The Nelder-Mead Simplex Method with Variables Partitioning for Solving Large Scale Optimization Problems*

Ahmed Fouad Ali[1], Aboul Ella Hassanien[2], and Václav Snášel[3]

[1] Suez Canal University, Dept. of Computer Science,
Faculty of Computers and Information, Ismailia, Egypt
Member of Scientific Research Group in Egypt
[2] Cairo University, Faculty of Computers and Information, Cairo, Egypt
Chair of Scientific Research Group in Egypt
[3] VSB-Technical University of Ostrava, Czech Republic
vaclav.snasel@vsb.cz

Abstract. This paper presents a novel method to solve unconstrained continuous optimization problems. The proposed method is called SVP (simplex variables partitioning). The SVP method uses three main processes to solve large scale optimization problems. The first process is a variable partitioning process which helps our method to achieve high performance with large scale and high dimensional optimization problems. The second process is an exploration process which generates a trail solution around a current iterate solution by applying the Nelder-Mead method in a random selected partitions. The last process is an intensification process which applies a local search method in order to refine the the best solution so far. The SVP method starts with a random initial solution, then it is divided into partitions. In order to generate a trail solution, the simplex Nelder-Mead method is applied in each partition by exploring neighborhood regions around a current iterate solution. Finally the intensification process is used to accelerate the convergence in the final stage. The performance of the SVP method is tested by using 38 benchmark functions and is compared with 2 scatter search methods from the literature. The results show that the SVP method is promising and producing good solutions with low computational costs comparing to other competing methods.

1 Introduction

Nelder and Mead devised a local search method for finding the local minimum of a function of several variables, the method is called the Nelder-Mead method

* This work was partially supported by Grant of SGS No. SP2013/70, VSB - Technical University of Ostrava, Czech Republic., and was supported by the European Regional Development Fund in the IT4Innovations Centre of Excellence project (CZ.1.05/1.1.00/02.0070) and by the Bio-Inspired Methods: research, development and knowledge transfer project, reg. no. CZ.1.07/2.3.00/20.0073 funded by Operational Programme Education for Competitiveness, co-financed by ESF and state budget of the Czech Republic.

A. Abraham et al. (eds.), *Innovations in Bio-inspired Computing and Applications,*
Advances in Intelligent Systems and Computing 237,
DOI: 10.1007/978-3-319-01781-5_25,

[13]. The method is one of the most popular derivative-free nonlinear optimization methods. A simplex is a triangle, for function of two variables and the method is a pattern search that compares function values at the three vertices of a triangle. The worst vertex is rejected and replaced with a new vertex. A new triangle is formed and the search is continued. The process generates a sequence of triangles for which the function values at the vertices get smaller and smaller. The size of the triangles is reduced and the coordinates of the minimum point are found. The algorithm is stated using the term simplex (a generalized triangle in N dimensions) and will find the minimum of a function of N variables. Four scaler parameter must be specified to define a complete Nelder-Mead method; coefficients of reflection ρ, $\rho > 0$, expansion χ, $\chi > 1$, contraction γ, $0 < \gamma < 1$ and shrinkage σ, $0 < \sigma < 1$ as shown in Figure 1. The Nelder-Mead algorithm steps are described in Algorithm 1 and all its parameters are defined in Table 1. In this paper, we proposed a new method based on the simplex Nelder-Mead method. The proposed method is called SVP (simplex variable partitioning). The main goal of the SVP method is construct an efficient method to solve unconstrained large scale optimization problems. SVP starts with a random initial solution, the iterate solution is divided into pre-specified number of partition. In order to generate a trail solution around the iterate solution the Nelder-Mead method has been applied in a random selected partitions. The trail solution with the best objective function is always accepted. In the final stage a local search method is applied in order to accelerate the search instead of letting the algorithm running for several iterations without much significant improvement of the objective function values. The SVP method is compared with 2 main scatter search methods by using 38 benchmark functions with different properties (uni-model, multi-model, shifted, rotated). The numerical results show that SVP method is a promising method and faster than other methods. The rest of the paper is organized as follows. The next section survey the related work on high dimension and large scale optimization problems. Section 3 describes the proposed SVP method. The performance of the SVP method and its numerical results are reported in Sections 4, 5. The conclusion of this paper is summarized in Section 6.

