The original algorithm works for unconstrained optimization, although a few modifications claim to be working for constrained cases too [4, 5]. It is an efficient local optimizer with noisy function values, provided the starting point is close to the optimum. Convergence proof of NMS is very limited (on restricted problems) in 1 or 2 dimensions [6]. NMS also fails the converge test on a strictly convex 2 dimensional function [7]. The proof of convergence of NMS is still a open problem. The expansion and contraction steps of NMS have been identified to possess a descent property [8]. Gao and Han have proven the descent property of NMS for a uniformly convex function. Different convergent variants of NMS method were proposed [9-11].

The inability of NMS to capture the optimum when applied for higher dimensional problems has attracted many researchers. Pham and Wilamowski proposed a modification in NMS by evaluating descent direction explicitly using the function values [12]. Yet the modified simplex loses its simplicity as the original NMS. Gao and Han proposes NMS with adaptive parameters for higher dimensional optimization problems (Now onwards referred as GHS). The expansion and contraction steps alter the size of simplex with a factor of $\frac{n_{iter}-1}{n_{iter}^2}$ where n_{iter} is the iteration number. When $n_{iter} \to \infty$ the change brought about with a expansion or contraction step tends to zero [8]. Thus, the approach requires large number of function calls. Also a few failures for GHS at higher dimensions have been recorded (see section III-B).

The convergence of simplex is rapid at the beginning and it slows down as the simplex progresses towards the optimum. It has been reported by many researchers that the convergence of simplex close to the optimum tend to zero [6, 8]. This makes the algorithm deployable for coarse optimization where the exact solution is not important. A few researchers suggest that restart simplex as an alternative to curtail the decline in convergence rate. Luersen and Riche propose the method of restarting degenerate simplices which they claim has made their optimizer capable to capture global optimum [13]. Chang proposed stochastic NMS for global optimization [14]. There have been many hybrid optimizers proposed in combination with NMS using Genetic Algorithms, Swarm colony, etc. [4, 15]. It is still an open problem to get a global optimizer with adaptation to native NMS.

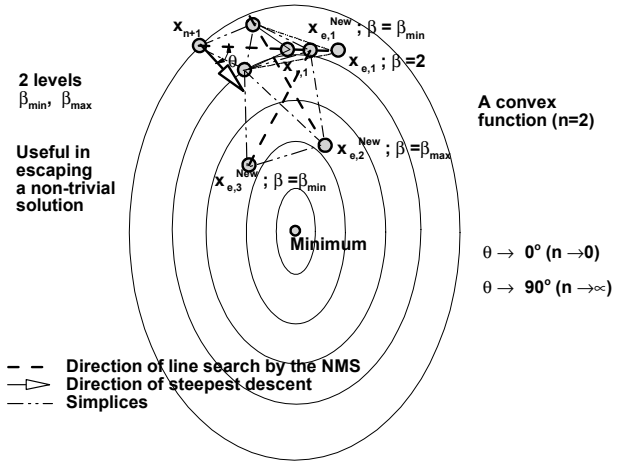

Fig. 1. MNMS for convex function optimization.

In this paper two of the issues of NMS are being addressed. First is to improve the success rate for higher dimensional problems and secondly to improve the convergence rate of the optimizer as it approaches the optimum. A multilevel methodology has been adopted for the purpose. The multilevel strategies have been a major contributor in fields of science, engineering, health services and many more. Multi grid approach in computational geometry and computational

mechanics has paved its way for the developed of efficient solvers [16]. In this paper the multilevel strategy is being exploited to improve the overall behavior of the NMS optimizer. The multilevel approach would mimic the restarting of simplices to some extent. It has shown its potential in avoiding the convergence to a non-critical point.

Section II explains the multilevel simplex algorithm with various options available to be implemented. Section III illustrates the ability of the new schemes vis-a-vis the popular older versions. Conclusions are listed in section IV.

II. Multilevel Nelder Mead's Simplex (MNMS) Algorithm

A. Nelder Mead's Simplex

Simplex is a geometrical figure with $n+1$ non collinear vertices in n dimensional space. The vertices are ranked as per their function values. The vertex of higher function value is relinquished for a better new vertex (with low function value). NMS uses a process of reflecting the worst vertex across the centroid of the simplex. Later readjust this reflected vertex so that it can expand or contract. If admissible vertex is not obtained within these steps the simplex is shrunk, anchoring on to the best vertex of the simplex. The convergence speed of the NMS is affected by four parameters α, β, γ and δ. These parameters alter the size of simplex through the operations reflection, contraction, expansion or shrink [3].

