算法实现计算结果

@author：豆奶 L

2020.07.18

摘要：本文是对算法实现部分的计算结果进行简单介绍，并汇总数值结果。文章首先按无约束问题和约束问题进行了分别介绍，最后进行本次算法实现的总结与收获。

1.无约束问题

在无约束问题中，我测试的问题是编号为 22、23 和 25 的问题（即 Extended Powell singular function, Penalty function and Trigonometric function，具体式子可见附录 1），实现的算法是最速下降法和 BFGS 算法。

1.1 参数选取说明

在无约束问题的两个算法实现中，本文采用的都是 Wolfe Powell 型线性搜索确定步长α。在这部分的参数选取中，也主要是对该搜索算法进行调参，其所需参数为$\sigma_1,\sigma_2,\beta,\rho,\rho_1$这五个参数，其中要满足$0<\sigma_1<1/2,\ \sigma_1<\sigma_2<1,\ \beta>0,\ \rho,\rho_1\in(0,1)$。对于以下参数的选取，采用控制变量法来寻找最优参数。

最速下降法：该算法的参数选取仅在确定步长α时需要，在这里，我将 Wolfe Powell 型线性搜索的五个参数设置为$\sigma_1=0.1,\ \sigma_2=0.9,\ \beta=15,\ \rho=0.2,\ \rho_1=0.5$。采用程序迭代来选取最优参数，如对$\sigma_1$，从 0.05 开始，以步长为 0.05 增长，一直到 0.45，一共 9 个数，分别代入算法中，寻找能使得运算结果最快、迭代次数最少、梯度最接近 0 的最优参数。得到结果如图 1 所示，综合来看，$\sigma_1=0.1$时，综合效果最好。类似的，可以得到其他参数的最优值。

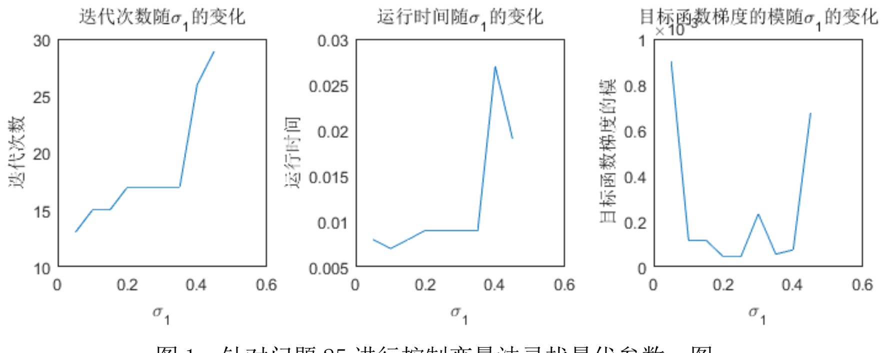

图 1：针对问题 25 进行控制变量法寻找最优参数σ_1图

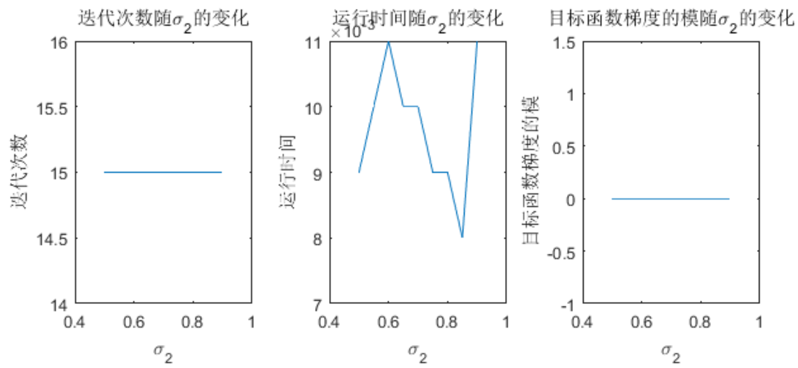

图 2：针对问题 25 进行控制变量法寻找最优参数σ_2图

BFGS 算法：对于 BFGS 算法，所需调整的参数也只有在确定步长α时需要，因此所需参数为$\sigma_1, \sigma_2, \beta, \rho, \rho_1$这五个参数。类似于上面，可得到这五个参数分别设置为$\sigma_1 = 0.01$，$\sigma_2 = 0.6$，$\beta = 15$，$\rho = 0.8$，$\rho_1 = 0.6$。