2 Related Work

Many researches have been attracted to apply their works to solve the global optimization problems, this problems can expressed as follows.

$$\begin{aligned} &Minimize \quad f(x) \\ &Subject\ to \quad l \leq x \leq u, \end{aligned} \tag{1}$$

where $f(x)$ is a nonlinear function, $x = (x_1, \ldots, x_n)$ is a vector of continuous and bounded variables, $x, l, u \in \Re^n$.

Although the efficiency of there works when applied with lower and middle dimensional problems e.g $D < 100$, they suffer from the curse of dimensionality

when applied to large scale and high dimensional problems. Some efforts have been done to overcome this problem. The quality of any proposed method to solve the large scale and optimization problem is the capability of performing the wide exploration and the deep exploration processes. These two processes have been invoked in many works through different strategies, see for instances [3], [6], [7], [12]. These two processes have been considered in the SVP method through three strategies as follows. The dimension reduction process which the search space can be divided into smaller partitions. The exploration process where the trail solutions are generated around the iterate solution. In the variable neighborhood search [14], the search space is treated as nested zones and each one is searched through iterative solutions [1], [10]. Finally the intensification process by applying a local search method with the elite solution obtained from the pervious stage [4], [5], [7]. Invoking these strategies together in the SVP method is the main difference between it and other related methods existing in the literature.

Table 1. Parameters used in Algorithm 1

Parameters	Definitions
x_r	Reflection point
x_e	Expansion point
x_{oc}	Outside contraction point
x_{ic}	Inside contraction point
ρ	Coefficients of reflection
χ	Coefficients of expansion
γ	Coefficients of contraction
σ	Coefficients of shrinkage

3 The Proposed SVP Method

In this section a proposed method is presented for solving large scale optimization problems. The proposed method is called SVP (simplex variable partitioning). SVP starts with a random initial solution and consists of n variables. The solution is divided into η partitions, each partition contains ξ variables (if the number of variables n is not a multiple of ξ, a limited number of dummy variables my be add to the last partition). At a fixed number of iteration (SVP inner loop), a random partition is selected, and a trail solution is generated by applying the simplex Nelder-Mead method in the selected partition. The overall trail solution is accepted if it's objective function value is better than the previous solution. Otherwise the trial solution is rejected. The scenario is repeated until the termination criteria satisfied (SVP outer loop). In order to refine the best solution, SVP method applies a local search method as final intensification process. The definitions of the used parameters in SVP method is reported in Table 2. In the next subsections we give more descriptions of SVP method.

Algorithm 1. The Nelder-Mead method

1. Set x_i denote the list of vertex in the current simplex, $i = 1, \ldots, n+1$.
2. Order Order and re-label the $n+1$ vertices from lowest function value $f(x_1)$ to highest function value $f(x_{n+1})$ so that $f(x_1) \leq f(x_2) \leq \ldots \leq f(x_{n+1})$.
3. Reflect. Compute the reflection point x_r by
$x_r = \bar{x} + \rho(\bar{x} - x_{(n+1)})$,
where $\bar{x}$ is the centroid of the n best points,
$\bar{x} = \sum(x_i/n), i = 1, \ldots, n$.
if $f(x_1) \leq f(x_r) < f(x_n)$ **then**
    replace x_{n+1} with the reflected point x_r.
    go to Step 7.
end if
4. Expand.
if $f(x_r) < f(x_1)$ **then**
    Compute the expansion point x_e by $x_e = \bar{x} + \chi(x_r - \bar{x})$.
end if
if $f(x_e) < f(x_r)$ **then**
    replace x_{n+1} with x_e and go to Step 7.
else
    replace x_{n+1} with x_r and go to Step 7.
end if
5. Contract.
if $f(x_r) \geq f(x_n)$ **then**
    perform a contraction between $\bar{x}$ and the better of x_{n+1} and x_r.
end if
Outside contract.
if $f(x_n) \leq f(x_r) < f(x_{n+1})$ **then**
    Calculate $x_{oc} = \bar{x} + \gamma(x_r - \bar{x})$.
    if $f(x_{oc}) \leq f(x_r)$ **then**
        replace x_{n+1} with x_{oc}
        go to Step 7.
    else
        go to Step 6.
    end if
end if
Inside contract.
if $f(x_r) \geq f(x_{(n+1)}$ **then**
    Calculate $x_{ic} = \bar{x} + \gamma(x_{n+1} - \bar{x})$.
end if
if $f(x_{ic}) \geq f(x_{(n+1)}$ **then**
    replace x_{n+1} with x_{ic}
    go to Step 7.
else
    go to Step 6.
end if
6. Shrink. Evaluate the n new vertices
$x' = x_1 + \sigma(x_i - x_1), i = 2, \ldots, n+1$.
Replace the vertices $x_2, \ldots, x_{n+1}$ with the new vertices $x'_2, \ldots, x'_{n+1}$.
7. Stopping Condition. Order and re-label the vertices of the new simplex as $x_1, x_2, \ldots, x_{n+1}$ such that $f(x_1) \leq f(x_2) \leq \ldots \leq f(x_{n+1})$.
if $f(x_{n+1}) - f(x_1) < \epsilon$ **then**
    stop, where $\epsilon > 0$ is a small predetermined tolerance.
else
    go to Step 3.
end if