Steps in Nelder Mead's simplex method [8]:

- Sorting: The vertices x_i, $i= 1,..,n+1$ are sorted such that the function values at $n+1$ vertices are

  $f(x_1) \le f(x_2) \le ... \le f(x_{n+1})$

- Reflection: Compute the reflection vertex x_r as

  $x_r = \bar{x} + \alpha(\bar{x} - x_{n+1})$, where $\bar{x} = (\sum_{i=1}^{n} x_i)/n$

  if $f(x_1) \le f(x_r) < f(x_n)$ then replace x_{n+1} with x_r.

- Expansion: If $f(x_r) < f(x_1)$ compute the expansion vertex

  x_e as $x_e = \bar{x} + \beta(x_r - \bar{x})$

  if $f(x_e) < f(x_r)$ then replace x_{n+1} with x_e else with x_r.

- Outside Contraction (OC): If $f(x_n) \le f(x_r) < f(x_{n+1})$ compute the OC vertex x_{oc} as $x_{oc} = \bar{x} + \gamma(x_r - \bar{x})$

  if $f(x_{oc}) \le f(x_r)$ then replace x_{n+1} with x_{oc} else Shrink

- Inside Contraction (IC): If $f(x_{n+1}) \ge f(x_r)$ compute the IC vertex x_{ic} as $x_{ic} = \bar{x} - \gamma(x_r - \bar{x})$

  if $f(x_{ic}) < f(x_{n+1})$ then replace x_{n+1} with x_{ic} else Shrink

- Shrink:

  $x_i = x_i + \delta(x_i - x_1); \forall\, 2 \le i \le n+1$

The details of the parameters are listed in Table I.

B. Multilevel Nelder Mead's Simplex (MNMS)

A single objective optimization problem has two concerns. First being the error between attained optimum f^{opt} and exact optimum f^*, *i.e.* $| f^{opt} - f^* |$ and convergence rate. The modifications performed in the process of developing MNMS does not alter the basic structure of NMS. This has not altered the CPU time required per function call. In the comparative studies performed in this paper, the convergence rate has been approximated by *Number of Function Calls* (#*FC*) required for obtaining the optimum. In order to minimize the error with good convergence rate, the problem needs to be treated as 'multivariate problem'. The Hierarchical Linear nonlinear Modeling(HLM) introduced by Snijders & Bosker [17] emphasizes the use of 'multilevels' for multivariate problems. Also multilevel strategy for grid topology is used in multigrid optimization approach where interpolations are used to accelerate the solvers. Multilevel has been popular in industry and research due to its basic nature of resolution of scales at different levels. In the present work the multilevel is used in a different sense. There are no explicit markings as various levels nor any interpolations to be performed. In NMS, the expansion step (parameter associated, β) is the cause for increase in maximum edge length of the simplex. A cyclic correction in β value exhibiting a multilevel behavior has been introduced. Fig. 1 shows a multilevel expansion step. Next section demonstrates the improvement of convergence speed with multilevel approach.

C. Usefulness of MNMS

A case study has been performed with the chained Rosenbrock function with functional tolerance of $1.\times10^{-10}$. Although the results deduced from this case study cannot be generalized. Yet, in the later portion of this paper, attempts have been made to prove that the deductions obtained based on Rosenbrock are valid for other functions which were tested. The simplex optimizer would be successful at higher dimensions if it uses less number of reflection operations [8]. Fig. 2 shows that the Fraction Of Reflections (FOR) taking place with the dimensionality of the problem. Comparison between the 4 methods (NMS, GHS, 2 new proposed variants i.e. V-MNMS and W-MNMS) show that the FOR increase with dimensionality of the problem. FOR tend to 1 for NMS and GHS when tested up to dimensions of 100. The zone marked in grey shows the zone in which the optimizer fails to reach the optimum. Using MNMS, FOR saturates at 0.91 as well as succeeds in all cases tested. NMS and GHS failure to optimize may be attributed to this high value of FOR. The maximum edge length (E) of the simplex should → 0 as the iterations proceed. This quantity is almost the ratio between the sizes of the initial and current simplexes. E may be defined as following

$$E = \alpha^{NR} \times \beta^{NE} \times \gamma^{NC} \times \delta^{NS} \quad (1)$$

where NE, NR, NC and NS are number of expansion, reflection, contraction and resizing steps respectively. Fig. 3 shows the E of simplex at convergence ($E_{iter \to \infty}$) with increasing dimensions for the chained Rosenbrock problem. The convergence assurance is inbuilt in the NMS due to its choice of parameters. i.e. the simplex would shrink to a point at

convergence. Fig. 3 illustrates that GHS is unable to handle the convergence issue at higher dimensions and would settle at a non-critical value. In contrast the parameters selected for the MNMS is making it successful at higher dimensions.

