Advances in Applied Mathematics 应用数学进展, 2024, 13(10), 4531-4546
Published Online October 2024 in Hans. https://www.hanspub.org/journal/aam
https://doi.org/10.12677/aam.2024.1310434

Hans汉斯

一种充分下降的修正PRP三项共轭梯度算法

刘紫烨[1*]，王松华[2]，赵世安[2#]

[1]暨南大学信息科学技术学院，广东 广州
[2]百色学院数理科学与统计学院，广西 百色

收稿日期：2024年9月14日；录用日期：2024年10月7日；发布日期：2024年10月15日

摘　要

共轭梯度算法是求解大规模无约束优化问题最有效的方法之一。文章提出一种新型的修正PRP三项共轭梯度算法，该算法具有不依赖任何线搜索充分下降的特点，搜索方向具有信赖域特征。在较为温和条件下，算法全局收敛。数值实验表明，新算法是有效的，比传统PRP三项共轭梯度算法更具竞争力。

关键词

大规模无约束优化，三项共轭梯度，充分下降，全局收敛

A Modified PRP Three-Term Conjugate Gradient Algorithm with Sufficient Descent Property

Ziye Liu[1*], Songhua Wang[2], Shi'an Zhao[2#]

[1]College of Information Science and Technology, Jinan University, Guangzhou Guangdong
[2]School of Mathematical Sciences and Statistics, Baise University, Baise Guangxi

Received: Sep. 14th, 2024; accepted: Oct. 7th, 2024; published: Oct. 15th, 2024

Abstract

The conjugate gradient algorithm is one of the most effective methods for solving large-scale unconstrained optimization problems. This paper proposes a novel modified Polak-Ribière-Polyak (PRP) three-term conjugate gradient algorithm, which possesses the characteristic of ensuring sufficient descent without relying on any line search conditions. The search direction of this algorithm

*第一作者。
#通讯作者。

文章引用: 刘紫烨, 王松华, 赵世安. 一种充分下降的修正 PRP 三项共轭梯度算法[J]. 应用数学进展, 2024, 13(10): 4531-4546. DOI: 10.12677/aam.2024.1310434

exhibits trust region properties. Under relatively mild conditions, the algorithm achieves global convergence. Numerical experiments demonstrate that the new algorithm is effective and more competitive compared to the classical PRP three-term conjugate gradient algorithm.

Keywords

Large-Scale Unconstrained Optimization, Three-Term Conjugate Gradient, Sufficient Descent, Global Convergence

1. 引言

考虑以下无约束优化问题：

$$\min\left\{f(x)\,|\,x\in R^n\right\},$$

其中，数值函数 $f:R^n\to R$ 为目标函数，连续可微。共轭梯度算法简单，存储需求少，效率高，是求解大规模无约束优化问题的有效工具之一[1]。其迭代公式为：

$$x_{k+1}=x_k+\alpha_k d_k,$$

搜索方向为：

$$d_k=\begin{cases}-g_k, & k=0\\ -g_k+\beta_k d_{k-1}, & k\geq 1\end{cases},$$

选取不同的 β_k 会得到不同的共轭梯度算法，不同的算法对解决优化问题会有不一样的数值效果。其中经典的 PRP 公式为：

$$\beta_k^{PRP}=\frac{g_k^{\mathrm{T}}(g_k-g_{k-1})}{\|g_{k-1}\|^2}。$$

经典 PRP 算法数值结果好，但是收敛性质一般。为此，一些修正的算法如混合共轭梯度法[2]-[4]，谱共轭梯度法[5]-[7]和三项共轭梯度法[8]-[11]等得到了很好的研究。其中三项共轭梯度法是重要的研究方向之一。

三项共轭梯度法由 Beale[8]在 1972 年首次提出，它的搜索方向为：

$$d_k=-g_k+\beta_k d_{k-1}+\gamma_{k-1}d_t,\ 1\leq t<k,$$

其中，$\beta_k=\beta_k^{HS}$ (或 $\beta_k^{FR},\beta_k^{DY}$)，

$$\gamma_{k-1}=\begin{cases}0, & k=t+1\\ \dfrac{g_k^{\mathrm{T}}y_t}{d_{k-1}^{\mathrm{T}}y_t}, & k>t+1\end{cases},$$

d_k 是一个重启方向。当 $k=t+1$ 时，该算法从 $-g_k+\beta_k d_{k-1}$ 方向重新开始。

Zhang [11]提出一个具有充分下降性的三项共轭梯度公式，它的搜索方向为：

$$d_k=\begin{cases}-g_k, & k=1\\ -g_k+\dfrac{g_k^{\mathrm{T}}y_{k-1}d_{k-1}-d_{k-1}^{\mathrm{T}}g_k y_{k-1}}{\|g_{k-1}\|\|g_{k-1}\|}, & k>1\end{cases},$$

其中，$y_{k-1}=g_k-g_{k-1}$，满足$d_k^{\mathrm{T}}g_k=-\|g_k\|^2$。通常，该算法被称呼为传统 PRP 三项共轭梯度算法，本文简称其为传统 PRP 三项算法。该算法一经提出，就引起了众多学者的关注，一些更为有效的三项共轭梯度算法被提出来。

Yuan [12]等提出了一个三项共轭梯度公式，搜索方向为：

$$d_k=\begin{cases}-g_k, & k=1\\ -g_k+\dfrac{g_k^{\mathrm{T}}y_{k-1}d_{k-1}-d_{k-1}^{\mathrm{T}}g_k y_{k-1}}{\max\left\{\mu\|y_{k-1}\|\|d_{k-1}\|,\|g_{k-1}\|^2\right\}}, & k>1\end{cases},$$

其中，$\mu>0$。该公式有充分下降性：$d_k^{\mathrm{T}}g_k=-\|g_k\|^2$，信赖域性质：$d_k\le\left(1+\dfrac{2}{\mu}\right)\|g_k\|$。

胡等[1]基于 Zhang [11]和 Yuan [12]算法，设计了一种三项共轭梯度公式，搜索方向为：

$$d_k=\begin{cases}-\mu_1 g_k, & k=1\\ -\mu_1 g_k+\mu_2\dfrac{g_k^{\mathrm{T}}y_{k-1}d_{k-1}-d_{k-1}^{\mathrm{T}}g_k y_{k-1}}{\max\left\{\left\|y_{k-1}^{\mathrm{T}}\right\|\|d_{k-1}\|,g_{k-1}^{\mathrm{T}}g_{k-1},d_{k-1}^{\mathrm{T}}d_{k-1}\right\}}, & k>1\end{cases},$$