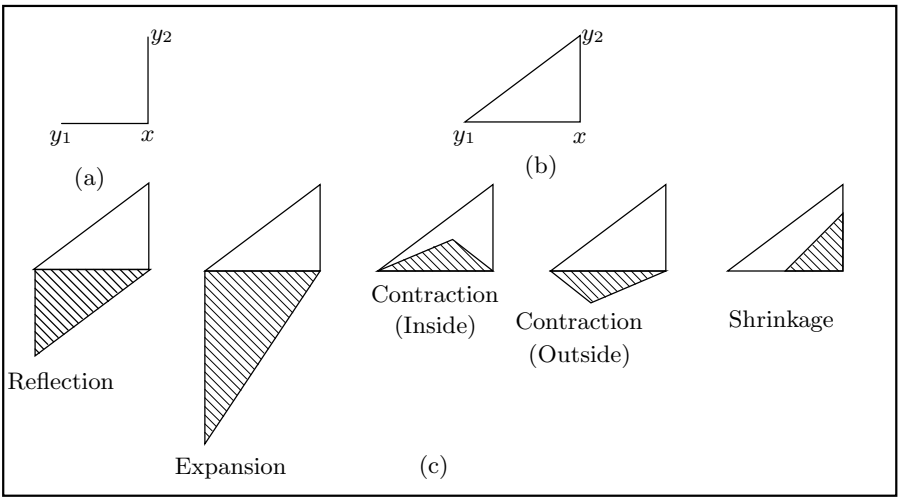

Fig. 1. Nelder-Mead search strategy in two dimensions

Table 2. Parameters used in Algorithm 2

Parameters	Definitions
n	No. of variables
ξ	Partition size
η	No. of partitions
ϕ	Random selected partition
x^0	Initial solution
$x\prime$	Best trail solution
$Maxitr$	Maximum number of iterations
N_{elite}	No. of best solution for intensification

3.1 Variable Partitioning and Trail Solutions Generating

SVP method starts with an iterate solution which divided into η small partitions, where $\eta = n/\xi$. Searching the partitioned subspaces is controlled by applying the search in a limited number of subspaces in the current iteration. This allows the SVP method to intensify the search process in each iteration. However, choosing different subspaces in consequent iterations maintains the search diversity. Moreover, searching a limited number of subspaces prevents SVP from wandering in the search space especially in high dimensional spaces. The variable partitioning process with $\xi = 4$ is shown in Figure 2. At a fixed number of iterations (SVP inner loop), a random partitions are selected in order to generate a trail solutions by applying the Nelder-Mead method in each selected partition ϕ as shown in Algorithm 1. If the overall trail solution objective function is better the previous solution, then the trail solution is selected to become the current solution. The operation of generating new trail solutions is repeated until stoping criteria satisfied (SVP outer loop).

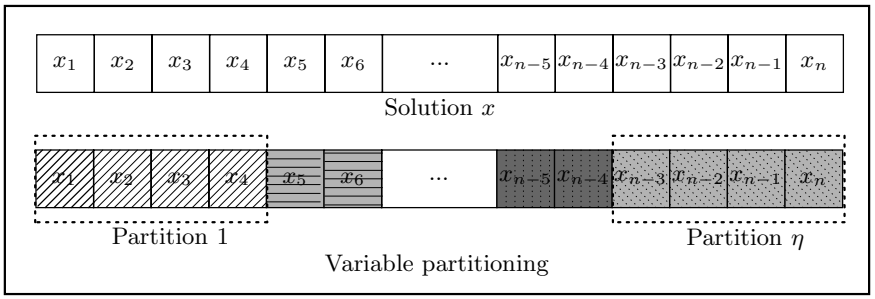

Fig. 2. Variable partitioning

3.2 The Description of the SVP Method

The SVP method scenario is described as follows. SVP starts with a random initial solution x^0. The solution is divided into η partitions, where $\eta = n/\xi$. In order to generate a trail solutions, SVP method uses two type of iterations as follows.