D. Conditions for convergence based on Rosenbrock function

1) $E_{iter\to\infty}\to 0$; 2) $FOR \le 0.92$

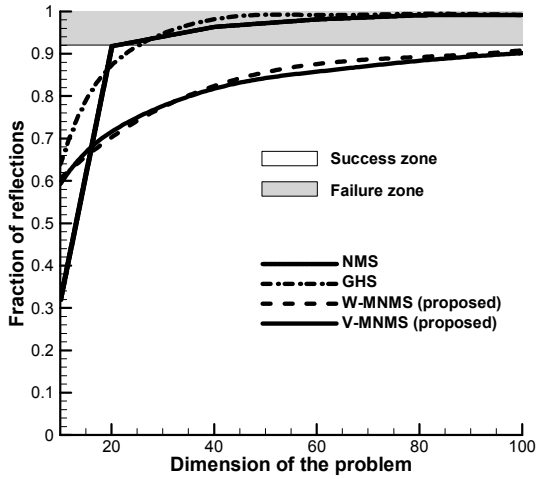

Fig. 2. The effect of reflection parameter with change in dimension for chained Rosenbrock problem.

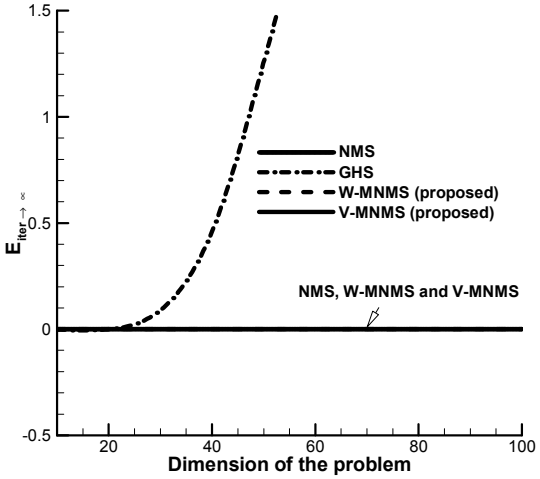

Fig. 3. The combined effect of various parameters on simplex (*E)* for chained Rosenbrock problem.

The second condition may be obtained by comparing the Fig. 2 with Table V shows a cutoff valve for reflection ratio below which the optimizer converges. Fig. 2 illustrates that the number of reflections at higher dimensions have decreased with use of MNMS. Apart from this the multilevel approach imparts faster convergence and also higher success rate for the algorithm.

E. Assigning values to α, β, γ by sensitivity analysis

Although the concept of multilevel has been adopted, parameters at each level needs to be determined. A sensitivity analysis has been performed using five test functions, namely i) Rosenbrock, ii) Griewank, iii) Rastrigin, iv) Vardim and v) Sphere [18]. Sensitivity of individual test functions was obtained as average over 100 evaluations. Each of these has unique 100 random initial points. Such individual sensitivities averaged for all the test functions has been plotted for various dimensions of the problem with varying values of α, β and γ. Fig. 4, 5 and 6 show sensitivity of coefficients α, β and γ respectively on dimensionality. Table I shows a comparison of these constants with the prominent literature data. It may be noted that GHS was proposed for higher dimensions, while others methods report up to 4 dimensions [6].

TABLE I. COMPARISON OF PARAMETERS FOR **NMS** VARIANTS

Para-meters	NMS	Gao and Han [8]	Fan and Zahara [15]	Wang and Shoup [19]	Present Work
α	1.0	1.0	1.5	Avg =1.29, σ=0.47	$1+\frac{0.6}{n}$
β	2.0	$1+\frac{2}{n}$	2.75	Avg =2.29, σ=0.44	1.2 and 1.05 in W or V multilevel
γ	0.5	$0.75-\frac{1}{2n}$	0.75	Avg =0.47, σ=0.17	$0.95-\frac{3}{n}-\frac{3}{n^2}$
δ	0.5	$1-\frac{1}{n}$	0.5	Avg =0.57, σ=0.19	$1-\frac{1}{n}$

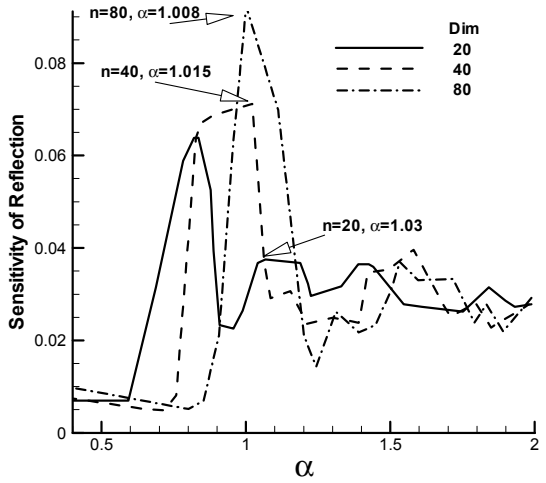

Fig. 4. The sensitivity of α at higher dimensions.