1.2 数值结果

表 1：最速下降法与 BFGS 算法的运行效果表(问题维度均为 100)

问题编号	使用的算法	运行次数	运行时间(s)	目标函数梯度的模
22	最速下降法	500	0.401	4.092791e-03
22	BFGS 算法	72	0.265	7.535416e-04
23	最速下降法	13	0.011	7.473050e-04
23	BFGS 算法	8	0.029	4.321353e-04
25	最速下降法	16	0.023	9.543579e-05
25	BFGS 算法	25	0.071	2.970324e-04

1.3 简单分析

从数值结果可以看出，BFGS 的迭代次数是比较少的，但对于某些问题，最速下降法也有其优势。此外，最速下降法比较简单，因此当迭代次数与 BFGS 算法差不多时，运行速度稍快于 BFGS 算法。

从解决问题的维度来看，随着问题维度不断增加，算法的运行时间也在增加，总体来看，BFGS 算法所需迭代次数较少，且所得目标函数的模也更小，但最速下降法的运行时间却比 BFGS 算法少，几乎可以忽略不记，这可能是因为 BFGS 算法要通过求解线性方程组来得到下降方向 d，而随着问题规模的增大，线性方程组的规模也在不断增大，从而使得算法运行时间较长。

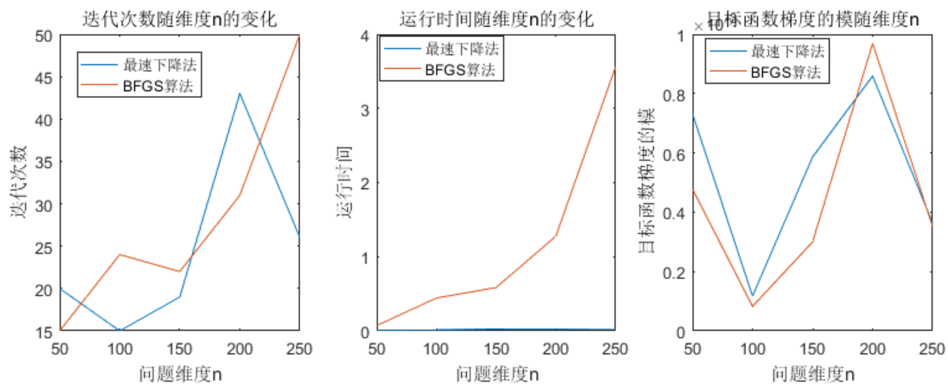

图 3：算法效果随问题维度的变化图

2. 约束问题

在约束问题中，我测试的问题是 De Jong's functions(A.2)和 Rastrigin's function(A.14)(具体式子见附录 2)，实现的算法是乘子法和 Frank Wolfe 算法。

2.1 参数选取说明

乘子法：乘子法所需要进行调整的参数有罚参数序列$\{\mu_k\}$和初始乘子向量λ。在这里我设置$\mu_k = 0.5$，为一常量。而$\lambda_j = 0(j \in E)$，$\lambda_i = 1(i \in I)$。

Frank Wolfe 算法：没有需要调整的参数。

2.2 选取的初始点

对问题 De Jong's functions(A. 2)，选取的初始点为(1, 1, ⋯, 1)

对问题 Rastrigin's function(A. 14)，选取的初始点为(1, 1, ⋯, 1)

2.3 求解无约束子问题的方法

乘子法中需要求解Lagrange函数的最小值，这里使用的是MATLAB的fmincon函数。该函数可用于求解带约束的非线性多变量函数的最小值，即可以用来求解非线性规划问题。

Frank Wolfe 算法中需要求解的无约束子问题有两个，一个是算法中的步 1 需要求解线性规划问题，这里使用的是 MATLAB 的 linprog 函数来求解线性规划问题；另一个是在算法中的步 3 求解线性搜索步长 t 时需要，不过这个是有约束子问题，所以这里使用的还是 MATLAB 的 fmincon 函数。