- **SVP inner loop**
  SVP uses a given number of iterations in order to generate a trail solutions by applying the simplex Nelder-Mead method as shown in Algorithm 1 in a random selected partitions. The number of the selected partitions is depend on the number of the applied iterations in the inner loop. The higher the number of the applied iterations, the higher the value of the cost function.
- **SVP outer loop**
  The main termination criterion in SVP is the number of the external loop which is the maximum number of iterations. The number of the best overall solutions is depends on the value of the external iterations.

Finally a local search method is applied as a final intensification process in order to refine the best solution N_{elite} which is obtained in the previous search stage. The structure of the SVP method with the formal detailed description is given in Algorithm 2, all variables of Algorithm 2 and it's definitions are reported in Table 2.

4 Numerical Experiments

In order to test the efficiency of the proposed SVP method and present the comparison results between it and other competing methods. SVP uses three sets of instances, LM-HG1, LM-HG2, CEC05 instances. The LM-HG1 instances consist of 10 uni-model and shifted functions, these functions have been used by Hvttum and Glover [9]. The LM-HG2 instances consist of 16 multi-model and shifted functions, these functions are based on functions in Laguna and Marti [11]. The CEC05 instances consist of 12 multi-model function based on classical

Algorithm 2. The proposed SVP method

Generate an initial solution x^0 randomly.
Set initial values for ξ, η, $Maxitr$, N_{elite}.
Set $x = x^0$.
repeat
  $I := 0$. ▷ Counter for inner loop.
  $K := 0$. ▷ Counter for outer loop.
  repeat
    Divide the solution x into η partitions,
    where $\eta = n/\xi$.
    Pick a random partition ϕ from η partitions of x.
    Apply Nelder-Mead Algorithm into the selected
    partition ϕ, as shown in **Algorithm 1** .
    Generate neighborhood trail solutions around x.
    $I := I + 1$.
  until $I \leq \eta$.
  Set $x\prime$ equal to the best trial solution.
  if $f(x\prime) \leq f(x)$ **then**
    $x = x\prime$. ▷ Accept the trial solution.
  end if
  $K := K + 1$.
until $K \leq Maxitr$. ▷ Stoping criteria satisfied.
Apply local search method starting from N_{elite} on the best solution in the previous stage. ▷ Intensification search.

function after applying some modifications (shifted, rotated, biased and added), these instances are described in detail in Suganthan [15]. The objective function values of all sets of instances are known. All details about the three mentioned sets of instances are described in [8]. The names and main feature of LM-HG1, LM-HG2, CEC05 functions are summarized in Table 4, Table 5 and Table 10, respectively. The SVP method was programmed in MATLAP. The results of the SVP method and the other competing benchmark methods are averaged over 10 runs. The parameter setting of the SVP method are reported in Table 3. The numerical results of all LM-HG1 instances are reported in Tables 6 - 8 where the numerical results of LM-HG2 and CEC05 instances are reported in Table 9, Table 11 respectively.

Table 3. Parameter setting

Parameters	Values
ξ	4
η	n/ξ
$Maxitr$	$n/4$
N_{elite}	1

Table 4. LM-HG1 test functions

f	Function name	f	Function name	f	Function name
f_1	Branin	f_5	Zakharov	f_9	Stair-Ros
f_2	Booth	f_6	Trid	f_{10}	Stair-LogAbs
f_3	Matyas	f_7	Sum Squares		
f_4	Rosenbrock	f_8	Sphere		

Table 5. LM-HG2 test functions

g	Function name	g	Function name	g	Function name	g	Function name
g_1	B2	g_5	Beale	g_9	Perm(o.5)	g_{13}	Powell
g_2	Easom	g_6	SixH.C.Back	g_{10}	Perm(10)	g_{14}	Dixon&Price
g_3	Gold.&Price	g_7	Schhwefel	g_{11}	Rastrigin	g_{15}	Levy
g_4	Shubert	g_8	Colville	g_{12}	Griewank	g_{16}	Ackley

4.1 Performance Analysis

In this subsection we analyze the performance of the SVP method as follows.

