Statistika, Vol. 9 No. 2, 99 – 107
Nopember 2009

A Co-joint Deterministic Search Direction Sampling Procedure with Probabilistic Soft Approach

ISMAIL BIN MOHD[1], GOH KHANG WEN[2], TAN EE LING[3]

[1] Department of Mathematics, Faculty of Science and Technology, Universiti Malaysia Terengganu, 21030 Kuala Terengganu, Terengganu, Malaysia. Email: ismailmd@umt.edu.my
[2] Department of Physical and Mathematical Science, Faculty of Science, Universiti Tunku Abdul Rahman, Perak Campus, Jalan Universiti, Bandar Barat, 31900 Kampar, Perak, Malaysia.
Email: gohkw@utar.edu.my
[3] Department of Mathematics, Faculty of Science and Technology, Universiti Malaysia Terengganu, 21030 Kuala Terengganu, Terengganu, Malaysia. Email: eling_tan@hotmail.com

ABSTRACT

Since hard continuous optimization models contain more than one solution and even continuum solution, it is impossible to seek all the solution by using the existent optimization methods. Therefore, in this paper we introduce a co-joint deterministic and probabilistic approach which modifies a soft approach for solving hard continuous optimization models. An algorithm of co-joint approach and several numerical experiments have been presented in this paper. The special numerical test results have shown that the co-joint approach is more effective than soft approach algorithm. Fortunately, we have found that the co-joint algorithm can be used to determine whether the optimization model is hard continuous optimization models or not.

Keywords: Hard continuous optimization, Deterministic and Probabilistic Approach, Steepest Descent Method.
2000 Mathematics Subject Classification: 90C29, 68Q10, 49M05

1. INTRODUCTION

In order to solve the daily life problem, normally a mathematician will firstly convert the problem into a mathematical models which might contain more than one solution or even continuum solution. Therefore, the mathematician will seek the best solution for the problem.

There are several known methods which have been designed for obtaining the solutions of the optimization problems such as Newton's method, steepest descent method, quasi-Newton method and others, but the said methods will obtain one solution for each searching if the right starting point is used. There are alot of mathematical models where its objective function is not differentiable at many points and there are many local or even infinite number of local minima as seeing in Example 1 and Example 2 of [5].

Xu and Ng (2006)[5] have introduced a method so-called soft approach for solving the above mentioned problems. In this paper, we will proposed a modification of their approach which can handle some of the disadvantages of their approach and to make this proposed method more attractive, we will introduce some idea obtained from deterministic as well as probabilistic approaches.

2. OPTIMIZATION PROBLEMS

The optimization problem to be considered in this paper is to find a good enough solution set of the no assumption about the objective function $\boldsymbol{f} : \boldsymbol{D} \subseteq \boldsymbol{R}^{\boldsymbol{n}} \to \boldsymbol{R}$ subject to

$$\boldsymbol{g_i}(\boldsymbol{x}) \le 0 , \quad \boldsymbol{i} = 1,\ldots,\boldsymbol{m} . \tag{2.1}$$

Traditionally, we would like to search an optimal solution $\boldsymbol{x}^* \in \boldsymbol{X}_{\boldsymbol{f}}$ where

$$X_f = \{x : g_i(x) \le 0, i = 1,\ldots,m\}. \tag{2.2}$$

We know that only problems with some good analytical structures can be solved properly. Therefore, we need to device a very good method which can be used to solve the optimization models with not good analytical structures as well as the good one as be seen in this paper.

3. A XU-NG'S SOFT APPROACH

In order to handle the time limit requirement for solving a larger class of optimization problems, a technique so-called soft approach has been introduced in (Xu and Ng, 2006) which can be summarized as follows.

Algorithm 3.1 (Xu and Ng, 2006)

Briefly, this algorithm consists of three stages as follows.

Stage 1:
Any descent method can be used for obtaining the solution.

Stage 2:
Determine the suitable taken number of samples. The solution obtained in stage 1 can be used as initial point for sampling.