其中，$\mu_1>1$，$\mu_2>1$。同样的，该三项共轭梯度法满足下降性和信赖域性。

王[13]等为克服 PRP 方法对非凸函数不收敛的缺点，提出一个修正的 PRP 公式，搜索方向为：

$$d_k=\begin{cases}-g_k, & k=1\\ -g_k+\dfrac{g_k^{\mathrm{T}}y_{k-1}d_{k-1}-d_{k-1}^{\mathrm{T}}g_k y_{k-1}}{\max\left\{2c\|y_{k-1}\|\|d_{k-1}\|,\|g_{k-1}\|^2\right\}+\left|d_{k-1}^{\mathrm{T}}y_{k-1}\right|}, & k>1\end{cases},$$

其中$c>0$。

基于以上讨论，本文主要结合胡等[1]和王[13]的特点，提出一个修正 PRP 三项共轭梯度法。搜索方向定义为：

$$d_k=\begin{cases}-g_k, & k=1\\ -g_k+\dfrac{g_k^{\mathrm{T}}y_{k-1}d_{k-1}-d_{k-1}^{\mathrm{T}}g_k y_{k-1}}{\mu\max\left\{\|y_{k-1}\|\|d_{k-1}\|,\|g_{k-1}\|^2\right\}+\left|d_{k-1}^{T}y_{k-1}\right|}, & k>1\end{cases},\tag{1}$$

其中$\mu>0$，$y_{k-1}=g_k-g_{k-1}$。

下面将给出新算法的步骤，然后讨论新算法的充分下降性，信赖域特征和全局收敛性。

2. MPRP 算法

为便于表述，本文所提修正 PRP 三项共轭梯度法记为 MPRP 算法。下面给出基于弱 Wolfe-Powell 线搜索方法的 MPRP 算法的一般步骤。

步骤 1：取初始点$x_1\in R^n$，其中$\mu>0$，$\varepsilon>0$，$0<\delta<\dfrac{1}{2}$，$\delta<\sigma<1$，令$\mathrm{d}_1=-\mathrm{g}_1$，$k=1$。

步骤 2：若 $\|g_k\| \le \varepsilon$，则算法停止；否则进入下一步。

步骤 3：通过弱 Wolfe-Powell 线搜索方法确定 α_k，其中 $0 < \delta < \frac{1}{2}$，$\delta < \sigma < 1$，

$$f\left(x_k + \alpha_k d_k\right) \le f\left(x_k\right) + \delta \alpha_k g_k^{\mathrm{T}} d_k, \tag{2}$$

$$g\left(x_k + \alpha_k d_k\right)^{\mathrm{T}} d_k \ge \sigma g_k^{\mathrm{T}} d_k。 \tag{3}$$

步骤 4：即令 $x_{k+1} = x_k + \alpha_k d_k$。

步骤 5：若 $\|g_{k+1}\| \le \varepsilon$，则算法停止；否则用式(2)计算 d_{k+1}。

步骤 6：令 $k = k+1$，并转到步骤 2。

3. 全局收敛性

在共轭梯度法全局收敛性分析中，一般需要以下假设[14] [15]：

假设 1：定义的水平集 $L_0 = \left\{x \mid f\left(x\right) \le f\left(x_1\right)\right\}$ 有界。

假设 2：目标函数 $f(x)$ 有下界，并且二次可微，它的梯度函数 $g(x)$ 是 Lipschitz 连续，即：

$$\left\|g\left(x\right) - g\left(y\right)\right\| \le L\left\|x - y\right\|,\ x, y \in R^n, \tag{4}$$

引理 1 假如搜索方向 d_k 由式(2)给出，那么 MPRP 算法具有以下两个性质：

$$g_k^{\mathrm{T}} d_k = -\left\|g_k\right\|^2, \tag{5}$$

$$\|d_k\| \le c\|g_k\|, \tag{6}$$

其中 $c > 0$，是一个常数。

证明：

1) 当 $k = 1$，容易得到式(5)和式(6)成立。

2) 当 $k > 1$ 时，式(1)左右两边同时左乘 g_k^{T}，有

$$g_k^{\mathrm{T}} d_k = g_k^{\mathrm{T}}\left(-g_k + \frac{g_k^{\mathrm{T}} y_{k-1} d_{k-1} - d_{k-1}^{\mathrm{T}} g_k y_{k-1}}{\mu \max\left\{\|y_{k-1}\|\|d_{k-1}\|, \|g_{k-1}\|^2\right\} + \left|d_{k-1}^{\mathrm{T}} y_{k-1}\right|}\right) = -\left\|g_k\right\|^2,$$

所以，式(5)成立。

对式(1)取模，根据 Cauchy-Schwarz 不等式性质，有

$$\begin{aligned}\|d_k\| &= \left\|-g_k + \frac{g_k^{\mathrm{T}} y_{k-1} d_{k-1} - d_{k-1}^{\mathrm{T}} g_k y_{k-1}}{\mu \max\left\{\|y_{k-1}\|\|d_{k-1}\|, \|g_{k-1}\|^2\right\} + \left|d_{k-1}^{\mathrm{T}} y_{k-1}\right|}\right\| \\ &\le \|g_k\| + \frac{2\|g_k\|\|y_{k-1}\|\|d_{k-1}\|}{\mu\|y_{k-1}\|\|d_{k-1}\|}, \\ &= \frac{\mu+2}{\mu}\|g_k\|\end{aligned}$$

令 $c \in \left[\frac{\mu+2}{\mu}, +\infty\right)$，那么式(6)成立。

式(5)证明了 MPRP 算法有充分下降性质，式(6)证明了 MPRP 算法有信赖域特征。这两个性质对共轭

梯度算法的全局收敛起着非常重要的作用。下面利用假设和引理 1，证明 MPRP 算法全局收敛。

定理 1 如果假设 1 和假设 12 成立，且迭代序列 $\{x_k,\alpha,d_k,g_k\}$ 由 MPRP 算法产生。那么

$$\lim_{k\to\infty}\|g_k\|=0\text{。} \tag{7}$$

证明：

由式(2)和式(5)，得到

$$f\left(x_k+\alpha_k d_k\right)\le f\left(x_k\right)+\delta\alpha_k g_k^{\mathrm{T}}d_k\le f\left(x_k\right)-\delta\alpha_k\|g_k\|^2,$$