2.4 数值结果

表 2：约束问题在两个算法中的运行效果（问题维度：20）

问题名称	使用的算法	运行次数	运行时间(s)	KKT 系统残量
De Jong	乘子法	5	1.393	2.382287e-06
De Jong	Frank Wolfe	1	1.211	2.047883e-06
Rastrigin	乘子法	2	2.176	2.483107e-07
Rastrigin	Frank Wolfe	1	1.074	3.703714e-10

2.5 简单分析

从上面的数值结果可以看出来，迭代次数都很少。但运行时间却比无约束问题要长，这主要是因为约束问题需要求解无约束子问题，这里就需要花费较多的时间了。两种算法对比，可以看出 Frank Wolfe 算法拥有更快的迭代次数，以及更高的精度。看起来，Frank Wolfe 算法会比乘子法更优秀。

然而，当我改变问题的维度，来对比两种算法时，却比较出乎我意料，从图 4 可以看出来，乘子法显然优于 Frank Wolfe 算法，而且乘子法更为稳定，随着问题维度的增加，乘子法所需要的运行时间和迭代次数几乎没有变化，而 Frank Wolfe 算法则随着问题维度的变化而不断变化。

综上，乘子法有着更为稳定的数值效果以及更快的收敛速度。

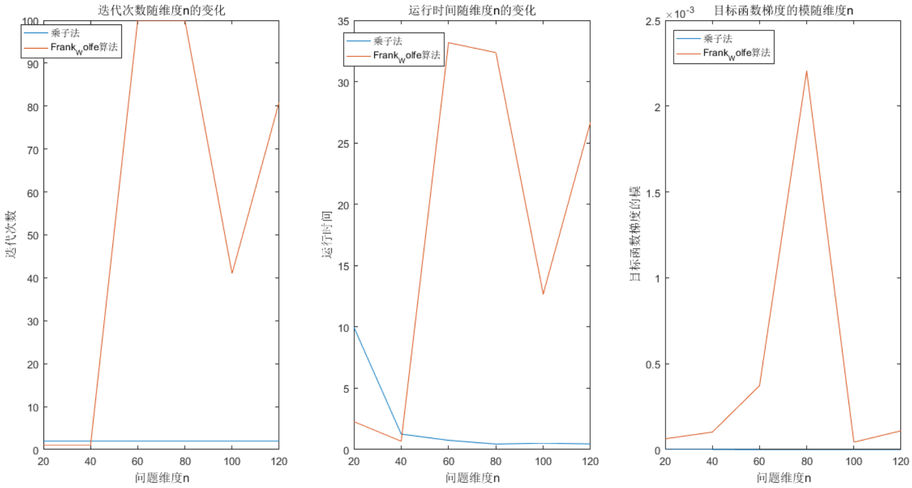

图 4：算法效果随问题维度的变化图

3. 总结与收获

通过这次算法实现，让我懂得了“纸上得来终觉浅，绝知此事要躬行”。在书上学到一个算法的时候，只是知道了这个算法的流程。而对于算法中每一步要用什么去求解、怎么去求解都没有很大的概念，比如，在实现 Wolfe Powell 算法时，其中步 1 便是找α_k是集合$\{\beta\rho^i|i = 0, \pm1,\pm2, \cdots\}$中使得(2.7)中第一个不等式成立的最大者这一步骤该如何实现。直到实现的时候，才想明白，可以先从 i=0 开始，然后去寻找到一个 i 最大，且 i-1 比 i 小。这就好像确定一个数是在哪两个数之间的道理是一样的。

此外，也教会了我实现代码的时候要认真仔细，比如当我实现无约束问题算法的时候，对每一个问题都需要求导和计算函数值，但由于我的粗心大意，经常求导错误，导致我一直在寻找算法实现的错误，直到最后才发现原来是我求导求错了，这可能也和我很少求解那么多维度的问题的导数有关。

当然，在算法的实现过程中，也让我对算法有了更多的了解，如对算法的运行速度、迭代次数和精度等。同时，也不得不说，想出这些算法的人真是聪明，讲一些无法直接求解的问题转化为可以直接求解的问题，如将约束问题转化为无约束问题来求解等。