Stage 3:
Select the good enough set like cone from the samples which have been taken in Stage 2.

In order to observe the capability of Algorithm 3.1, according to our reading of their paper, we implement this algorithm as given in the following algorithm.

Algorithm 3.2 (Xu and Ng, 2006)

```
1. Find $x^{(0)} \in \mathbf{int}(X_f)$ uniformly, let $i = 0$ ! int means interior
2. Select a direction $d$ uniformly over $D = \{d \in R^n : \|d\| = 1\}$
3. Find two hit points $P^{(i)}$ and $Q^{(i)}$ on $b(X_f)$ from $x^{(i)}$ along $d$ and $-d$ . ! $b(X_f)$ means
   boundary of $X_f$ .
4. Choose $x^{(i+1)}$ uniformly by using $x^{(i+1)} = x^{(i)} + \mu(P^{(i)} - Q^{(i)})$
   where $\mu$ is a random variable distributed uniformly over [0,1].
5. $i := i + 1$ go to Step 2 till necessary samples are obtained.♦
```

4. STEEPEST DESCENT APPROACH

The classical steepest descent method which is designed by Cauchy (1847) ([1][2, Section 1]) can be considered among the most important procedures for minimization of real-valued function defined on R^n. The steepest descent step size appeared in

$$x_{k+1} = x_k + \lambda_k d_k, \tag{4.1}$$

with the $d_k = -\nabla f(x_k)$, and the step size λ_k is obtained using exact line search ([1] [2])

$$\lambda_k = \frac{g_k g_k}{g_k A g_k}, \tag{4.2}$$

where $g_k = \nabla f(x_k)$ and $A = \nabla^2 f(x_k)$. The algorithm of the classical steepest descent is as follows.

Algorithm 4.1 (Steepest Descent)

Data : $x_0 \in R^n$, $\varepsilon \in R$, and $f : R^n \to R$.

```
1. k = 0
2. $d_k = -\nabla f(x_k)$
3. if $|d_k| \le \varepsilon$ do
```

```
        3.1. $x_k$ is the minimizer
        3.2. **stop**
4.  Solve $\lambda_k = \frac{g_k g_k}{g_k A g_k}$, where $g_k = \nabla f(x_k)$ and $A = \nabla^2 f(x_k)$.
5.  $x_{k+1} = x_k + \lambda_k d_k$
6.  **if** $\|x_{k+1} - x_k\| \le \varepsilon$
    **then**
        6.1. $x_k$ is the minimizer
        6.2. **stop**
             **else**
        6.3. $k = k + 1$
7.  **return.**♦
```

5. DETERMINATION OF SAMPLE SIZE

Suppose that S is a sample set which contains at least one good enough solution with high probability. Let $|S|$ the cardinality of set S, denotes the number of sample points in set S.

In our algorithm, $|S|$ the smallest number of necessary samples from which we can obtain the set of good enough solutions, is derived as follows.
Suppose that the solution should be one of the top $k\%$ alternatives with a probability not smaller than $q\%$, then $|S|$ can be obtained by solving

$$P\{|S \cap G| \ge 1\} = 1 - (1 - k\%)^{|S|} = q\% \tag{5.1}$$

where G is a set of good enough solutions [3].
For example if $k = 1$, and $q = 99.99$, then by (5.1) the minimal number of samples is $|S| = \frac{\ln(1 - 1\%)}{\ln(1 - 99.99\%)} = 917$,
but normally we take 1000 instead.

6. SEARCH DIRECTION

Our search direction is determined by an integer $i \in \{1,\dots,m\}$ in a unit vector defined by

$$d^{(i)} = \left(\cos\left(\frac{2\pi i}{m}\right) \quad \sin\left(\frac{2\pi i}{m}\right) \right)^T, \quad (i = 1,\dots,m) \tag{6.1}$$

where

$$m = \min\left\{ a \in Z^+ : a + a^2 + a^3 + \ldots + a^{n_g} \ge |S| \right\} \tag{6.2}$$

in which S is described in Section 5, Z^+ is a set of positive integers, and $n_g \in Z^+$ is the number of generation. But in our studies, we set the $n_g = 3$ which means the searching process will end after third generation.
The vector $d^{(i)}, (i = 1,\dots,m)$ is used to generate a sample point which belongs to S as describe in next section.