The Efficiency of Variables Partitioning. We compared the SVP method with variable partitioning process, with the basic simplex Nelder-Mead method (BSNM) in order to check the efficiency of variable partitioning process. The same parameters and termination criteria are used in both methods. The results are shown in Figure 3. In Figure 3, the dotted line represents the results of the basic simplex Nelder-Mead method, the solid line represents the results of the SVP method. Two functions f_3, f_7 are selected with dimensions 512, 1000 by plotting the number of iterations versus the function values. Figure 3 shows that the function values are rapidly decreases as the number of generations increases for the SVP method results than those of the basic simplex Nelder-Mead method.

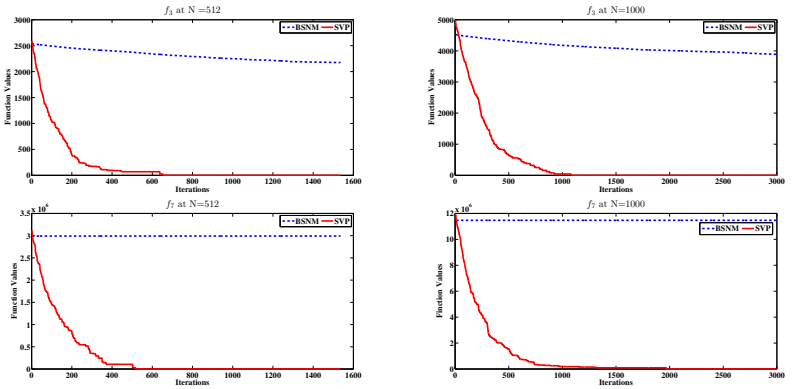

Fig. 3. Basic simplex Nelder-Mead algorithm Vs. SVP algorithm

The Performance of Final Intensification. The SVP method uses the MATLAB function "fminunc" as a local search method in the final intensification process. The final intensification can accelerate the convergence in the final stage instead of letting the algorithm running for several iterations without much significant improvement of the objective function values as shown in Figure 4. Figure 4 represents the general performance of the SVP method and the effect of the final intensification by selecting two functions f_2, g_{12} with different properties and plotting the values of objective functions versus the number of iterations. Figure 4 shows that the objective values are decreases as the number of function iterations increases. The behavior in the final intensification phase is represented in Figure 4 by dotted lines, in which a rapid decrease of the function values during the last stage is clearly observed.

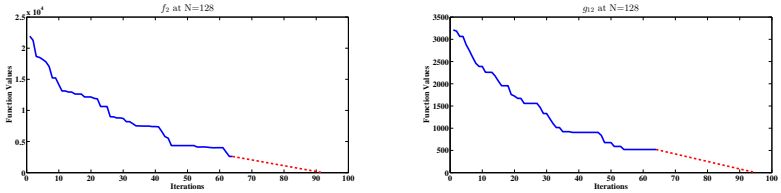

Fig. 4. The effects of final intensification process

The Performance of the Final Local Search Method. The performance of the SVP method without applying the last final intensification process on two functions f_1, f_8 with dimensions 512, 1000 is shown in Figure 5. Figure 5 represents the number of iterations versus the function values. The function values in Figure 5 are rapidly decreased and reached to its objective values without applying the final intensification process. We can conclude from Figure 5 that the MATLAB function "fminunc" is not doing the majority work of the SVP method.

4.2 The Proposed SVP Method and the Other Competing Methods

The SVP method is compared with two scatter search methods, whereas the two scatter methods were recently developed by [9] for solving high dimensional problems. The first scatter search method is called scatter search with randomized subset combination (SSR), where the second scatter search method is called scatter search with clustering subset combination (SSC). Also we compare our SVP method with a combination between the main two scatter search methods and six direct search methods. The SVP method uses the same termination criteria such as the gap between a heuristic solution x and the optimal solution x^* is $\mid f(x) - f(x^*) \mid < \epsilon$, where $\epsilon = 10^{-8}$, the other termination criterion is the maximum number of function evaluation is 50,000. The main local search methods are listed as follows.

