



基于网格的 Nelder-Mead 不等式约束优化算法

Charles Audet¹ - Christophe Tribes¹

收稿日期：2017 年 11 月 9 日 / 在线发表日期：2018 年 6 月 12 日 2017 年 11 月 9 日 / 在线发表：

2018 年 6 月 12 日

© Springer Science+Business Media, LLC, part of Springer Nature 2018

摘要 尽管缺乏理论和实际收敛性支持，Nelder-Mead (NM) 算法仍被广泛用于解决无约束优化问题。它是一种无导数算法，试图用一个更好的点迭代替换单纯形中最差的点。本文提出了一种将 NM 算法扩展到不等式约束优化的方法。这是通过网状自适应直接搜索 (Mads) 算法的搜索步骤实现的，该算法受到 NM 算法的启发。所提出的算法不存在 NM 算法收敛性不足的问题，而是继承了 Mads 收敛性分析的全部内容。数值实验表明，使用该搜索步骤生成的解的质量有了显著提高。

关键词 内尔德-梅德 - MADS - 无衍生优化 - 黑箱优化 - 受限优化

1 引言

无导数的 Nelder-Mead 算法 (NM) 于 1965 年被引入 \mathbb{R}^n [38]，用于无约束优化问题。虽然该算法非常流行，但它无法解决某些优化问题 [17,42]。1998 年，McKinnon [32] 提出了

✉ Christophe Tribes
christophe.tribes@polymtl.ca
Charles Audet
Charles.Audet@gerad.ca

¹ GERAD 和蒙特利尔理工学院数学与工业数学系、
C.P. 6079, Succ.Centre-ville, Montreal, QC H3C 3A7, Canada

R^2 中的平滑严格凸函数，在该函数上，NM 算法产生的试验点序列无法接近唯一最小值。

为了保证在平滑情况下收敛到临界点，人们提出了使用目标函数值充分下降的不同变体 [26,37,45]。Price 等人[40] 提出了一种基于框架的 NM 方法[15]，要求充分下降。Gao 和 Han [20] 研究了 NM 算法在凸目标函数上的行为，并针对具有自适应功能的较大问题提出了一种有效的变体。
单工参数。

在非光滑情况下，提出了一个随机版本 [11]。Brea [9] 针对非线性混合问题提出了一种受 NM 启发的算法。Bu'rmes 等人[10] 对 NM 进行了调整，使试验点在底层网格（即变量空间的离散化）上生成。本文在此方向上更进一步，但针对的是无导数的不等式约束优化。具体来说，在*网格自适应直接搜索*（Mads）算法[2,3]的搜索步骤中插入了 NM 算法。由此产生的名为 Mads-NM 的算法与 Mads 算法一样，受益于针对非光滑约束优化的分层收敛分析。

本文结构如下。第 2 节介绍了用于无约束优化的原始 NM 算法。第 3 节对 Mads 算法进行了高层次的描述，但只关注了当前工作所需的要素。接下来，第 4 节提出了一种插入 NM 算法的方法。4 节提出了在 Mads 中插入 NM 搜索步骤的方法。第 5 节报告了数值实验。我们在一组 87 个测试问题上进行了密集测试，以调整新算法参数。然后，在不添加任何额外参数的情况下启动算法，并将其应用于三个受限的黑箱工程测试问题。我们的数值实验表明，在 Mads 算法中插入一个 NM 搜索步骤可显著提高其性能。

2 NM 无约束优化算法

无约束优化的 NM 算法有多种呈现方式

分钟

$$\min_{x \in R^n} f(x)$$

其中， $f: R^n \rightarrow R$ 是目标函数。我们的表述方式可能会显得过于复杂，但却

受约束的情况。以下定义用于对试验点进行排序。

定义 2.1 如果出现以下情况，则称一个点 $x \in \mathbb{R}^n$ 支配 $y \in \mathbb{R}^n$ ，记为 $x < y$ $f(x) < f(y)$ 。

在两个试验点的目标函数值相同（即 $f(x) = f(y)$ ）的情况下，我们不能对它们任意排序，而是要使用下面的确定性函数，返回其中最老的点。

定义 2.2 如果 x 在 y 之前由算法生成, 则称 $x \in R^n$ 的点比 $y \in R^n$ 的点更早。
函数 Older : $R^n \times R^n \rightarrow R^n$

$$\text{Older}(x, y) = \begin{cases} x & \text{如果 } x \text{ 在 } y \text{ 之前被算法生成, 则 } y \text{ 否则} \\ y & \end{cases}$$