将不等式从 $k=1$ 到 ∞ 求和，并根据假设 2，得到

$$\sum_{k=1}^{\infty}\delta\alpha_k\|g_k\|^2\le f\left(x_1\right)-f_\infty<\infty\text{。} \tag{8}$$

不等式(8)说明

$$\lim_{k\to\infty}\alpha_k\|g_k\|=0 \tag{9}$$

成立。

根据式(3)，式(4)，式(5)和 Cauchy-Schwarz 不等式性质，以下不等式

$$\begin{aligned}-(\sigma-1)\|g_k\|^2&\le(\sigma-1)g_k^{\mathrm{T}}d_k=\sigma g_k^{\mathrm{T}}d_k-g_k^{\mathrm{T}}d_k\le g\left(x_k+\alpha_k d_k\right)_k^{\mathrm{T}}d_k-g_k^{\mathrm{T}}d_k\\&\le\left[g\left(x_k+\alpha_k d_k\right)-g\left(x_k\right)\right]^{\mathrm{T}}d_k\le\left\|g\left(x_k+\alpha_k d_k\right)-g\left(x_k\right)\right\|\|d_k\|\le L\alpha_k\|d_k\|^2\end{aligned}$$

成立。于是，由式(6)，可得

$$\alpha_k\ge\frac{(1-\sigma)\|g_k\|^2}{L\|d_k\|^2}\ge\frac{(1-\sigma)\|g_k\|^2}{Lc^2\|g_k\|^2}=\frac{1-\sigma}{Lc^2},$$

结合式(9)，得

$$\lim_{k\to\infty}\|g_k\|=0,$$

所以式(7)成立。证明完毕。

4. 数值实验

本节主要进行数值实验，并将 MPRP 算法和传统 PRP 三项算法进行比较，分析 MPRP 算法在求解大规模无约束优化问题的有效性。传统 PRP 三项算法步骤：将 MPRP 算法的求 d_k 的公式替换成传统 PRP 算法[11]三项共轭梯度公式：

$$d_k=\begin{cases}-g_k, & k=1\\ -g_k+\dfrac{g_k^{\mathrm{T}}y_{k-1}d_{k-1}-d_{k-1}^{\mathrm{T}}g_k y_{k-1}}{\|g_{k-1}\|\|g_{k-1}\|}, & k>1\end{cases},$$

其他步骤与 MPRP 算法相同。

所有实验均在处理器为 Intel(R) Core(TM) i5-8265U，内存为 8.00 GB，Windows10，matlabR2021a 版本上运行。**参数选取：** $\delta=0.001$，$\sigma=0.82$，$\mu=0.01$。**Himmelblau 终止条件[16]：** 若 $\left|f\left(x_k\right)\right|>e_1$，设 $stop1=\dfrac{\left|f\left(x_k\right)-f\left(x_{k+1}\right)\right|}{\left|f\left(x_k\right)\right|}$；若 $\left|f\left(x_k\right)\right|\le e_1$ 则设 $stop1=\left|f\left(x_k\right)-f\left(x_{k+1}\right)\right|$。当条件式 $\left\|g\left(x\right)\right\|<e_2$ 或 $stop1<e_1$

成立，算法停止。如果线搜索次数大于 6，则线搜索技术接受 α_k。如果总迭代次数大于 800，算法停止。其中 $e_1=10^{-5}$， $e_2=10^{-6}$。

测试函数维数：用 Dim 表示，$n=900$，1500，4500，9000。**测试函数：**采用 Andrei 等文献中函数库中的 70 个测试函数[17] [18]。其中 No.表示测试函数的序号。

主要通过四个方面，对算法在给定问题上的数值性能进行分析，具体为：迭代次数(NI)、运行时间(CPU time)、目标函数计算次数(NF)和梯度函数计算次数(NG)。数值实验结果见表 1。

Table 1. Test questions and numerical results
表 1. 测试问题及数值结果

No.	Dim	MPRP 算法				传统 PRP 三项算法			
		NI	CPU time	NF	NG	NI	CPU time	NF	NG
1	900	24	0.031250	34	29	15	0.000000	22	20
	1500	23	0.015625	32	27	15	0.015625	22	20
	4500	23	0.093750	32	27	14	0.046875	21	19
	9000	23	0.296875	32	27	15	0.218750	23	20
2	900	47	0.000000	55	54	46	0.015625	54	54
	1500	44	0.000000	52	52	44	0.000000	53	52
	4500	49	0.250000	58	58	48	0.187500	57	57
	9000	50	0.562500	60	60	49	0.375000	59	59
3	900	25	0.000000	32	28	64	0.000000	118	85
	1500	13	0.000000	21	17	47	0.062500	93	65
	4500	11	0.000000	16	14	63	0.000000	115	88
	9000	18	0.109375	33	24	79	0.250000	161	113
4	900	53	0.015625	74	66	43	0.031250	89	62
	1500	49	0.062500	83	64	59	0.125000	115	86
	4500	29	0.406250	61	44	48	0.437500	103	72
	9000	37	0.515625	75	53	48	0.781250	89	68
5	900	23	0.000000	35	29	15	0.000000	23	19
	1500	21	0.031250	33	27	16	0.000000	25	21
	4500	27	0.046875	43	35	18	0.062500	29	24
	9000	23	0.171875	34	28	22	0.234375	33	27
6	900	63	0.000000	74	74	62	0.000000	73	73
	1500	65	0.000000	78	77	62	0.000000	74	74
	4500	71	0.093750	86	85	68	0.062500	83	82
	9000	75	0.062500	91	90	74	0.125000	90	89
7	900	239	0.031250	296	259	800	0.015625	801	801
	1500	253	0.015625	304	271	800	0.031250	801	801
	4500	596	0.375000	772	658	800	0.421875	801	801
	9000	179	0.343750	207	189	800	1.281250	801	801