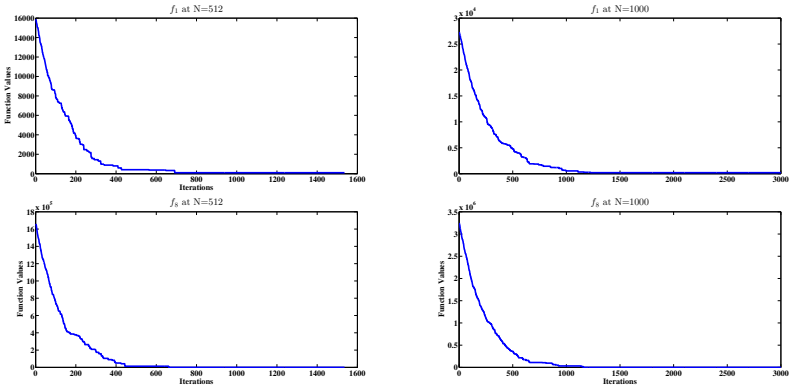

Fig. 5. The performance of the SVP method without applying final local search method

- **NM** Nelder-Mead simplex methods
- **MDS** Multi-directional search
- **CS** Coordinate search method
- **HJ** Hook and Jeeves method
- **SW** Solis and Wet's algorithm
- **ROS** Rosenbrock's algorithm

The description of these methods are reported in [8].

Comparison Results on LM-HG1 Instances. The SVP method was tested on the LM-HG1 instances in Havattum and Glover [9]. The dimensions of functions are 16, 32, 64, 128, 256 and 512. Each execution is repeated 10 times with different initial solutions. All tested methods have the same maximum number of function evaluations (50,000). We report the largest dimension n for which the method successfully found the optimal solution in all 10 runs. The comparison results between the SVP method and the other methods for all LM-HG1 instances are reported in Table 6. The best results are highlighted in bold. Table 6 shows that the SVP method outperforms in most of all functions except f_9. Also we can reach to the global minimum of these instances for $n > 512$ by increasing the number of function evaluation values as shown in Table 8. In case of at least one run fails to reach to the global minimum of the function, the ratio of successful run is recorded in parentheses.

Table 6 shows the comparison results between the SVP method and two scatter search methods. Now we evaluate the SVP method with the combined scatter search (SS) method and the direct search method as a improvement methods (IM). In addition to the combined SS method with the improvement direct search methods, there are extra two methods. SS method which is the scatter search function without improvement methods and STS method which is a hybrid scatter tabu search method. All details of STS method are reported in [2]. Table 7

Table 6. Largest successful dimension for SVP and other improvement methods on the LM-HG1 instances

f	SSR	SSC	SVP
f_1	**512**	**512**	**512**
f_2	**512**	**512**	**512**
f_3	256	256	**512**
f_4	32	**64**	**64**
f_5	32	32	**64**
f_6	16	16	**128**
f_7	**512**	**512**	**512**
f_8	**512**	**512**	**512**
f_9	32	**64**	32
f_{10}	**512**	**512**	**512**

Table 7. Largest successful dimension for the SVP method and the other combined SS with the improvement methods on the LM-HG1 instances

f	SS	STS	SS+NM	SS+MDS	SS+SW	SS+HJ	SS+ROS	SS+CS	SS+SSC	SS+SSR	SVP
f_1	2	16	4	4	8	8	16	16	16	16	**512**
f_2	2	8	4	4	16	128	8	128	128	128	**512**
f_3	2	16	8	4	32	128	32	256	256	256	**512**
f_4	0	4	2	2	2	4	4	4	4	2	**64**
f_5	2	8	4	4	8	16	8	16	8	8	**32**
f_6	2	8	4	4	8	32	8	16	16	16	**32**
f_7	2	16	4	4	16	32	32	32	32	32	**512**
f_8	2	32	4	4	64	32	64	32	32	32	**512**
f_9	0	4	2	2	2	4	2	2	4	2	**32**
f_{10}	0	4	0	0	0	128	2	128	256	256	**512**

Table 8. Function Evaluations (feval) of the SVP method with 1000 Dimension

f	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}
feval	112,949	70,802	61,357	(0.0)	(0.0)	545,698	196,552	56,253	(0.0)	203,485

shows that the SVP method is outperforms all combined scatter search with the improvement methods and the hybrid scatter tabu search method in all LM-HG1 instances.

Comparison Results on LM-HG2 Instances. Now we test the performance of the SVP method with the multi-model functions in LM-HG2 sets. The committing methods is the same methods in Table 7. The terminations criteria is the same as in the LM-HG1 functions. The results in Table 9 show that the performance of the SVP method in the LM-HG2 is better then the other competing methods, although the functions in LM-HG2 seem to be more difficult and only smaller dimensions can be solved as seen in functions g_2, g_8, g_9, g_{10}, g_{11}, g_{16}. The best results are highlighted in bold.