Sensitivity analysis of α showed that the peak shifts towards 1.0 with increase in dimension of the problem. Fig. 4 shows the values for α for example at n=20, α=1.03, likewise the peaks at various problem dimension are determined. A best response surface that could represent this data was arrived.

$$\alpha = 1 + \frac{0.6}{n} \tag{2}$$

Sensitivity analysis of β reveals that there are two prominent peaks at higher dimensions at 1.20 and 1.05. They do not depend on dimension, for all $n \ge 10$. With the above observations the following conclusions have been arrived.

$$\beta = \{1.2,\ 1.05\}\ \forall\, n \ge 10 \tag{3}$$

There are only two prominent peaks in the sensitivity study of β which prompts us to use 2-level methods. Two multi-level approaches have been proposed in this paper. They are illustrated in Fig. 7.

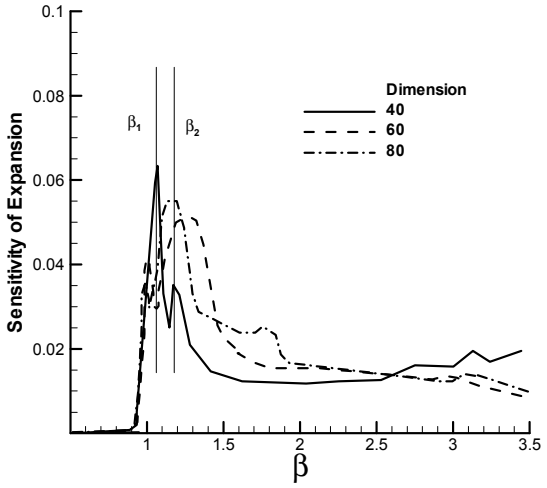

Fig. 5. The sensitivity of β at higher dimensions.

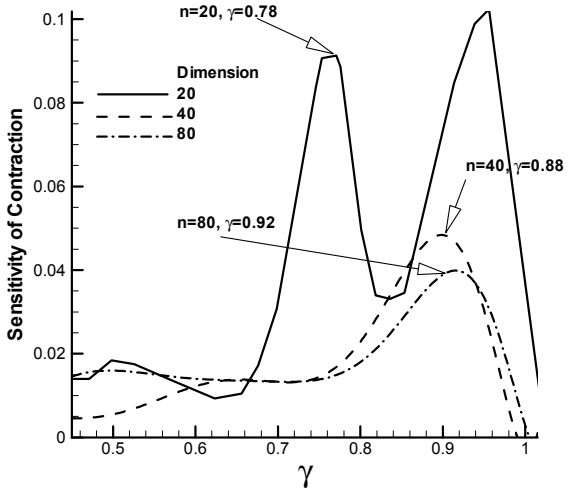

Fig. 6. The sensitivity of γ at higher dimensions.

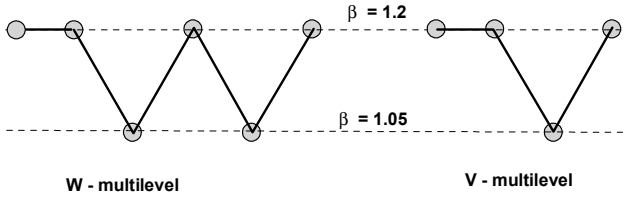

Fig. 7. The multi-level approach W-Multilevel Nelder Mead Simplex (MNMS) and V-MNMS.

Sensitivity analysis of γ reveals that there are multiple prominent peaks. For example at n=20 there are two peaks at γ = 0.78 and 0.95. Fan *et al.* [15] has reported a value of γ = 0.75, which is close to 0.78. The success rate of chained Rosenbrock function was chosen as a measure for selection of best suited γ value at each dimension.