返回两点中最古老的点。

结合优势的定義, 我们可以利用函数 Older 引入以下规则来比较两个试验点。

定义 2.3 函数 Best : $R^n \times R^n \rightarrow R^n$

$$\text{Best}(x, y) = \begin{cases} x & \text{如果 } x < y \\ y & \text{如果 } y < x \\ \text{Older}(x, y) & \text{如果 } f(x) = f(y) \end{cases}$$

返回 R^n 中两个点 x 和 y 的最佳值。一个点 $x \in R^n$ 称为优于 $y \in R^n$, 如果 $x = \text{Best}(x, y)$ 。

函数 Best 显然具有传递性和交换性, $\text{Best}(x, x)$ 返回 x , 因此定义明确。

现在可以用上述定义来介绍 NM 算法。NM 算法的每次迭代都从 R^n 中 $n+1$ 个仿射独立点的集合 $Y = \{y^0, y^1, \dots, y^n\}$ 开始, 该集合定义了一个单纯形。

单纯形点是有序的, 因此 y^{i-1} 是

优于 y^i , 每 $i = 1, 2, \dots, n$ 。每个 NM 迭代的目标是用一个更好的点替换最差的单纯形点 y^n 。为此, NM 使用

以下是一些要点和参数:

$$\begin{aligned} x^c &= \frac{1}{n+1} \sum_{i=0}^n y^i & n \text{ 个最佳点的中心点;} \\ x^r &= x^c + (x^c - y^n) & \text{最差点相对于中心点的反射;} \\ x^e &= x^c + \delta^e (x^c - y^n) & \text{其中 } \delta^e \in]1, \infty[\text{ 是扩展参数;} \\ x^{oc} &= x^c + \delta^{oc} (x^c - y^n) & \text{其中 } \delta^{oc} \in]0, 1[\text{ 是外部收缩参数;} \quad x^{ic} = x^c + \delta^{ic} \\ & & \text{其中 } \delta^{ic} \in]-1, 0[\text{ 是内收缩参数;} \quad \gamma \in]0, 1[\text{ 是} \\ & & \text{收缩参数。} \end{aligned}$$

在每次迭代中, NM 要么用 x^r , x^e , x^{oc} 或 x^{ic} 替换有序单纯形的最后一个点 y^n , 要么通过替换除最佳顶点 y^0 以外的所有顶点来缩小单纯形, 这样单纯形就变成了

$$Y \leftarrow \{y, y^{00} + \gamma(y^1 - y^0), y^0 + \gamma(y^2 - y^0), \dots, y^0 + \gamma(y^n - y^0)\} \quad (1)$$

$$\delta^e = 2, \delta^{ic} = -\frac{1}{2}, \delta^{oc} = \frac{1}{2} \text{ 和 } \gamma = \frac{1}{2}$$

确定新单线的比较是相对于以下区域进行的。

定义 2.4（无约束优化的区域） 让 $Y = \{y, y^{01}, ..., y\}^n$ 是 R^n 中的有序单纯形。试验点 $x \in R^n$

- 如果 $y^n < x$ ，则属于内收缩区；
- 否则，如果 $x < y^0$ ，它就属于扩展区；
- 否则，如果 x 至少支配 Y 的 2 个点，它就属于反射区；
- 否则，它属于外收缩区， x 主导 Y 的 0 或 1 点。

NM 算法简洁地写在算法 1 中。

算法 1：内尔德-梅德无约束算法（NM）

给定初始单纯形 $Y = \{y^0, y^1, ..., y^n\}$ 的顶点。

1. 更新单纯形：

重新排序 Y 。

2. 建立一个新的单丁：

如果 属于内收缩区

如果 x^{ic} 属于内收缩区，则利用公式 (1) 缩小 Y ，否则设置 $y^n \leftarrow x^{ic}$ ；

否则，如果 x^r 属于扩展区，则设 $y^n \leftarrow \text{Best} (x^r, x^e)$ ；

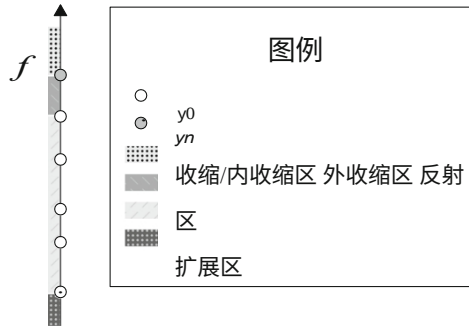
否则，如果 x^r 属于反射区，则设置 $y^n \leftarrow x^r$ ；