续表

8	900	52	0.000000	53	53	20	0.000000	22	22
	1500	49	0.000000	50	50	20	0.000000	22	22
	4500	20	0.093750	24	22	20	0.046875	22	22
	9000	20	0.125000	24	22	20	0.062500	22	22
9	900	12	0.000000	13	13	12	0.000000	13	13
	1500	12	0.000000	13	13	12	0.000000	13	13
	4500	12	0.062500	13	13	12	0.046875	13	13
	9000	12	0.093750	13	13	12	0.109375	13	13
10	900	2	0.000000	7	2	2	0.000000	7	2
	1500	2	0.000000	7	2	2	0.000000	7	2
	4500	2	0.000000	7	2	2	0.000000	7	2
	9000	2	0.046875	7	2	2	0.000000	7	2
11	900	10	0.000000	16	11	63	0.031250	87	75
	1500	11	0.000000	22	13	54	0.031250	78	68
	4500	14	0.062500	28	20	13	0.000000	25	18
	9000	19	0.062500	34	26	8	0.125000	15	10
12	900	11	0.031250	12	12	17	0.000000	18	18
	1500	11	0.000000	12	12	17	0.031250	18	18
	4500	11	0.046875	12	12	17	0.000000	18	18
	9000	11	0.062500	12	12	19	0.156250	20	20
13	900	14	0.000000	19	16	11	0.000000	17	13
	1500	12	0.000000	16	14	13	0.031250	18	16
	4500	2	0.000000	7	2	2	0.000000	7	2
	9000	3	0.000000	13	5	5	0.000000	25	7
14	900	7	0.031250	9	8	8	0.015625	10	9
	1500	6	0.031250	8	7	7	0.000000	9	8
	4500	6	0.093750	8	7	6	0.078125	8	7
	9000	5	0.093750	7	6	6	0.109375	8	7
15	900	31	0.046875	39	35	33	0.031250	42	38
	1500	36	0.093750	46	40	26	0.046875	31	29
	4500	32	0.453125	37	35	31	0.468750	42	37
	9000	37	0.687500	50	43	32	0.562500	39	36
16	900	10	0.000000	12	11	9	0.000000	13	11
	1500	10	0.000000	14	12	18	0.000000	19	19
	4500	24	0.046875	25	25	16	0.046875	19	18
	9000	24	0.062500	25	25	14	0.109375	17	16

续表

17	900	29	0.031250	38	32	33	0.015625	38	35
	1500	32	0.031250	38	34	28	0.031250	34	30
	4500	36	0.312500	45	39	24	0.203125	29	26
	9000	40	0.750000	46	42	33	0.687500	38	35
18	900	4	0.000000	7	6	3	0.000000	5	5
	1500	4	0.000000	7	6	3	0.000000	5	5
	4500	4	0.000000	7	6	3	0.000000	5	5
	9000	5	0.000000	9	8	3	0.000000	5	5
19	900	3	0.000000	5	4	3	0.000000	5	4
	1500	3	0.000000	5	4	3	0.000000	5	4
	4500	3	0.000000	5	4	3	0.000000	5	4
	9000	3	0.062500	5	4	3	0.031250	5	4
20	900	22	0.000000	37	23	22	0.000000	37	26
	1500	21	0.000000	28	21	52	0.000000	64	55
	4500	25	0.000000	34	27	37	0.000000	45	38
	9000	9	0.000000	44	5	31	0.109375	57	34
21	900	28	0.000000	43	36	23	0.000000	29	27
	1500	29	0.046875	43	36	23	0.046875	28	27
	4500	24	0.062500	31	29	21	0.000000	26	25
	9000	25	0.125000	33	31	26	0.203125	31	30
22	900	10	0.000000	46	9	9	0.000000	40	9
	1500	10	0.000000	46	9	9	0.000000	40	9
	4500	10	0.125000	46	9	9	0.000000	40	9
	9000	10	0.093750	46	9	9	0.125000	40	9
23	900	101	0.046875	165	128	75	0.031250	122	96
	1500	106	0.046875	176	136	114	0.140625	198	153
	4500	203	1.156250	360	273	117	0.671875	201	154
	9000	109	1.500000	174	134	102	1.406250	160	127
24	900	40	0.000000	136	74	11	0.000000	61	26
	1500	30	0.000000	128	64	11	0.000000	61	26
	4500	18	0.062500	98	45	11	0.000000	58	25
	9000	18	0.203125	98	45	11	0.109375	58	25
25	900	15	0.000000	15	15	15	0.000000	15	15
	1500	20	0.000000	20	20	20	0.000000	20	20
	4500	13	0.000000	14	14	15	0.000000	27	18
	9000	7	0.000000	8	8	7	0.000000	8	8

续表

26	900	55	0.000000	68	68	55	0.031250	68	68
	1500	50	0.000000	63	63	50	0.000000	63	63
	4500	51	0.140625	64	64	51	0.093750	64	64
	9000	54	0.359375	67	67	54	0.296875	67	67
27	900	24	0.000000	43	32	15	0.000000	24	19
	1500	26	0.000000	47	35	23	0.000000	39	30
	4500	46	0.000000	86	64	32	0.000000	56	43
	9000	72	0.250000	136	102	76	0.234375	134	102
28	900	77	0.000000	130	98	31	0.000000	43	39
	1500	33	0.000000	91	40	28	0.000000	39	35
	4500	42	0.125000	73	52	34	0.000000	45	42
	9000	41	0.125000	73	52	31	0.125000	42	39
29	900	6	0.000000	19	7	4	0.000000	11	6
	1500	6	0.000000	19	7	4	0.000000	11	6
	4500	6	0.062500	19	7	4	0.000000	11	6
	9000	6	0.000000	19	7	4	0.000000	11	6
30	900	325	0.062500	425	360	800	0.000000	801	801
	1500	216	0.000000	289	241	800	0.062500	801	801
	4500	487	0.281250	646	542	800	0.187500	801	801
	9000	679	0.875000	918	762	800	0.875000	801	801
31	900	22	0.000000	26	26	22	0.000000	26	26
	1500	24	0.000000	28	28	24	0.000000	28	28
	4500	25	0.000000	30	30	25	0.000000	30	30
	9000	27	0.109375	32	32	27	0.000000	32	32
32	900	21	0.000000	23	23	21	0.000000	23	23
	1500	31	0.000000	33	33	31	0.000000	33	33
	4500	24	0.062500	27	27	24	0.062500	27	27
	9000	37	0.171875	40	40	37	0.140625	40	40
33	900	4	0.000000	5	4	4	0.000000	5	4
	1500	4	0.000000	5	4	4	0.000000	5	4
	4500	3	0.000000	4	3	3	0.000000	4	3
	9000	2	0.000000	3	2	2	0.000000	3	2
34	900	35	0.000000	35	35	35	0.000000	35	35
	1500	7	0.046875	7	7	7	0.046875	7	7
	4500	4	0.000000	4	4	4	0.000000	4	4
	9000	6	0.000000	6	6	6	0.078125	6	6