TABLE II. SUCCESS RATE OF OPTIMIZATION WITH DIFFERENT CONTRACTION COEFFICIENTS

Function [a]	Dimension	γ	W-simplex	V-simplex
Ch. Rosenbrock	20	0.78	100%	100%
(various initial points)	20	0.95	85%	80%

[a] $f(\vec{x}^*) = \vec{0}$

TABLE III. CONTRACTION PARAMETER γ AS FUNCTION OF PROBLEM DIMENSION

Dimension	*Value of* γ
10	0.65
20	0.78
40	0.88
60	0.90
80	0.92
100	0.92

An equation best representing the data in Table III can be written as :

$$\gamma = 0.95 - \frac{3}{n} - \frac{3}{n^2} \qquad (4)$$

An aggravated resizing has shown accelerated convergence to a non-critical point. Such a behavior is unwarranted. The resizing coefficient δ has been borrowed from the GHS approach.

F. Comparison with Dakota

With the new set of parameters and multilevel approach for β the new method MNMS is ready for testing. A comparison study is performed with simplex approach implemented in Dakota solver. The Dakota (Design Analysis Kit for Optimization and Terascale Applications) is an open source toolkit developed by Sandia National Laboratories. Two Simplex based methods are available in Dakota. 1) Parallel Direct search method (*optpp_pds*) that can handle bound constraints and 2) Constrained Optimization by Linear Approximations (*coliny_cobyla*) which can handle nonlinear constraints [20]. COBYLA method has been used to draw a comparison with MNMS in this paper. The default configuration has been retained and tests with chained Rosenbrock function were conducted at higher dimensions. The convergence tolerance was set to $1.\times10^{-10}$ with a limit on number of function calls as $1.\times10^{6}$. Table IV shows the comparison of MNMS with Dakota solver. Dakota requires large number of function calls when compared with MNMS.

TABLE IV. COMPARISON OF CONVERGENCE OF ROSENBROCK FUNCTION ($\vec{x} = (0.0,...,0.1), f(\vec{x}^*) = \vec{0}$) FOR V-MNMS WITH DAKOTA

Dimension	Dakota		V-MNMS	
	# FC	f_{final}	***# FC***	f_{final}
10	79,020	9.98×10^{-7}	4,026	7.57×10^{-21}
20	151,960	9.98×10^{-7}	18,042	3.41×10^{-20}
40	348,680	9.99×10^{-7}	89,320	3.99×10^{-20}
60	643,350	9.99×10^{-7}	234,469	6.35×10^{-20}
80	10^6	1.28×10^{-6}	491,450	1.94×10^{-19}
100	10^6	5.86×10^{-5}	860,744	1.37×10^{-19}

III. NUMERICAL EXPERIMENTS

Exhaustive comparative study was performed using the original Nelder Mead implementation (NMS), Gao & Hun (GHS) method and the proposed W-MNMS & V-MNMS. The following test functions have been chosen so that a vast variety of problems could be mimicked through them.

A. Test Functions chosen for comparative study [18]

(i)Ackley Function, (ii)Dixon and Price Function, (iii) Chained Rosenbrock Function, (iv)Trid Function, (v)Zakharov Function, (vi)Powell Function and (vii)Sphere Function.

B. Results and Discussion

a) Higher Dimensions: Table V shows the comparative study with seven test functions with increasing problem dimensions from 10 to 100. The functional tolerance was maintained at 10^{-10} and maximum number of function evaluations was fixed at 10^6. The parameters that were used with MNMS can be summarized as below.

$$\alpha = 1 + \frac{0.6}{n};\ \beta = 1.2\ \&\ 1.05;\ \gamma = 0.95 - \frac{3}{n} - \frac{3}{n^2};\ \delta = 1 - \frac{1}{n} \qquad (5)$$

β is implemented in W or V multilevel algorithm. Table V shows that the MNMS methods irrespective of whether they are W or V multilevel have performed much better than GHS and NMS. The comparison trials use same initial points. It can be seen that the NMS is the worst performer among the lot. The GHS method that is claimed to be developed for higher dimensions fails for the functions considered. The two new methods proposed converge to optimum with less number of function calls. The new methods serve in attaining (i) assured convergence to optimum and (ii) faster convergence within lesser number of function calls.

b) Faster Convergence : Table VI illustrates the convergence of the tested algorithms. The tolerance for convergence has