7. SAMPLING TECHNIQUE

Suppose that in 2-dimensional case, starting with the point $x^{(0)}$, we wish to generate a set of four (=m) points denoted by ♦ one for each direction d, and from each of these points ♦ we generate other four (=m) points denoted by • and for the next generation, the same procedure is carried out. Therefore for three generations, starting with $x^{(0)}$, we obtain $4+4^2+4^3=84$ points as shown in Fig. 7.1.

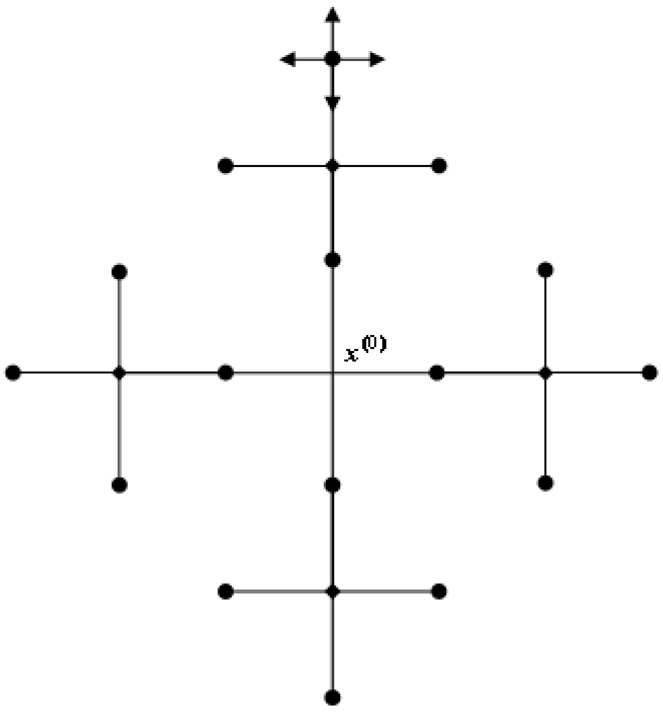

Figure 7.1

Starting with point $x^{(1,0)} = x^{(0)}$, for m = 4, our mesh points which are denoted by its indices for each generation are given as follows.

First generation : $(1,1)$, $(2,1)$, $(3,1)$, $(4,1)$

Second generation : $(1,2)$, $(2,2)$, $(3,2)$,..., $(15,2)$, $(16,2)$

Third generation : $(1,3)$, $(2,3)$, $(3,3)$, ..., $(28,3)$, $(29,3)$, $(30,3)$,..., $(63,3)$, $(64,3)$

Since, for example, m = 4, the points $(1,2)$, $(2,2)$, $(3,2)$, $(4,2)$ are obtained by using the point represented by $(1,1)$ indices, in the direction $d^{(1)}$, $d^{(2)}$, $d^{(3)}$, $d^{(4)}$ respectively through the formula $x^{(i,2)} = x^{(1,1)} + d^{(i)}$ $(i = 1,2,3,4)$.

For obtaining all the points produced by the method visualized in the Fig. 7.1 detaily, the following algorithm can be used.

Sample($n_g, m, n \in Z^+, d \in R^{mxn}; x \in R^{|S|xn}$)

This procedure calculates the $|S|$ number of sample points where

$$|S| = \frac{m(m^{n_g} - 1)}{m - 1}$$

and m is the number of directions. In this procedure, n_g and d are the input parameter, and x is the input-output parameter.

```
1. $x^{(1,0)} = x^{(0)}$
2. for g = 1 to $n_g$ do
   2.1. j = 1
   2.2. for i = 1 to $m^{g-1}$ do
        2.2.1. for k = 1 to m do
               2.2.1.1. $x^{(j,g)} = x^{(i,g-1)} + d^{(k)}$
               2.2.1.2. j = j+1
3. return.♦
```

As alternative, we can use the followng algorithm for computing the sample points.