续表

35	900	7	0.000000	7	7	7	0.000000	7	7
	1500	9	0.000000	9	9	9	0.000000	9	9
	4500	14	0.000000	14	14	14	0.000000	14	14
	9000	19	0.000000	19	19	19	0.000000	19	19
36	900	24	0.062500	38	30	19	0.015625	36	27
	1500	23	0.093750	35	29	16	0.031250	32	24
	4500	13	0.093750	26	19	8	0.062500	16	12
	9000	14	0.343750	26	20	8	0.140625	16	12
37	900	779	0.031250	913	827	692	0.015625	694	693
	1500	274	0.046875	348	299	800	0.031250	813	805
	4500	210	0.093750	278	232	800	0.484375	803	801
	9000	698	1.562500	830	747	800	1.593750	815	806
38	900	10	0.000000	13	11	8	0.000000	11	10
	1500	10	0.000000	14	13	8	0.000000	11	10
	4500	14	0.000000	18	16	9	0.000000	16	10
	9000	12	0.000000	15	14	9	0.000000	16	10
39	900	24	0.000000	24	24	24	0.000000	24	24
	1500	25	0.031250	25	25	25	0.000000	25	25
	4500	27	0.000000	27	27	27	0.078125	27	27
	9000	28	0.000000	28	28	28	0.078125	28	28
40	900	740	0.875000	1143	898	800	0.859375	887	837
	1500	307	0.593750	457	367	800	1.484375	1023	892
	4500	294	3.703125	428	348	800	9.593750	973	878
	9000	238	6.093750	365	289	800	19.812500	1003	886
41	900	40	0.000000	48	44	43	0.000000	56	49
	1500	44	0.000000	52	48	33	0.000000	43	38
	4500	42	0.000000	49	45	40	0.000000	54	46
	9000	42	0.156250	51	46	27	0.109375	36	31
42	900	4	0.000000	19	2	4	0.000000	19	2
	1500	4	0.000000	19	2	4	0.000000	19	2
	4500	4	0.000000	19	2	4	0.000000	19	2
	9000	4	0.093750	19	2	4	0.000000	19	2
43	900	34	0.062500	36	36	32	0.031250	34	34
	1500	35	0.078125	37	37	33	0.031250	35	35
	4500	37	0.500000	39	39	35	0.531250	37	37
	9000	39	1.000000	41	41	36	0.609375	38	38

续表

44	900	28	0.015625	31	31	32	0.015625	35	35
	1500	18	0.015625	21	21	18	0.015625	21	21
	4500	17	0.187500	20	20	17	0.125000	20	20
	9000	17	0.312500	20	20	17	0.406250	20	20
45	900	22	0.015625	25	25	21	0.031250	24	24
	1500	22	0.015625	25	25	22	0.062500	25	25
	4500	23	0.156250	26	26	23	0.375000	26	26
	9000	24	0.468750	27	27	24	0.562500	27	27
46	900	75	0.046875	92	83	71	0.062500	74	74
	1500	76	0.109375	97	85	71	0.078125	74	74
	4500	101	0.953125	135	114	176	1.296875	182	180
	9000	79	1.468750	97	87	191	4.140625	197	195
47	900	63	0.171875	80	70	64	0.171875	78	70
	1500	50	0.343750	65	56	47	0.312500	61	54
	4500	24	1.593750	38	30	34	2.156250	47	40
	9000	38	9.312500	64	49	36	8.328125	52	42
48	900	24	0.031250	26	25	33	0.015625	37	35
	1500	46	0.125000	54	49	29	0.046875	36	32
	4500	38	0.281250	50	43	35	0.453125	44	40
	9000	38	0.640625	46	41	34	0.625000	41	37
49	900	218	0.000000	288	243	800	0.000000	801	801
	1500	272	0.015625	363	303	800	0.031250	801	801
	4500	481	0.281250	632	535	800	0.328125	801	801
	9000	764	0.984375	874	803	800	0.968750	801	801
50	900	225	0.171875	297	251	800	0.484375	801	801
	1500	210	0.203125	261	228	800	0.734375	801	801
	4500	386	2.812500	513	431	800	5.406250	801	801
	9000	460	7.859375	618	515	800	12.734375	801	801
51	900	20	0.015625	22	22	18	0.031250	20	20
	1500	19	0.031250	21	21	18	0.015625	20	20
	4500	19	0.171875	21	21	18	0.234375	20	20
	9000	19	0.390625	21	21	18	0.296875	20	20
52	900	84	0.000000	101	101	84	0.000000	101	101
	1500	90	0.000000	108	108	90	0.000000	108	108
	4500	103	0.046875	123	123	103	0.093750	123	123
	9000	109	0.218750	131	131	109	0.218750	131	131

续表

53	900	399	0.000000	537	449	800	0.031250	803	802
	1500	619	0.031250	812	690	800	0.046875	803	802
	4500	800	0.609375	876	830	800	0.468750	803	802
	9000	360	0.687500	474	401	800	1.437500	803	802
54	900	27	0.000000	50	35	25	0.000000	44	34
	1500	29	0.000000	46	36	20	0.000000	34	25
	4500	17	0.078125	30	23	27	0.000000	49	37
	9000	33	0.000000	49	41	37	0.125000	59	49
55	900	20	0.046875	21	21	20	0.000000	21	21
	1500	20	0.031250	21	21	20	0.031250	21	21
	4500	21	0.156250	22	22	21	0.140625	22	22
	9000	22	0.281250	23	23	22	0.265625	23	23
56	900	249	0.015625	325	275	800	0.015625	802	801
	1500	344	0.000000	468	387	800	0.031250	802	801
	4500	172	0.156250	228	192	800	0.421875	802	801
	9000	499	1.046875	665	558	800	1.328125	802	801
57	900	56	0.031250	71	63	77	0.046875	82	81
	1500	49	0.062500	61	55	65	0.078125	70	69
	4500	48	0.375000	60	54	71	0.609375	76	75
	9000	60	1.046875	74	66	81	1.453125	86	85
58	900	84	0.046875	106	94	113	0.078125	120	118
	1500	100	0.125000	117	108	94	0.093750	101	99
	4500	108	1.046875	145	124	136	1.125000	143	141
	9000	109	2.437500	142	123	126	1.812500	133	131
59	900	45	0.031250	58	52	122	0.078125	126	126
	1500	45	0.062500	57	52	117	0.125000	121	121
	4500	54	0.500000	72	63	146	1.109375	150	150
	9000	58	1.296875	75	66	176	3.437500	180	180
60	900	71	0.046875	88	79	75	0.046875	78	78
	1500	54	0.062500	69	61	73	0.062500	76	76
	4500	101	0.890625	113	107	562	4.859375	568	566
	9000	119	2.000000	153	133	758	15.437500	764	762
61	900	54	0.078125	69	61	77	0.046875	82	81
	1500	53	0.062500	66	59	65	0.093750	70	69
	4500	55	0.421875	71	62	71	0.546875	76	75
	9000	65	1.406250	84	73	81	1.531250	86	85