TABLE V. COMPARISON OF RESULTS FOR THE 4 APPROACHES CONSIDERED

Function[b]	Dimension	NMS		GHS		W-MNMS		V-MNMS	
		# FC	f_{final}	# FC	f_{final}	# FC	f_{final}	# FC	f_{final}
Ackley	10	3,053	2.63×10^{-10}	2,094	1.13×10^{-10}	1,845	1.17×10^{-10}	1,847	1.16×10^{-10}
$\vec{x} = 0.0,...,0.1$	20	52,027	3.29×10^{-10}	7,539	9.01×10^{-11}	8,464	1.66×10^{-10}	7,832	9.76×10^{-11}
$f(\vec{x}^*) = \vec{0}$	40	167,654	0.056	53,350	8.43×10^{-11}	31,047	8.84×10^{-11}	36,083	1.05×10^{-10}
	60	620,954	0.067	170,321	1.84×10^{-10}	68,636	8.03×10^{-11}	72,269	8.47×10^{-11}
	80	10^6	0.046	624,262	1.45×10^{-10}	119,924	7.75×10^{-11}	112,537	8.05×10^{-11}
	100	10^6	0.058	10^6	1.30×10^{-3}	203,931	7.85×10^{-11}	190,634	7.35×10^{-11}
Sphere	10	5,512	2.82×10^{-20}	3,100	7.99×10^{-21}	2,559	8.35×10^{-21}	2,556	8.35×10^{-21}
$\vec{x} = 0.0,...,5.0$	20	159,203	1.48×10^{-19}	11,006	1.03×10^{-20}	10,466	1.22×10^{-20}	10,447	1.30×10^{-20}
$f(\vec{x}^*) = \vec{0}$	40	683,688	0.155	111,951	1.73×10^{-20}	41,799	2.20×10^{-20}	39,856	2.29×10^{-19}
	60	10^6	10.12	519,765	2.16×10^{-20}	96,217	2.54×10^{-20}	90,153	3.20×10^{-20}
	80	10^6	30.64	10^6	3.19×10^{-2}	183,657	3.67×10^{-20}	116,152	3.47×10^{-20}
	100	10^6	28.90	10^6	7.85×10^{-2}	270,286	3.54×10^{-20}	262,484	3.55×10^{-20}
Chained Rosenbrock	10	6,394	8.56×10^{-21}	4,492	6.02×10^{-21}	4,026	7.57×10^{-21}	4,025	7.59×10^{-21}
$\vec{x} = 0.0,...,0.1$	20	140,650	3.47×10^{-19}	28,292	1.07×10^{-20}	18,042	3.41×10^{-20}	17,946	2.16×10^{-20}
$f(\vec{x}^*) = \vec{0}$	40	205,503	38.54	388,590	1.13×10^{-20}	89,320	3.99×10^{-20}	88,311	4.13×10^{-20}
	60	592,470	58.38	10^6	31.74	234,469	6.35×10^{-20}	233,968	9.26×10^{-20}
	80	10^6	78.23	10^6	77.47	491,450	1.94×10^{-19}	478,403	1.16×10^{-19}
	100	10^6	97.76	10^6	97.46	860,744	1.37×10^{-19}	847,003	1.27×10^{-19}
Trid	10	6,583	0.664	3,233	1.52×10^{-19}	2,925	1.65×10^{-19}	2,650	4.35×10^{-19}
$\vec{x} = -5.0,...,0.0$	20	36,594	2.926	11,559	2.09×10^{-19}	10,046	2.73×10^{-19}	10,767	2.47×10^{-19}
$f(\vec{x}^*) = \vec{0}$	40	476,158	10.40	103,166	2.89×10^{-19}	38,100	4.93×10^{-19}	39,631	3.10×10^{-19}
	60	10^6	11.30	466,423	4.82×10^{-19}	90,476	5.82×10^{-19}	84,046	5.84×10^{-19}
	80	10^6	13.60	10^6	2.275	152,116	8.09×10^{-19}	164,372	8.62×10^{-19}
	100	10^6	38.80	10^6	2.546	292,601	1.28×10^{-18}	260,523	9.61×10^{-19}
Dixon & Price	10	2,337	3.19×10^{-20}	1,695	5.50×10^{-20}	1,723	3.39×10^{-20}	1,656	4.86×10^{-20}
$\vec{x} = 0.9,...,1.0$	20	57,339	1.73×10^{-19}	11,176	6.67×10^{-20}	6,209	4.68×10^{-20}	6,469	3.50×10^{-20}
$f(\vec{x}^*) = \vec{0}$	40	333,668	1.192	22,785	3.58×10^{-20}	21,144	1.12×10^{-19}	20,864	1.17×10^{-19}
	60	10^6	8.113	48,498	7.37×10^{-20}	38,400	2.27×10^{-19}	40,133	1.97×10^{-19}
	80	10^6	0.994	110,039	3.94×10^{-19}	62,188	2.97×10^{-19}	67,666	1.74×10^{-19}
	100	10^6	3.330	192,212	3.19×10^{-19}	87,845	4.06×10^{-19}	88,441	2.32×10^{-19}
Powell	10	8,325	5.65×10^{-46}	7,605	1.05×10^{-44}	6,972	1.62×10^{-41}	6,725	1.01×10^{-41}
$\vec{x} = 0.9,...,1.0$	20	337,823	2.79×10^{-25}	42,645	2.14×10^{-35}	42,649	2.16×10^{-35}	39,625	4.87×10^{-35}
$f(\vec{x}^*) = \vec{0}$	40	10^6	1.30×10^{-5}	445,659	9.86×10^{-33}	215,223	2.58×10^{-34}	196,767	2.56×10^{-35}
	60	10^6	5.33×10^{-4}	10^6	4.74×10^{-14}	555,196	3.07×10^{-34}	566,154	4.44×10^{-35}
	80	10^6	7.60×10^{-3}	10^6	7.40×10^{-5}	10^6	5.27×10^{-31}	10^6	5.00×10^{-27}
Zakhov	10	6,921	1.94×10^{-20}	3,041	5.20×10^{-21}	3,186	1.09×10^{-20}	2,938	7.14×10^{-21}
$\vec{x} = 0.0,...,0.1$	20	36,186	5.50×10^{-3}	18,899	1.04×10^{-20}	15,788	1.84×10^{-20}	14,782	1.31×10^{-20}
$f(\vec{x}^*) = \vec{0}$	40	97,975	0.099	161,548	1.95×10^{-20}	90,643	2.64×10^{-20}	84,927	2.64×10^{-20}
	60	159,420	0.215	900,699	9.90×10^{-20}	227,803	3.69×10^{-20}	243,346	2.85×10^{-20}
	80	337,429	0.266	10^6	8.70×10^{-3}	490,759	4.94×10^{-20}	488,503	4.19×10^{-20}
	100	504,844	0.276	10^6	1.83×10^{-2}	955,984	4.65×10^{-20}	837,186	4.08×10^{-20}