附录 1 无约束优化算法测试问题

Almost all of the test functions that have appeared in the optimization literature are nonlinear least squares. Given a nonlinear least squares problem defined by $f_1, \ldots, f_m$, we can obtain an unconstrained minimization problem by setting

$$f(x) = \sum_{i=1}^{m} f_i^2(x). \tag{3.1}$$

问题 22 Extended Powell singular function

(a) n variable but a multiple of 4, $m = n$

(b) $f_{4i-3}(x) = x_{4i-3} + 10x_{4i-2}$

$f_{4i-2}(x) = 5^{1/2}(x_{4i-1} - x_{4i})$

$f_{4i-1}(x) = (x_{4i-2} - 2x_{4i-1})^2$

$f_{4i}(x) = 10^{1/2}(x_{4i-3} - x_{4i})^2$

(c) $x_0 = (\xi_j)$

where $\xi_{4j-3} = 3,\quad \xi_{4j-2} = -1,\quad \xi_{4j-1} = 0,\quad \xi_{4j} = 1$

(d) $f = 0$ at the origin

问题 23 Penalty function I

(a) n variable, $m = 2n$

(b) $f_1(x) = x_1 - 0.2$

$$f_i(x) = a^{1/2}\left(\exp\left[\frac{x_i}{10}\right] + \exp\left[\frac{x_{i-1}}{10}\right] - y_i\right), \qquad 2 \le i \le n$$

$$f_i(x) = a^{1/2}\left(\exp\left[\frac{x_{i-n+1}}{10}\right] - \exp\left[\frac{-1}{10}\right]\right), \qquad n < i < 2n$$

$$f_{2n}(x) = \left(\sum_{j=1}^{n} (n - j + 1)x_j^2\right) - 1$$

where $a = 10^{-5}$ and $y_i = \exp\left[\frac{i}{10}\right] + \exp\left[\frac{i-1}{10}\right]$

(c) $x_0 = (\frac{1}{2}, \ldots, \frac{1}{2})$

(d) $f = 9.37629 \cdots 10^{-6}$ if $n = 4$

$f = 2.93660 \cdots 10^{-4}$ if $n = 10$

问题 25 Variably dimensioned function

(a) n variable, $m = n + 2$
(b) $f_i(x) = x_i - 1, \quad i = 1, \ldots, n$

$$f_{n+1}(x) = \sum_{j=1}^{n} j(x_j - 1)$$

$$f_{n+2}(x) = \left(\sum_{j=1}^{n} j(x_j - 1)\right)^2$$

(c) $x_0 = (\xi_j)$ where $\xi_j = 1 - (j/n)$
(d) $f = 0$ at $(1, \ldots, 1)$

问题 26 Trigonometric function

(a) n variable, $m = n$

(b) $f_i(x) = n - \sum_{j=1}^{n} \cos x_j + i(1 - \cos x_i) - \sin x_i$

(c) $x_0 = (1/n, \ldots, 1/n)$
(d) $f = 0$

附录 2 约束优化测试问题

De Jong's functions

De Jong's functions: The simplest of De Jong's functions is the so-called sphere function

$$f(\boldsymbol{x}) = \sum_{i=1}^{n} x_i^2, \qquad -5.12 \le x_i \le 5.12, \tag{A.2}$$

whose global minimum is obviously $f_* = 0$ at $(0, 0, ..., 0)$. This function is unimodal and convex. A related function is the so-called weighted sphere function or hyper-ellipsoid function

$$f(\boldsymbol{x}) = \sum_{i=1}^{n} i x_i^2, \qquad -5.12 \le x_i \le 5.12, \tag{A.3}$$

which is also convex and unimodal with a global minimum $f_* = 0$ at $\boldsymbol{x}_* = (0, 0, ..., 0)$. Another related test function is the sum of different power function

$$f(\boldsymbol{x}) = \sum_{i=1}^{n} |x_i|^{i+1}, \qquad -1 \le x_i \le 1, \tag{A.4}$$

which has a global minimum $f_* = 0$ at $(0, 0, ..., 0)$.

Rastrigin's function

$$f(\boldsymbol{x}) = 10n + \sum_{i=1}^{n} \Big[x_i^2 - 10\cos(2\pi x_i)\Big], \qquad -5.12 \le x_i \le 5.12, \qquad \text{(A.14)}$$

whose global minimum is $f_* = 0$ at $(0, 0, ..., 0)$. This function is highly multi-modal.