The 7th International Symposium on Operations Research and Its Applications (ISORA'08)
Lijiang, China, October 31–Novemver 3, 2008

A Modified Pattern Search Approach for Unconstrained Optimization Problem*

Xin-Wei Lu[1] Fu-Li Rong[2]

[1]College of Science, Xi'an Jiaotong University, Xi'an, Shaanxi, 710049, China
[2]Department of Computer Science, Sun Yat-Sen University
Guangzhou, Guangdong, 510275, China

Abstract In this paper, a new modified pattern search approach combined with filter technique for unconstrained global optimization problem is proposed. The filter technique is improved first and the relevant modified algorithm is presented then. Numerical experiments show the feasibility and effectiveness of the proposed method. Especially for higher dimensional dimensional problems.

Keywords Pattern search; Filter technique; Unconstrained; Optimization.

1 Introduction

This paper discusses a new method on solving derivative-free optimization problem. Usually, people tend to solve the optimization problem according to the derivative information of the objective function. However, lots of optimization models derived from engineering field and everyday life such as industry, economics, biology, chemistry are the ones whose objective functions are underivative, or the information of the derivative are unavailable or unreliable. Thus, it's urgent to develop new strategies and methods.
Direct search method [1] is a class of derivative-free method firstly appeared in 1950s with ignorance of scholars in the following decades as the springing up of the derivative based methods. It has been revalued in recent years due to its unique advantages on solving functions whose information of derivative are unavailable or unreliable. The direct search method contains many types such as linear search, conjugate direct search, pattern search,quadratic approximation method and simplex method. Among which, the pattern search is a most popular one that received a lot of attention world wide.
Pattern search approach is a kind of very important direct search strategies, which was first describe by Box in 1950s [2] and Hooke-Jeeves [3] in the early 1960s. The main idea is to generate a sequence of iterate $x^{(k)}$ without using any information of derivatives, including gradient and second-order derivative of objective functions. It only depends on the function values. At each iteration, the objective function is evaluated at a finite number of trial points. If a point can yield a lower function value than the current one, then let it replace the current point and the iterate is called successful. Or, it is called unsuccessful and new trial points will be tested [4,5].

*This work is supported by The National Natural Science Foundation of China, NSFC(10671057)

2 The algorithm template

We consider the following optimization problem:

$$\min f(x)$$

where $f : R^n \to R$ is a continuously differentiable function whose derivative information is unavailable or unreliable.
In our approach, we combine the filter technique with pattern search method to build a new algorithm.
First of all, we introduce some useful symbols used hereinafter:
λ: mesh size
V: a base in R^n
V_+: a positive base generated from V
v_i: the member in V_+
$|V_+|$: the number of members in V_+
F: a filter
k: the kth iteration
In recent years, ordered positive bases have been introduced into pattern search method. A positive base V_+ satisfies the following properties [6]:
(1) each vector in R^n is a nonnegative combination of the members of V_+;
(2) any proper subset of V_+ is not a positive basis.
Any positive basis V_+ satisfies:

$$n+1 \leq |V_+| \leq 2n$$

Two simple and common examples are:

$$V_+ = \{v_1, v_2, ..., v_n, -\sum_{n}^{k} v_i\}$$

$$V_+ = \{v_1, v_2, ..., v_n, -v_1, -v_2, ..., -v_n\}$$

The pattern search approach tries to find any points that can produce a lower objective function value along each directions in turn. When no improved points can be found at current iterate, the grid will be fined by adjusting the mesh size and the new search begins. Nevertheless, such search strategy increases the calculation burden or even make the search not practical when the problem is a high dimensional one with a huge number of positive base. Meanwhile, the criterion also plays a key role in the success of pattern search by determining the acceptance and reject of a trail point.
Based on the discussion above, we modify the criterion of iterate acceptance first by utilizing the filter the technique [7,8].
Because the objective function value on a pattern search point is at least a local optimal value which means it's lower than all others' round it on the grid. That is, $\forall v_i \in V_+, i = 1, 2, ..., |V_+|$

$$f(x) \leq f(x + \lambda v_i)$$

Then define the filter function as follows:

$$h(x)=\begin{cases}0 & if \quad w(x)\le 0\\ w(x) & if \quad w(x)>0\end{cases}$$

where $w(x)=f(x)-\alpha\min\{f(x+\lambda v_i), i\in\Lambda\}-(1-\alpha)f^*$
where Λ is a set of members of the positive base whose included angle with the current search direction are less than 90°; f^* is the current global optimal value; $\alpha\in(0,1)$ is a positive parameter close to 1.
Furthermore, we define the filter as: a trail point can be accepted by the filter if $\forall x_j\in F^{(k)}$,it satisfies:
$h(x)<(1-\zeta)h(x_j)$ or $f(x)<f(x_j)-\delta h(x)$ (2.1)
where $\zeta,\delta\in(0,1)$ are parameters close 0.
Any point x that can be accepted by the filter is defined as a filter point in the paper. When a filter point is found the filter should be updated by deleting all the points that dominated by x [9].
We modified the filter based on the following rules:
(1)Increase the algorithm capability on solving higher dimensional problems.
(2)Balance the searching capability and calculation speed at the same time. On one side, search range is broader than before. For those points who can make a decrease on $h(x)$ not on $f(x)$ can also be the new iterate. On the other side, discard the directions along which the optimal values are unlikely to appear.
(3)Ensure the convergence.
Before we outline the algorithm, present a filter update rule first:
Rule 1 (Filter update): If a new point is accepted by the filter $F^{(k)}$, then add it into the filter and update $F^{(k)}$ to $F^{(k+1)}$ by removing all the points dominated by it in the filter.
Algorithm:
step1: Initialize $k=1$, α,ζ,δ. Let $x^{(1)}\in\Omega$ be the initial point. Choose the initial filter $F^{(1)}$ that satisfies $x^{(1)}\in F^{(1)}$. Choose positive base $V_i^{(k)}$ and the initial step length $\lambda^{(1)}$.
step 2: Choose $\lambda^{(k)}$. Set $i=1$, $p=0$; let $t^{(k)}=|V_+|$.
step 3: Compute the function values of $f(x)$ and $h(x)$ of the point $x^{(k)}+\lambda^{(k)}v_i^{(k)}\in\Omega, i\in\Lambda$.
step 4: If the trial point in the feasible region Ω can be accepted by filter $F^{(k)}$, then let it be $x^{(k+1)}$. Update the filter $F^{(k)}$ according to Rule 1 and expand $\lambda^{(k)}$, $k=k+1$, $p=0$, and goto step 6, else goto step 5.
step 5: Execute a finite search process, if there are any points in the feasible region Ω can be accepted by the filter, let it be $x^{(k+1)}$; let $p=0$ and update the filter according Rule 1. Else $p=p+1$.
step6: Let $i=i+1$. If $i>t^{(k)}$ then let $i=1$; if $p<t^{(k)}$ then goto step 2, else goto step 7.
If $i>t^{(k)}$, let $i=1$.
If $p<t^{(k)}$, then go to step 2. Else, go to step 7.
step 7: Execute a finite search process. If there are any points in the feasible region Ω can be accepted by the filter, let it be $x^{(k+1)}$, $k=k+1$, update the filter and expand $\lambda^{(k)}$. Else, contract $\lambda^{(k)}$.
step 8: If stopping conditions are not satisfied, then go to step 2. Else, output the optimal value and terminate.
Remark: 1. The choice of step length is free. But on one side, if the step length is too

small, then no obvious progress can be obtained. On the other side, if the step length is too large, the it will decrease the efficiency of methods.
2. The finite search processes in step 5 and 7 are arbitrary. The simplest choice is search for points randomly on the mesh.
3. The stopping condition is free. The normal one is $\lambda^{(k)} < \varepsilon$.
The convergence proof is similar to [7].

3 Numerical experiment

In this paper, we test the feasibility and effectiveness of the algorithm via some famous test functions. All the experiments are carried out on the computer with Intel(R) Pentium(R) M1.73G, 768M of memory using VC++ 6.0. We set $\alpha = 0.1$, $\zeta = 0.1$, $\delta = 0.1$, $\varepsilon = 10^{-7}$. The positive base is $e_1, e_2, ..., e_n, -e_1, -e_2, ..., -e_n$, where the e_i is the unit vector. We set the stopping criterion as $\lambda < \varepsilon$, and the number of iterate should less than 10^4.
The test functions (TF) in use are:
NO 1: Beale [10]: $x^{(1)} = (1,1)$
NO 2: Rosenbrock [10]: $x^{(1)} = (-1.2,1)$
NO 3: Extented Powell [10]: $x^{(1)} = (3,-1,0,1,...,3,-1,0,1)$
NO 4: Dixon [11]: $x^{(1)} = (-2,-2,...,-2)$
NO 5: Power [12]: $x^{(1)} = (1,1,...,1)$
NO 6: Trigonmetric [10]: $x^{(1)} = (1/n,1/n,...,1/n)$
NO 7: Broyden Tridiagonal [13]: $x^{(1)} = (-1,-1,...,-1)$
NO 8: Nondia [14]: $x^{(1)} = (-1,-1,...,-1)$
NO 9: Tridia [14]: $x^{(1)} = (1,1,...,1)$
NO 10: Variably Dimensioned [10]: $x^{(1)} = (1-1/n, 1-2/n, ...1-i/n, ...,0)$
The numerical results are:

TF NO	n	f*	f_t	CPU(s)
1	2	0	9.27366E-011	0.01
2	2	0	1.57755E-006	0.04
3	12	0	3.92884E-010	0.06
		0	1.75E-007**	0.34**
4	10	0	2.3205E-010	0.04
		0	3.58E-006**	1.39**
	20	0	7.18471E-009	0.06
	40	0	5.122E-010	0.13
	50	0	1.03969E-007	0.10
5	10	0	1.05136E-011	0.02
	20	0	4.5335E-009	0.071
	30	0	4.4262E-010	0.06
	40	0	1.17215E-009	0.13
	50	0	6.35923E-007	1.823
6	4	0	4.31694E-012	0.01
		0	0**	0.04**
	100	0	4.96482E-007	1.412
	200	0	2.72988E-007	3.936
	300	0	5.84374E-008	7.17
7	3	0	8.22183E-012	0.00
		0	4.58E-008**	0.02**
	10	0	4.7211E-010	0.01
	20	0	1.06987E-008	0.03
	30	0	8.18231E-009	0.05
8	8	0	1.32813E-005	2.945
	10	0	2.73146E-005	0.732
	20	0	1.50366E-009	0.05
9	10	0	2.30387E-010	0.00
	20	0	7.40262E-010	0.02
	30	0	3.20689E-009	0.08
10	10	0	1.42468E-008	0.03
	20	0	1.27946E-006	0.24
	40	0	1.92605E-005	1.452
	50	0	3.42749E-005	2.564

*Table*1

In the above table, n represents the dimension of the test function, f^* represents the theoretical global optimal value, the f_t is the global optimal value we obtain and the CPU is the CPU time. The datum marked with ** are the results calculated by [7].
From the table we can draw the conclusion easily that the proposed approach is not only feasible but also more efficient than current methods with 2 main superiorities:
(1) more efficient with higher accuracy
The proposed method can solve all the test functions with the high accuracy no less than 10^{-5}. Besides, most of the problems can be solved with less CPU time. For example, we

cost only 0.04 seconds to solve the NO.4 test function with 10 dimensions while it cost 1.39 seconds to do the same work by the method in [7]. The similar case appeared in the calculation of NO.6 test function with 10 dimensions, etc.
(2) more efficient on solving higher dimensional problems
It is obvious compared with some other methods. Generally, the existing method can solve the test functions with 20 dimensions at most. However, the proposed approach can solve higher dimensional problems efficiently. For instance, it can solve the NO.6 test function with 300 dimensions.
All in all, the proposed approach is both feasible and efficient and its superiority becomes more obviously as the dimension of the problem increases.

4 Conclusions and future study

In this paper, a new modified pattern search approach combined with filter technique for global optimization problem is proposed. We modify the filter and the relevant modified approach becomes much more efficient then. Numerical experiments show the feasibility and effectiveness of the proposed method. Especially, for the problems with higher dimensions. Of course the proposed algorithm isn't perfect. For instance, the finite process in Step 2 is optional. We need to find a more effective way of this process. Besides, how to choose the better positive bases is another problem for future study.

References

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

[2] Box G.E.P, Evolationary operation: A method for increasing industrial productivity, Appl.Stat.,6(1957)81-101.

[3] R.Hooke, T.A.Jeeves, Direct search solution of numerical and statistical problems, J.ACM. 8(1961)212-229.

[4] Audet Charles, Dennis Jr.J.E., A pattern search filter method for nonlinear programming without derivatives, SIAM J. Optim.14(2004)980-1010.

[5] Coope I. D., Price C. J., On the convergence of grid-based methods for unconstrained minimization, SIAM J. Optim.11(2001) 859-869.

[6] Coope I. D., Price C. J., Frame based methods for unconstrained optimization, J. Optimiz. Theory Appl., 107(2000)261-274.

[7] Ting Wu, Lingping Sun, A filter-based pattern search method for unconstrained optimization, Numerical Mathematics.15(2006)209-216.

[8] C. J. Price, I.D. Coope, Frames and grids in unconstrained and linearly constrained optimization, A Nonsmooth approach, SIAM J. Optim. 14(2003)415-438.

[9] R. Fletcher, S. Leyffer, Nonlinear programming without a penalty function, Math.Program. 91(2002)239-269.

[10] J.J.More, B.S.Garbow, K.E.Hillstrom, Testing Unconstrained optimization software, ACM Transaction on Mathematical Software, 7(1981)17-41.

[11] M.A.Wolfe, C.Viazminsky, Supermemory descent methods for unconstrained minimization,J. Optimiz. Theory Appl.,18(1976)455-469.

[12] E. Spedicato, Computational experience with quasi-Newton algorithms for minimization problems of moderately large size, Report CISE-N-175, CISE Documentation Series, Segrato,1975.

[13] C.G.Broyden, A class of methods for solving nonlinear simultaneous equations, Mathematics of Computation, 19(1965)577-593.

[14] D.F.Shanno, Conjugate gradient methods with inexact searches, Mathematics of Computation, 34(1978)499-514.