Algorithm 7.1 (Alternative)

Data : $x^{(0)}$, $n \in Z^+$

```
1. $x^{(1,0)} = x^{(0)}$
2. for j = 1 to n do
   2.1. for i = 1 to $m^j$ do
        2.1.1. a = quotient (i÷m)
        2.1.2. b = remainder (i÷m)
        2.1.3. $x^{(i,j)}$ = if b = 0
                         then $x^{(a,j-1)} + d^{(m)}$
                         else $x^{(a+1,j-1)} + d^{(b)}$
3. return.♦
```

8. A COJOINT APPROACH

In order to device a method which can handle the structure of the objective function and contol the time needed for solving (depending on the sample size), beside contains the same ideas given in ([5]), our method so-called CA method which is obtained by combining the ideas from Xu-Ng's approch ([5]), steepest decent method ([2]), deterministic and probabilistic approaches and also some modifications of Xu-Ng's algorithm, can be arranged in the three stages as follows.

Stage 1

Obtain the solution point x^* by using the steepest descent approach (Algorithm 4.1).

Stage 2

Determine the needed taken number of samples using

$$P\{|S \cap G| \geq 1\} = 1 - (1 - k\%)^{|S|} = q\% \tag{8.1}$$

where k is the highest percentage, noted that the problem has been solved completely, q is the probability that the sample which has been determined in the top $k\%$ solutions, $G \subseteq X_f$ is a set of good enough solutions, and $|S|$ is the cardinality of S.

Stage 3

Use the uniform sampling algorithm for samples collection.

Stage 4

Select the good enough set defined by

$$G = \left\{ x : \left| f(x) - f(x^{(0)}) \right| \leq \varepsilon \right\} \tag{8.2}$$

from the samples collection in Stage 3.

The modified sampling algorithm which used in our CA method to replace the uniform sampling used by Xu and Ng (2006) in stage 3 have shown in Algorithm 8.1.

Algorithm 8.1 (Stage 3)

Data: $x^{(0)} \in \boldsymbol{R}^n$ is the solution from Algorithm 4.1 (Stage 1), the even $|\boldsymbol{S}|$ is the needed number of samples.

```
1. case true of
   1.1. $n = 1$ ! Dimension 1
        1.1.1. $i = 1$
        1.1.2. $|S| = |S| \div 2$
        1.1.3. while ($i \le |S|$) do
               1.1.2.1. $d = 1$; $\lambda = 0.0001$! Within the domain
               1.1.2.2. $x^{(i)} = x^{(0)} \pm i\lambda d$
               1.1.2.3. $i = i + 1$
        1.1.4. $x^{(i)} \to S$
        1.1.4. stop
   1.2. $n = 2$ ! Dimension 2
        1.2.1. $m = \min(a \in Z^+ : a^{n_g} + ... + a^2 + a \ge S)$ ! the number of searches
        1.2.2. $d^{(i)} = \left( \cos\left(\frac{2\pi i}{m}\right) \quad \sin\left(\frac{2\pi i}{m}\right) \right)^T, \quad (i = 1, ..., m)$
        1.2.3. S = Sample ($n_g, m, n, d$ ; $x^{(0)}$) ! We also can use Algorithm 7.1
        1.2.4. stop
   1.3. default
        1.3.1. write "error in n"
        1.3.2. stop ! n is not equal to 1 or 2
2. check whether S satisfies (8.2) or not
3. return.♦
```

9. SPECIAL TESTING EXAMPLES

In order to observe the behaviour of both cojoint and soft methods for the objective function with the continuum solutions, we have used two examples which are given in [5], and the results for the comparison are given in Table 9.1.