续表

62	900	49	0.031250	57	54	104	0.062500	110	109
	1500	74	0.078125	93	83	109	0.109375	115	114
	4500	59	0.500000	72	66	106	0.953125	112	111
	9000	82	1.625000	108	93	117	2.218750	123	122
63	900	95	0.062500	115	105	112	0.093750	119	117
	1500	132	0.156250	169	146	121	0.125000	128	126
	4500	127	1.187500	164	143	228	2.093750	248	238
	9000	191	3.406250	253	214	800	14.453125	811	806
64	900	36	0.031250	39	39	35	0.031250	38	38
	1500	37	0.062500	40	40	37	0.078125	40	40
	4500	39	0.265625	42	42	39	0.312500	42	42
	9000	40	0.484375	43	43	40	0.984375	43	43
65	900	21	0.015625	25	24	18	0.046875	22	21
	1500	21	0.031250	25	24	17	0.031250	21	20
	4500	19	0.171875	21	21	18	0.156250	22	21
	9000	15	0.171875	17	17	15	0.375000	17	17
66	900	460	0.312500	632	523	800	0.468750	802	801
	1500	224	0.218750	296	251	800	0.703125	802	801
	4500	156	1.250000	208	175	800	5.687500	802	801
	9000	68	1.062500	96	79	800	12.953125	802	801
67	900	6	0.000000	31	2	6	0.000000	26	3
	1500	6	0.000000	31	2	7	0.000000	37	2
	4500	7	0.000000	37	2	7	0.078125	37	2
	9000	28	0.281250	79	27	5	0.109375	20	3
68	900	33	0.000000	34	34	33	0.000000	34	34
	1500	34	0.000000	35	35	34	0.000000	35	35
	4500	36	0.125000	37	37	36	0.000000	37	37
	9000	37	0.125000	38	38	37	0.125000	38	38
69	900	28	0.000000	30	30	31	0.000000	33	33
	1500	29	0.000000	31	31	32	0.000000	34	34
	4500	32	0.000000	34	34	34	0.000000	36	36
	9000	33	0.125000	35	35	35	0.125000	37	37
70	900	11	0.046875	27	17	21	0.046875	35	28
	1500	30	0.078125	60	41	29	0.109375	51	38
	4500	24	0.234375	49	33	23	0.328125	45	32
	9000	16	0.437500	34	22	48	1.343750	107	73

为直观比较两种算法的性能，本文引用 Dolan 和 Moré 文献[19]评价方法作出了 MPRP 算法和传统 PRP 三项算法的 CPU time、NI、NF 和 NG 的性能分析比较图，见图 1 至图 4。图 1 表明，MPRP 算法以最少的运行时间解决了大约 37%的测试问题，而传统 PRP 三项算法以最少的运行时间只解决了大约 27%的测试问题。图 2 表明，MPRP 算法以最少的迭代次数解决了大约 65%的测试问题，而传统 PRP 三项算法以最少的运行时间解决了大约 63%的测试问题，尽管最开始时两个算法的差距不大，但是当 $\tau=1.8$，MPRP 可以解决约 98%的问题，这个结果表明 MPRP 算法有更强的鲁棒稳定性。图 3 表明，虽然以最少的目标函数计算次数传统 PRP 三项算法解决的测试问题略比 MPRP 算法多，但是当 $\tau=1.8$，MPRP 可以解决 97%以上的问题，这个结果表明 MPRP 算法有更强的鲁棒稳定性。图 4 表明，以最少的梯度函数计