FC = Number of function calls

been reduced to 10^{-4}. This has been done so that the ability of the algorithm for coarse optimization can be estimated. Here the NMS has lost its capability to converge to the optimum. GHS algorithm starts to diverge above 60 dimensions. So, a comparison is shown only up to 60 dimensions. Having known the number of function calls (FC_{n1}) for a given dimension *n1*, the requirement of number of function calls (FC_{n2}) for a new dimension *n2* can be approximated as follows.

$$\frac{FC_{n2}}{FC_{n1}} = A + B\,\frac{n2}{n1} + C\left(\frac{n2}{n1}\right)^2 \tag{6}$$

Using GHS algorithm, the constants are A = 52, B = -61.67 and C = 2.03, which shows that there is quadratic growth in number of function calls with dimension of the problem. Whereas, the values of constants are A = -6.93, B = 5.93 and C = 0 when MNMS algorithms is used, which emphasizes the linear growth of number of function calls with dimension of the problem.

TABLE VI. COMPARISON OF FUNCTION CALLS REQUIRED FOR THE 4 APPROACHES CONSIDERED

Function[b]	Dimension	NMS	GHS	W-MNMS	V-MNMS
Sphere $\vec{x}=0.0,...,1.0$ $f(\vec{x}^*)=\vec{0}$	**20**	#	8,764	7,602	7,398
	40	#	107,319	32,183	30,716
	60	#	512,638	79,038	73,079
Ackley $\vec{x}=0.0,...,1.0$ $f(\vec{x}^*)=\vec{0}$	**20**	#	5,428	5,393	4,804
	40	#	48,167	21,455	24,881
	60	#	154,286	51,201	54,479
Trid $\vec{x}=0.0,...,1.0$ $f(\vec{x}^*)=\vec{0}$	**20**	#	9,499	7,296	7,903
	40	#	98,575	29,125	30,671
	60	#	459,172	73,272	66,996
Powell $\vec{x}=0.0,...,1.0$ $f(\vec{x}^*)=\vec{0}$	**20**	#	14,613	15,714	14,461
	40	#	157,852	74,527	75,765
	60	#	824,939	179,425	176,109

[b] $Tol_f = 10^{-4}$, # unsuccessful runs

IV. CONCLUSIONS

The Nelder Mead Simplex a forerunner in unconstrained optimization has been evaluated for its performance at higher dimensions. Also a recent modification proposed by Gao and Han for higher dimensions was also evaluated. Gao and Han proposed that the reflection percentage in the simplex would decide the possibility of success or failure of the simplex. We have extended this idea. The necessary condition for convergence has been formulated [Section II-D] and is the appropriate measure for assigning of success of convergence.