Example 9.1 : Find the minimum of

$$f(x) = \begin{cases} \dfrac{x\left|\sin(1/x)\right|}{x^2+1} & x \ne 0 \\ 0 & x = 0 \end{cases}$$

in the interval [-2,2].

The minimizer is $x^{(0)} = -0.7374$ with $f(x^{(0)}) = -0.4667$. According to [5], it is enough to take 1000 uniform samples from $X_f = [-2,2]$ for possibility that at least one point in G is one of the top 1% alternatives with a probability not smaller than 99.99% in the samples set as refer to (5.1).

In order to illustrate the effectiveness of their method ([5]), this problem has been solved 50 times and got one good enough point each time.

In our approach, when we take 1000 uniform samples from $X_f = [-2,2]$, we observe that at least 135 samples contained in G.

Example 9.2 : Find the minimum of

$$f(x_1, x_2) = -\frac{|\sin(5\pi\ x_1) + \sin(5\pi\ x_2)|}{e^{10(x_1-0.25)^2+10(x_2-0.5)^2}}$$

in the feasible set given by

$$\{(x_1, x_2) \in [0,1]\times[0,1] : 0 \le 1 - x_1 - x_2 \le 1, x_1 + 2x_2 \ge 1\}.$$

The minimizer is $x^{(0)} = (0.120765, 0.5)$ with $f(x^{(0)}) = -1.64776$. According to ([5]), it is enough to take 9091 uniform samples from $X_f = [-2,2]$ for possibility that at least one point in G is one of the top 0.1% alternatives with a probability not smaller than 99.998% in the sample set as refer to (5.1). But in our studies, we take 9723 samples which take 21 search directions in 3 generation searching process.

In our approach, when we take 9723 uniform samples from $X_f = [-2,2]$, we observe that at least 3885 samples contained in G.

The comparison of the results between our studies and ([5]) on both examples in this section is shown in Table 9.1.

Table 9.1 : The numerical results for both methods

Example	Xu and Ng soft approach	Ismail and Ling soft approach
1	Take 1000 samples. Test and select 50 samples only as the good enough set $G = \{x : \lvert f(x) - f(x^*)\rvert \le 1\times10^{-4}\}$	Take 1000 samples The good enough set contains 135 samples $G = \{x : \lvert f(x) - f(x^*)\rvert \le 5\times10^{-5}\}$
2	Take 9091 samples Test and select 50 samples only as the good enough set $G = \{x : \lvert f(x) - f(x^*)\rvert \le 0.05\}$	Take 9723 samples The good enough set contains 3885 samples $G = \{x : \lvert f(x) - f(x^*)\rvert \le 0.005\}$

10. OTHER NUMERICAL RESULT

On other hand, we have tested the modified algorithm to some other test examples ([2][4]) as follows.

1. $f(x) = \sin(x) + \sin\left(\frac{2x}{3}\right)$, $x^0 = (4.3)$
2. $f(x_1, x_2) = 2x_1 + 3x_2$, $x^0 = (5,5)$
3. $f(x_1, x_2) = 4x_1 + x_2 - x_1^{\,2}x_2$, $x^0 = (1,1)$
4. six hump camel back function
   $f(x_1, x_2) = 4x_1^{\,2} - 2.1x_1^{\,4} + \frac{x_1^{\,6}}{3} - x_1x_2 - 4x_2^{\,2} + 4x_2^{\,4}$, $x^0 = (-1.6, 0.9)$
5. Three hump camel back function
   $f(x_1, x_2) = 2x_1^{\,2} - 1.05x_1^{\,4} + \frac{x_1^{\,6}}{6} - x_1x_2 + x_2^{\,2}$, $x^0 = (-1.6, 0.9)$
6. Booth Function
   $f(x_1, x_2) = (x_1 + 2x_2 - 7)^2 + (2x_1 + x_2 - 5)^2$, $x^0 = (0.4, 1.6)$