否则（ x^r 属于外部收缩区域），设 $y^n \leftarrow \text{Best} (x^r, x^{oc})$ 。

转到 1。

图 1 提供了对 NM 算法机制的图解。纵轴表示目标函数 f ，六个圆圈表示目标函数值 $f(y^0) < f(y^1) < --- < f(y^5) = f(y^n)$ 。只有当 x^r 属于外部收缩区域时，才能用 x^{oc} 替换顶点 y^n 如图所示，只有当 x^r 属于内收缩区时，才能用 x^{ic} 替换 y^n ，等等。

图 1 相对于 a
6 点 单工



3 马德斯约束优化算法

我们考虑用渐进障碍 (PB) 算法 [3] 来处理不平等约束条件，并用动态缩放算法 [8] 来处理变量的大小变化。优化问题的目标类别是

$$\begin{aligned} & \underset{x \in X \subset \mathbb{R}^n}{\text{分钟}} \quad f(x) \\ & \text{s.t. } c(x) \leq 0, \end{aligned}$$

其中 $f: X' \rightarrow \mathbb{R} \cup \{\infty\}$ 和 $c: X' \rightarrow (\mathbb{R} \cup \{\infty\})^m$ 是函数， $c = (c_1, c_2, \dots, c_m)^T$ ， X 是 \mathbb{R}^n 的某个子集ⁿ。注意，允许函数返回值 ∞ ，这是表示求值失败的一种方便技术。整个可行区域用集合 $\blacktriangle = \{x \in X: c(x) \leq 0\}$ 表示。Mads 算法的目标问题是对 f 或 c 的平滑性不做任何假设的问题。

5]。

PB 使用违反约束函数 [18,19]

$$h(x) := \begin{cases} \sum_{j \in J} \max\{c_j(x), 0\}^2 & \text{若 } x \in X \\ \infty & \text{否则、} \end{cases}$$

其中， $J = \{1, 2, \dots, m\}$ 是约束条件的索引集。约束违反函数值 $h(x)$ 是非负值，并且只有当且仅当点 x 属于以下情况时才等于零到 \blacktriangle 。

Mads 是一种搜索-轮询直接搜索迭代算法，每次迭代由两个主要步骤组成。搜索步骤可以使用各种策略（包括使用替代函数和启发式方法）来探索变量空间。

轮询步骤遵循更严格的规则，在轮询大小向量 $\Delta^k \in \mathbb{R}^n$ 所限定区域内的变量空间进行局部探索。在实践中，一个精心设计的搜索步骤可以让算法摆脱局部最优解的束缚

(例如，[1,7])，轮询步骤确保了实际和理论上的收敛性[2]。

这两个步骤都是在被称为网格的变量空间离散化上生成候选点。离散化的精细度由网格大小向量 $\delta^k \in \mathbb{R}^n$ 控制。形式上，迭代 k 时的网格定义如下：

$$M^k = \{V^k + \text{diag}(\delta^k)z : z \in \mathbb{Z}^n\}。 \tag{2}$$

其中，缓存 V^k 是算法在迭代 k 开始时访问过的点的集合。换句话说，当且仅

当 $f(x)$ 和 $h(x)$ 在迭代 k 开始时都被求值时, 点 x 才属于缓存 V^k 。第一个集合 V^0 是用户提供一个或多个初始点的集合。

网格和投票大小向量在每次迭代结束时更新。如果迭代无法改进现有解决方案, 则会减少这两个向量的条目, 否则就会增加或保持相同的值。算法 2

提供了对 Mads 的高层次描述；有关详细描述，请读者参阅 [8]。

算法 2：网格自适应直接搜索算法（Mads）

给定一组用户定义的起点： $V^0 \subset \mathbb{R}^n$
以及初始网格和轮询大小向量：通常为 $\delta^0 = \Delta^0 = \frac{1}{i}, i = 1, 2, \dots, n$ 。

设置迭代计数器： $k \leftarrow 0$ 。

1. 搜索步骤（可选）：

在网格点的有限集 S^k 上启动模拟。如果成功，转到 3。

2. 投票步骤：

在投票点集合 P^k 上启动模拟。

3. 更新：

更新缓存 V^{k+1} 、现任 x^{k+1}
以及网格和轮询大小向量 δ^{k+1} 和 Δ^{k+1} 。将迭代计数器 $k \leftarrow k + 1$ 增加到 1。