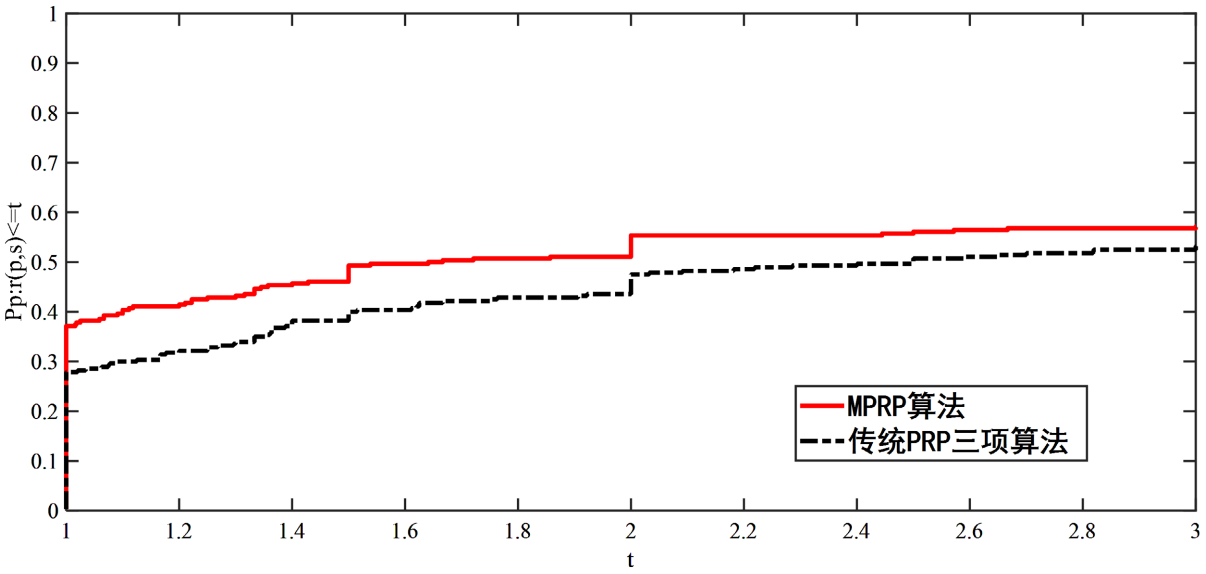

Figure 1. Comparison graph of runtime
图 1. 运行时间对比图

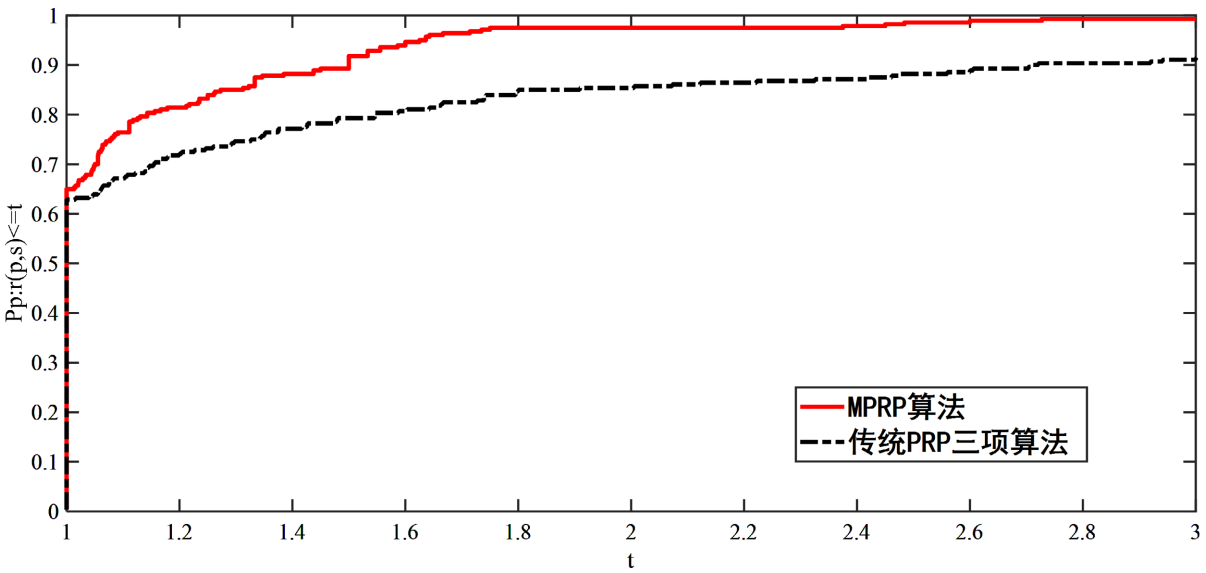

Figure 2. Comparison graph of iteration number
图 2. 迭代次数对比图

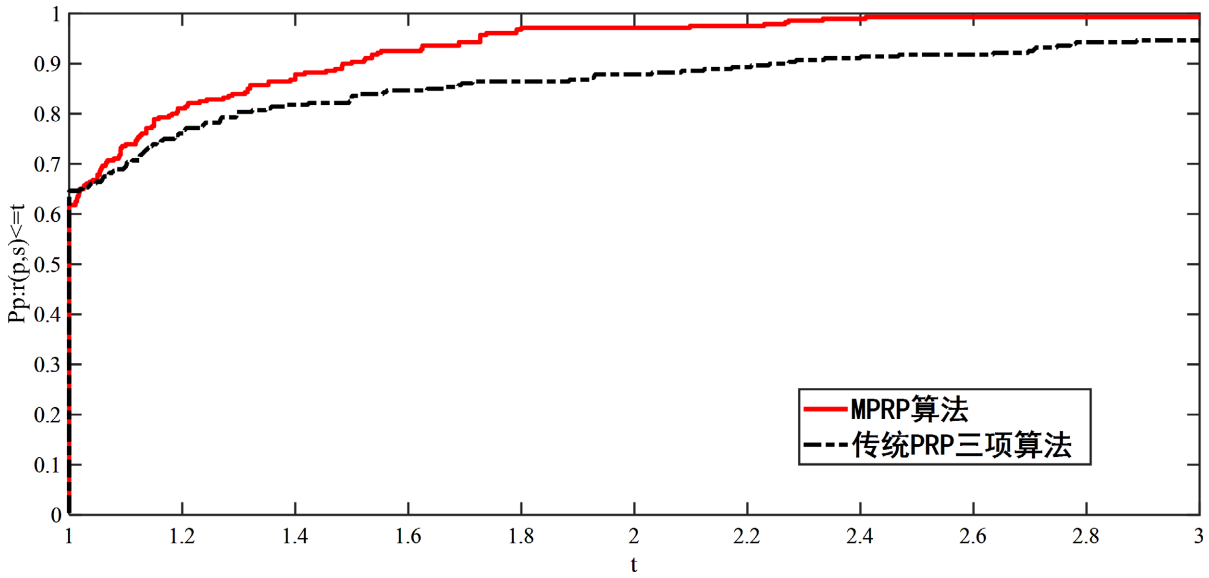

Figure 3. Comparison graph of objective function calculation counts
图 3. 目标函数计算次数对比图

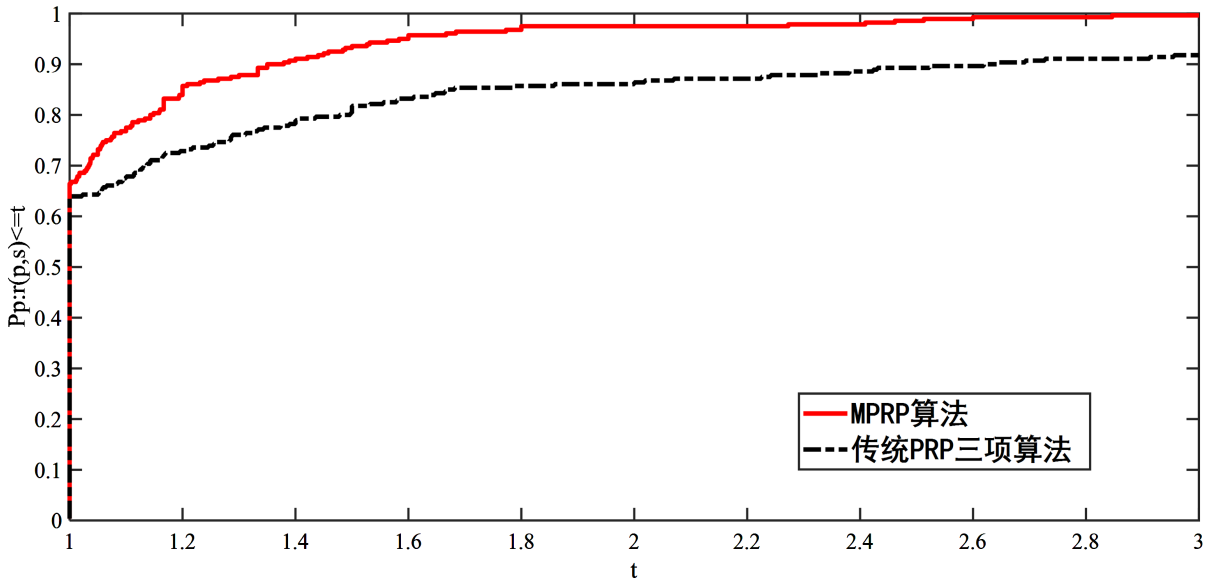

Figure 4. Comparison graph of gradient function computation count
图 4. 梯度函数计算次数对比图