7. $f(x_1,x_2)=\frac{x_1^4}{4}-\frac{x_1^2}{2}+0.1x_1+\frac{x_2^2}{2}$, $x^0=(-1,1)$
8. $f(x_1,x_2)=(2x_1^3x_2-x_2^3)^2+6(x_1-x_2^2+x_2)^2$, $x^0=(1,1)$
9. Two-Dimension Function
   $f(x_1,x_2)=[1-2x_2+c\sin(4\pi x_2)-x_1]^2+[x_2-0.5\sin(2\pi x_1)]^2$, $c=0.2$, $x^0=(6,-2)$
10. Two-Dimension Function
   $f(x_1,x_2)=[1-2x_2+c\sin(4\pi x_2)-x_1]^2+[x_2-0.5\sin(2\pi x_1)]^2$, $c=0.5$, $x^0=(0,0)$
11. Two-Dimension Function
   $f(x_1,x_2)=[1-2x_2+c\sin(4\pi x_2)-x_1]^2+[x_2-0.5\sin(2\pi x_1)]^2$, $c=0.05$, $x^0=(-1,1)$

Since there are a few mistakes found in the algorithm proposed in a soft approach which is introduced in [5], we cannot implement such algorithm as willing by them. Therefore, in Table 10.1, we can only display our results which are obtained by implementing our approach to the testing examples given in this section.

Table 10.1

Example	**The obtained number of good enough set.**
1	$\frac{902}{1000}\times 100\% = 90.2\%$
2	$\frac{8420}{8420}\times 100\% = 100\%$
3	$\frac{8420}{8420}\times 100\% = 100\%$
4	$\frac{8420}{8420}\times 100\% = 100\%$
5	$\frac{8420}{8420}\times 100\% = 100\%$
6	$\frac{8420}{8420}\times 100\% = 100\%$
7	$\frac{8420}{8420}\times 100\% = 100\%$
8	$\frac{7947}{8420}\times 100\% = 94.38\%$
9	$\frac{8420}{8420}\times 100\% = 100\%$
10	$\frac{8420}{8420}\times 100\% = 100\%$
11	$\frac{8420}{8420}\times 100\% = 100\%$

11. CONCLUSION

By modified soft approach, we observe that the way of sampling is more effective compared to previous study on the method done by [5]. The good enough set G selected by Xu and Ng ([5]) is not too good compared with the good enough set selected by our modified soft method as shown by Example 9.1 and Example 9.2.

We also have shown that our method can be used for solving eleven examples given in Section 10, whereas the compared method cannot be used due to several mistakes found in the corresponding algorithm.

Besides solving the optimization problems given in Section 10, our modified method still can determine whether the problems have the local minimum or not.

According to the discussion given, and based on 13 testing examples, we can say that our method is superior to the soft approach proposed in [5].

REFERENCE

[1]. Cauchy A., (1847). Méthodes générales pour la resolution des systèmes déquations simultanées, *C. R. Acad. Sci. Par.,* 25, 536-538.

[2]. Goh Khang Wen and Ismail Bin Mohd, (2007), An Alternative Newton-like Exact Line Search for Steepest Descent Method. *Proceedings of 3rd International Conference on Research and Education in Mathematics 2007*, Universiti Putra Malysia, Applied Mathematics and Mathematics Education, page 222-227

[3]. Lau T.W.C., Ho Y.C., (1997), Universal alignment probabilities and subset selection for ordinal optimization, *JOTA*, 93(3), 455-489.

[4]. Tan Ee Ling and Ismail Bin Mohd, A Soft Approach for Hard Continuous Optimization, *National Research and Innovation Competition 2007 (NRIC 2007), USM, Penang, Malaysia.* 20th – 25th May 2007

[5]. Xu C. and Ng P., (2006), A Soft Approach For Hard Continuous Optimization, *European Journal of Operational Research*, 173, pp. 18 – 29.