马德斯算法的基本收敛结果[3]指出，如果整个试算点序列属于一个有界集合，且所谓的细化方向集合足够丰富，则存在一个累积点 x^* ，使得认为克拉克[13]的广义方向导数 $f^\circ(x^*; d)$ 在只要 x^* 可行，在 x^* 处域 \blacktriangle 的每个高切线 [24] 方向 d 都是可行的。类似的结果（涉及 h 而非 f ）也适用于迭代的情况。

永远不会接近可行区域：如果累积点 x^* 不可行，那么广义方向导数 $h^\circ(x^*; d)$ 在 x 处的集合 X 的每个高切线方向 d 上都是非负的。*

这些收敛结果与算法 2 的搜索步骤无关，因为只要集合 S^k 是有限的，并且属于网格 M^k 。在本工作中，我们确保满足这些条件，从而使收敛分析不发生变化。

4 Mads-NM 约束优化算法

现在我们介绍受 NM 启发的 Mads 搜索步骤。第 4.1 节 将支配关系 \prec 和函数 Best 的定义扩展到受约束的情况。第 4.2 节重点介绍需要调整的主要 Mads 要素，以便实现 NM 搜索功能。第 4.3 节介绍了 NM 搜索步骤的算法规范。

这种新算法的设计共涉及三个参数，其默认值将在第 5 节中调整。这些参数用 $\pi = (\pi_{radius}, \pi, \pi_{svdeval})$ 表示。它们指的是采样半径、奇异值阈值和阈值。分别是函数评估的次数。

4.1 订购试验点

无约束优化的 NM 算法基于将顶点从最佳到最差排序的概念，每次迭代时，算法都会尝试替换最差的顶点。在没有约束条件的情况下，给单纯形中的点排序很容易，因为只需要比较目标函数值。在有约束的优化中，排序就变得更加有技术含量了。为了定义试算点之间的传递关系，需要对优势的定義以及返回两点最佳值的函数进行调整。这些定义依赖于目标函数和违反约束函数 f 和 h 。

定义 4.1 如果出现以下情况，则称点 $x \in \mathbb{R}^n$ 支配 $y \in \mathbb{R}^n$ ，记为 $x < y$ 。

- 两点都可行，且 $f(x) < f(y)$ ；或
- 两点都不可行，且 $f(x) \leq f(y)$ 和 $h(x) \leq h(y)$ 至少有一个严格不等式。

上述定义与渐进式条形障碍[3]中使用的术语一致，其中可行点和不可行点的处理方法不同。

在两个试验点的函数值相同的情况下，即 $f(x) = f(y)$ 和 $h(x) = h(y)$ ，我们不能对它们任意排序，而是要再次使用定义 2.2 中的确定性函数老式函数，返回两个点的最老值。

结合优势的定義，我们可以利用函数 Older 引入以下规则来比较两个试验点 x 和 y 。

定义 4.2 函数 Best : $\mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}^n$

$$\text{Best}(x, y) = \begin{cases} x & \text{如果 } x < y \text{ 或如果 } h(x) < h(y) \\ y & \text{如果 } y < x \text{ 或如果 } h(y) < h(x) \\ \text{Older}(x, y) & \text{如果 } f(x) = f(y) \text{ 和 } h(x) = h(y) \end{cases}$$

返回 \mathbb{R}^n 中两个点 x 和 y 的最佳值。一个点 $x \in \mathbb{R}^n$ 称为优于 $y \in \mathbb{R}^n$ ，如果 $x = \text{Best}(x, y)$ 。

请注意，根据这一定义，可行点 x 与不可行点 y 的比较结果总是 $x = \text{Best}(x, y)$ 。比较两个不可行点时，如果其中任何一个点都不占优势，则根据 h 的值返回最可行的一个点。

命题 4.3 函数 Best 在一组试验点上具有传递性。

证明 考虑 \mathbb{R}^n 中的三个试探点 x 、 y 和 z ，使得 $x = \text{Best}(x, y)$ 和 $y = \text{Best}(y, z)$ 。我们需要证明 $x = \text{Best}(x, z)$ 。分析分为四种情况。

• 首先, 如果 x 和 y 都是可行的, 那么 z 也是可行的, 因此 $x = \text{Best}(x, z)$.