算次数 MPRP 算法比传统 PRP 三项算法解决的测试问题多，且 MPRP 算法有更强的鲁棒稳定性。综上，根据图 1 至图 4 比较图综合分析，对选取的 70 个测试问题，MPRP 算法相对于传统 PRP 三项算法，具有一定的竞争力。其中在运行时间、迭代次数、目标函数计算次数、梯度函数计算次数等方面，其稳定性都优于传统 PRP 三项算法。而在运行时间、迭代次数、梯度函数计算次数等方面，其效率都优于传统 PRP 三项算法。所以 MPRP 算法共轭梯度法比传统 PRP 三项算法更具有竞争性。

5. 结论

本文提出一种新型的 PRP 三项共轭梯度算法，该算法具有充分下降性和信赖域特征，结合弱 Wolfe-

Powell 线搜索，算法全局收敛。数值实验表明，所提算法是有效的，比传统 PRP 三项共轭梯度算法更具竞争力。未来我们将进行更大规模问题的数值实验，并检验该算法在大规模非线性方程组问题和图像处理问题的效率，进一步增强算法的竞争力。

基金项目

广西自然科学基金资助项目(2024GXNSFAA010478, 2020GXNSFAA159069)；中央引导地方科技发展专项资金资助(桂科 ZY20198003)。

参考文献

[1] 胡午杰, 袁功林. 求解非线性方程组的一种三项共轭梯度法[J]. 广西大学学报(自然科学版), 2019, 44(5): 1485-1490.

[2] Andrei, N. (2008) Hybrid Conjugate Gradient Algorithm for Unconstrained Optimization. *Journal of Optimization Theory and Applications*, **141**, 249-264. https://doi.org/10.1007/s10957-008-9505-0

[3] Livieris, I.E., Tampakas, V. and Pintelas, P. (2018) A Descent Hybrid Conjugate Gradient Method Based on the Memoryless BFGS Update. *Numerical Algorithms*, **79**, 1169-1185. https://doi.org/10.1007/s11075-018-0479-1

[4] Djordjević, S.S. (2019) New Hybrid Conjugate Gradient Method as a Convex Combination of Ls and Fr Methods. *Acta Mathematica Scientia*, **39**, 214-228. https://doi.org/10.1007/s10473-019-0117-6

[5] Jian, J., Chen, Q., Jiang, X., Zeng, Y. and Yin, J. (2016) A New Spectral Conjugate Gradient Method for Large-Scale Unconstrained Optimization. *Optimization Methods and Software*, **32**, 503-515. https://doi.org/10.1080/10556788.2016.1225213

[6] Faramarzi, P. and Amini, K. (2019) A Modified Spectral Conjugate Gradient Method with Global Convergence. *Journal of Optimization Theory and Applications*, **182**, 667-690. https://doi.org/10.1007/s10957-019-01527-6

[7] Fatemi, M. (2016) A Scaled Conjugate Gradient Method for Nonlinear Unconstrained Optimization. *Optimization Methods and Software*, **32**, 1095-1112. https://doi.org/10.1080/10556788.2016.1233971

[8] Beale, E. (1972) A Derivation Conjugate Gradient. Academic Press, 39-43.

[9] Yuan, G., Wei, Z. and Li, G. (2014) A Modified Polak-Ribière-Polyak Conjugate Gradient Algorithm for Nonsmooth Convex Programs. *Journal of Computational and Applied Mathematics*, **255**, 86-96. https://doi.org/10.1016/j.cam.2013.04.032

[10] Andrei, N. (2013) A Simple Three-Term Conjugate Gradient Algorithm for Unconstrained Optimization. *Journal of Computational and Applied Mathematics*, **241**, 19-29. https://doi.org/10.1016/j.cam.2012.10.002

[11] Zhang, L., Zhou, W. and Li, D. (2006) A Descent Modified Polak-Ribière-Polyak Conjugate Gradient Method and Its Global Convergence. *IMA Journal of Numerical Analysis*, **26**, 629-640. https://doi.org/10.1093/imanum/drl016

[12] Yuan, G. and Zhang, M. (2015) A Three-Terms Polak-Ribière-Polyak Conjugate Gradient Algorithm for Large-Scale Nonlinear Equations. *Journal of Computational and Applied Mathematics*, **286**, 186-195. https://doi.org/10.1016/j.cam.2015.03.014

[13] 王松华, 黎勇. 求解大规模无约束优化问题的一种新的 PRP 三项共轭梯度法[J]. 广西民族大学学报(自然科学版), 2018, 24(4): 58-66

[14] Fletcher, R. (1964) Function Minimization by Conjugate Gradients. *The Computer Journal*, **7**, 149-154. https://doi.org/10.1093/comjnl/7.2.149

[15] Dai, Y.H. and Yuan, Y. (1999) A Nonlinear Conjugate Gradient Method with a Strong Global Convergence Property. *SIAM Journal on Optimization*, **10**, 177-182. https://doi.org/10.1137/s1052623497318992

[16] 袁亚湘, 孙文瑜. 最优化理论与方法[M]. 北京: 科学出版社, 1997: 183-218.

[17] Andrei, N. (2008) An Unconstrained Optimization Test Functions Collection. *Environmental Science and Technology*, **10**, 6552-6558.

[18] Bongartz, I., Conn, A.R., Gould, N. and Toint, P.L. (1995) CUTE: Constrained and Unconstrained Testing Environment. *ACM Transactions on Mathematical Software*, **21**, 123-160. https://doi.org/10.1145/200979.201043

[19] Dolan, E.D. and Moré, J.J. (2002) Benchmarking Optimization Software with Performance Profiles. *Mathematical Programming*, **91**, 201-213. https://doi.org/10.1007/s101070100263