Table 9. Largest successful dimension for the SVP method and the other combined SS and improvement methods on the LM-HG2 functions

g	SS	STS	SS+NM	SS+MDS	SS+SW	SS+HJ	SS+ROS	SS+CS	SS+SSC	SS+SSR	SVP
g_1	0	2	2	2	4	8	4	16	16	**32**	**32**
g_2	0	2	2	2	2	2	4	2	**4**	**4**	**4**
g_3	0	8	2	2	4	8	8	8	8	8	**16**
g_4	0	8	0	2	4	8	2	2	2	2	**16**
g_5	2	8	4	4	4	8	**16**	**16**	8	**16**	**16**
g_6	2	16	2	4	16	32	16	32	**64**	**64**	**64**
g_7	0	8	2	2	2	4	0	2	2	2	**128**
g_8	0	4	0	0	4	**8**	4	**8**	**8**	4	4
g_9	2	**4**	2	2	2	2	2	2	2	2	**4**
g_{10}	0	4	2	2	4	4	4	4	4	4	**8**
g_{11}	0	**16**	0	0	2	4	2	2	2	2	4
g_{12}	0	0	0	0	0	2	2	2	0	32	**256**
g_{13}	4	16	4	4	16	64	16	64	128	128	**256**
g_{14}	2	8	2	2	8	8	8	8	16	16	**32**
g_{15}	2	256	4	4	16	128	128	**512**	128	64	64
g_{16}	0	8	0	0	0	**32**	8	**32**	**32**	**32**	4

5 CEC05 Benchmark Test Instances

In order to evaluate the proposed SVP method, a new set of a modified benchmark functions with various properties provided by CEC2005 special session [15] are used in this section. Most of these functions are the shifted, rotated, expanded, and combined variants of the classical functions. These modifications make them to be more hard, and resistant to simple search. We used 12 functions h_1-h_{12} as shown in Table 10. Specifically, these functions span a diverse set of problem features, including multi-modality, ruggedness, noise in fitness, ill-conditioning, non-separabilty, interdependence(rotation), and high-dimensionality. Most of these functions are based on classical benchmark functions such as Rosenbrocks, Rastrigins, Swefels, Griwank and Ackleys function. The applied maximum evaluation function value as a termination criterion is increased to 100,000. Table 11 shows the CEC2005 functions h_1 - h_{12} with its objective function values for 10-100 dimensions. The exact global minimum is highlighted in bold. Results in Table 11 show that the SVP method obtains the exact global minimum of functions h_1 and h_2 at 100 dimension, where obtained the exact global minimum for function h_7, h_{12} at 80 dimension and for function h_6, h_{10} the exact global minimum at 40, 20 dimensions respectively. For functions h_3, h_4, h_5, h_8, h_{11} the SVP method fails to obtain the exact global minimum for any dimension and the obtained function values of these functions are reported in Table 11.

Table 10. CEC05 benchmark test functions

h	Function name	Bounds	Global minimum
h_1	Shifted Sphere	[-100,100]	-450
h_2	Shifted Schwefel's 1.2	[-100,100]	-450
h_3	Shifted rotated high conditioned elliptic	[-100,100]	-450
h_4	Shifted Schwefel's 1.2 with noise in fitness	[-100,100]	-450
h_5	Schwefel's 2.6 with global optimum on bounds	[-100,100]	-310
h_6	Shifted Rosenbrock's	[-100,100]	390
h_7	Shifted rotated Griewank's without bounds	[0,600]	-180
h_8	Shifted rotated Ackley's with global optimum on bounds	[-32,32]	-140
h_9	Shifted Rastrigin's	[-5,5]	-330
h_{10}	Shifted rotated Rastrigin's	[-5,5]	-330
h_{11}	Shifted rotated Weierstrass	[-0.5,0.5]	90
h_{12}	Schwefel's 2.13	[-100,100]	-460