A sensitivity study was performed to determine the reflection, contraction and expansion coefficients and their dependence with dimension of the problem. The expansion coefficient converged to a fixed value of 1.05 and 1.20 and is independent of dimension. A multi level expansion step along with modifications in other coefficients has been proposed. Two types of multilevel method W-MNMS and V-MNMS were proposed. Numerical studies performed revealed the superiority of the new approaches in comparison to the existing NMS algorithms for unconstrained local optimization problems at higher dimensions. This method when coupled with a global search technique like genetic algorithm, etc. can lead to a fast optimization solver.

ACKNOWLEDGMENT

The authors would like to thank Dr. A.K. Mahendra, BARC, Mumbai for useful discussions on NMS and Dakota.

REFERENCES

[1] L. Nazareth, P. Tseng, Gilding the Lily: A Variant of the Nelder-Mead Algorithm Based on Golden-Section Search, Comput. Optim. Appl, vol. 22, no. 1, pp. 133–144 (2002)

[2] K. Klein, J. Neira, Nelder-Mead Simplex Optimization Routine for Large-Scale Problems: A Distributed Memory Implementation, Comput Econ., DOI 10.1007/s10614-013-9377-8 (March 2013)

[3] J. A. Nelder, R. Mead, A Simplex Method for Function Minimization, Computer Journal, 7, 308-313 (1965)

[4] E. Zahara, Y. Kao, Hybrid Nelder–Mead simplex search and particle swarm optimization for constrained engineering design problems, Expert Systems with Applications, 36, 3880–3886 (2009)

[5] H. Ghiasi, D. Pasini, L. Lessard, Constrained Globalized Nelder–Mead Method for Simultaneous Structural and Manufacturing Optimization of a Composite Bracket, J of Compos. Mat., 42, Issue 7, 717-736 (2008)

[6] J.C. Lagrias, J.A. Reeds, M.H. Wright, P.E. Wright, Convergence properties of the Nelder-Mead Simplex method in low dimensions, SIAM J. Optimization, 9, Issue 1, 112-147 (1998)

[7] K. I. M. McKinnon. (1998) Convergence of the Nelder--Mead Simplex Method to a Nonstationary Point. SIAM Journal on Optimization 9:1, 148-158.

[8] F. Gao, L. Han, Implementing the Nelder-Mead simplex algorithm with adaptive parameters, Comput Optim Appl, DOI 10.1007/s10589-010-9329-3 (May 2010)

[9] Árpád Burmen, Tadej Tuma, Unconstrained derivative-free optimization by successive approximation, Journal of Computational and Applied Mathematics, vol. 223, no. 1, pp. 62-74, 2009

[10] Burmen A, Puhan J, Tuma T, Grid restrained Nelder-Mead algorithm, Computational Optimization and Applications 2006; 34: 359-375.

[11] C. J. Price, I. D. Coope, D. Byatt, Journal of Optimization Theory and Applications, Volume 113 Issue 1, April 2002 Pages 5 - 19.

[12] N. Pham, B.M. Wilamowski, Improved Nelder Mead's Simplex Method and Applications, Journal of Computing, 3,issue 3, 55-63 (2011)

[13] M.A. Luersen, R.Le Riche, Globalized Nelder–Mead method for engineering optimization, Computers and Structures, 82, 2251–2260 (2004)

[14] K. Chang, Stochastic Nelder–Mead simplex method – A new globally convergent direct search method for simulation optimization, European Journal of Operational Research, 220, 684–694 (2012)

[15] S.S. Fan, E. Zahara, A hybrid simplex search and particle swarm optimization for unconstrained optimization, European Journal of Operational Research, 181, 527–548 (2007)

[16] A. Borzi, V. Schulz, Multigrid methods for PDE optimization, SIAM Review, 51(2), 361-395 (2009)

[17] Tom Snijders and Roel Bosker, Multilevel Analysis: An introduction to basic and advanced multilevel modeling, 2nd edn., Sage Publishers, London (2012)

[18] P.C. Wang, T.E. Shoup, Parameter sensitivity study of the Nelder–Mead Simplex Method, Advances in Engineering Software, 42, 529–533 (2001)

[19] Downloaded from http://www.sfu.ca/~ssurjano/index.html accessed on 18 Dec 2013

[20] Dakota, A multilevel parallel OOF for design optimization, parameter estimation, uncertainity quantification & sensitivity analysis, V5.3, Feb 2013, SAND2010-2183, Sandia National Laboratories, LM,CA94551.