Table 11. Mean number of function values with 10∼ 100 dimensions

h	10	20	30	40	50	60	70	80	90	100
h_1	**-450**	**-450**	**-450**	**-450**	**-450**	**-450**	**-450**	**-450**	**-450**	**-450**
h_2	**-450**	**-450**	**-450**	**-450**	**-450**	**-450**	**-450**	**-450**	**-450**	**-450**
h_3	9235.12	6432.56	5429.27	4312.7	7134.2	5426.3	3458.45	4869.57	6778.24	4879.125
h_4	7143.61	52,848.2	72,582.4	86,931	315,056	370,425	416,241	504,957	565,784	590,668
h_5	194,431	201,456	293,84	312,58	614,58	640,15	714,26	785.12	798,147	812,58
h_6	**390**	**390**	**390**	**390**	465.23	485.45	486.78	512.56	540.25	545.13
h_7	**-180**	**-180**	**-180**	**-180**	**-180**	**-180**	**-180**	**-180**	-179.993	-179.78
h_8	-120	-120	-120	-120	-120	-120	-119.99	-119.98	-119.99	-119.98
h_9	**-330**	**-330**	**-330**	**-330**	512.716	643.05	714.95	740.12	815.56	845.89
h_{10}	**-330**	**-330**	-237.47	-215.45	112.48	145.26	242.23	361.68	412.25	415.23
h_{11}	101.56	120.731	123.908	126.56	131.25	152.36	158.69	162.54	165.98	167.57
h_{12}	**-460**	**-460**	**-460**	**-460**	**-460**	**-460**	**-460**	**-460**	-457.15	-441.25

6 Conclusion

In this paper, the simplex Nelder-Mead method which is called SVP (simplex variable partitioning) has been proposed to solve large scale optimization problems. The use of variable partitioning process assists effectively the SVP method to achieve promising performance specially with high dimensional test functions. Moreover, applying the Nelder-Mead method in different partitions helps the SVP method to achieve a wide exploration and a deep exploitation before stopping the search by generating a trail solution around the iterate solution. Finally the intensification process has been inlaid in the SVP method to accelerate the search process. The numerical experiments on 38 test benchmark functions with different properties have been presented to show the efficiency of the proposed SVP method. The comparison with other competing methods indicate that the SVP method is promising and it is cheaper than other methods.

References

1. Conn, A.R., Gould, N.I.M., Toint, P.L.: Trust-region methods. MPS-SIAM series on optimization. SIAM, Philadelphia (1987)
2. Duarte, A., Mart, R., Glover, F., Gortazar, F.: Hybrid scatter tabu search for unconstrained global optimization. Ann. Oper. Res. 183, 95–123 (2011a)
3. Garcia, S., Lozano, M., Herrera, F., Molina, D., Snchez, A.: Global and local real-coded genetic algorithms based on parentcentric crossover operators. Eur. J. Oper. Res. 185, 1088–1113 (2008)
4. Hedar, A., Fukushima, M.: Hybrid simulated annealing and direct search method for nonlinear unconstrained global optimization. Optim. Methods Softw. 17, 891–912 (2002)
5. Hedar, A., Fukushima, M.: Minimizing multimodal functions by simplex coding genetic algorithm. Optim. Methods Softw. 18, 265–282 (2003)
6. Hedar, A., Fukushima, M.: Directed evolutionary programming: towards an improved performance of evolutionary programming. In: Proceedings of Congress on Evolutionary Computation, CEC 2006, IEEE World Congress on Computational Intelligence, Vancouver, Canada, pp. 1521–1528 (July 2006)
7. Hedar, A., Ali, A.F.: Tabu search with multi-level neighborhood structures for high dimensional problems. Appl. Intell. 37, 189–206 (2012)
8. Hvattum, L.M., Duarte, A., Glover, F., Mart, R.: Designing effective improvement methods for scatter search: an experimental study on global optimization. Soft Comput 17, 49–62 (2013)
9. Hvattum, L.M., Glover, F.: Finding local optima of high dimensional functions using direct search methods. Eur. J. Oper. Res. 195, 31–45 (2009)
10. Kolda, T.G., Lewies, R.M., Torczon, V.J.: Optimization by direct search: new perspectives on some classical and modern methods. SIAM Rev. 45, 385–482 (2003)
11. Laguna, M., Mart, R.: Experimental testing of advanced scatter search designs for global optimization of multimodal functions. J. Glob. Optim. 33, 235–255 (2005)
12. Linhares, A., Yanasse, H.H.: Search intensity versus search diversity: a false trade off? Appl. Intell. 32, 279–291 (2010)
13. Nelder, J.A., Mead, R.: A simplex method for function minimization. Comput. J. 7, 308–313 (1965)
14. Mladenovic, N., Drazic, M., Kovac, V., Angalovic, M.: General variable neighborhood search for the continuous optimization. Eur. J. Oper. Res. 191, 753–770 (2008)
15. Suganthan, P.N., Hansen, N., Liang, J.J., Deb, K., Chen, Y.-P., Auger, A., Tiwari, S.: Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization. Technical report, Nanyang Technological University, Singapore (2005)