11

- 其次, 如果 x 和 z 都不可行, 则 $0 < h(x) \leq h(y) \leq h(z)$ 。
因此, $z = \text{Best}(x, z)$ 的唯一可能是

$$f(x) = f(y) = f(z), h(x) = h(y) = h(z), z = \text{Older}(x, z)。$$

但如果是这样的话，这就与 $x = \text{Older}(x, y)$ 和 $y = \text{Older}(x, y)$ 相矛盾了。

$\text{Older}(y, z)$ 。

- 第三，如果 x 是可行的，而 z 是不可行的，那么在命题陈述之前的观察结果确保 $x = \text{Best}(x, z)$ 。
- 最后， x 不可行而 z 可行的情况是不可能的，因为如果是这种情况，那么 y 就不可行，因为 $x = \text{Best}(x, y)$ ，而 y 也不会不可行，因为 $y = \text{Best}(y, z)$ 。 \square

4.2 在网格上构建单纯形

第 4.3 节定义了受 NM 算法启发的 Mads 搜索步骤。这个 NM 搜索步骤使用了 [14] 中提出的二次模型搜索步骤中的元素。在该论文中，建立了目标和每个约束函数的二次模型，并用于识别候选试验点。但请注意，Mads 算法的一个重要要求是每个候选点都属于网格 M^k 。本文采用以下符号。让 x 是 \mathbb{R}^n 中的某个点。

下标 \oplus ，我们表示 $x_{\oplus} \in M^k$ 为最接近 x 的网格点（通过向上舍入打破平局），详见 [6]。我们会说 x_{\oplus} 是被舍入到网格中的点 x 。[14]的二次模型是通过考虑所有的

以前访问过的试验点位于以现存解决方案为中心的区域内，其大小与 Δ^k 有关。本研究也考虑了类似的区域。

在渐进障碍算法[3]中，最多有两个现任解，分别称为*主要*投票中心和*次要*投票中心。其中一个是该算法迄今为止发现的最佳可行解，另一个是 h 值最小的不可行解。在构建单纯形时，只考虑主投票中心。给定现任解 x^k 、投票大小向量 Δ^k 和缓存 V^k ，定义集合

$$T_{\pi \text{ 半径}} = \{x \in V^k : |x_i - x^k| \leq \pi \text{radius} \Delta_i^k, i \in \{1, 2, \dots, n\}\}$$

其中 $\pi \text{radius} \geq 1$ 是一个固定参数，称为采样半径（它是三个新算法参数中的第一个）。与算法 1 一样，对 $T_{\pi \text{ 半径}}$ 的点重新排序，使 $i = 1, 2, \dots, p$ 时， $x^{i-1} = \text{Best}(x^{i-1}, x^i)$ 。

算法 3 利用集合 $T_{\pi \text{ 半径}}$ ，尝试迭代构建单纯形 Y 。

它还使用了三个新算法参数中的第二个参数 π_{svd} 。算法
 π_{0i} 首先，构建由 $\{y - x^0 : y \in Y, y \neq x^0\}$ 列和 $\{x^i - x^0\}$ 列组成的矩阵 A 。其次
 ，为了确保最终的单纯形远离
 相对于投票规模向量的退化，要求矩阵 $\text{diag}(\Delta)^{k-1} A$ 的每个奇异值都超过给定
 阈值 π_{svd} 。

算法 3：构建有序单纯形 Y

给定参数 $\pi_{\text{radius}} \geq 1$ 、 $\pi_{\text{svd}} > 0$ 和集合 $T_{\pi_{\text{radius}}} = \{x^0, x^1, \dots, x^p\}$ 。

1. 构造单纯形：
 重新排列 $T_{\pi_{\text{radius}}}$ 并设置 $Y = \{x^0\}$ 。
 For $i = 1$ to p do
 让 A 由 $\{y - x^0 : y \in Y, y \neq x^0\} \cup \{x^i - x^0\}$ 的列组成，如果
 $\text{diag}(\Delta^k)^{-1} A$ 的所有奇异值 $\geq \pi_{\text{svd}}$ ，则设 $Y \leftarrow Y \cup \{x^i\}$

2. 终止：
 如果 $|Y| = n + 1$ 则返回 Y ，否则
 返回 \emptyset 。

4.3 NM 搜索步骤

现在，我们介绍并描述将插入算法 2 的 NM 搜索步骤。给定采样半径 $\pi_{\text{radius}} \geq 1$ 和阈值参数 $\pi_{\text{svd}} > 0$ ，应用算法 3 尝试建立一个有序单纯形 $Y = \{y, y^{01}, \dots, y^n\} \subset \mathbb{R}^n$ 以启动搜索。如果尝试的结果是 $Y = \emptyset$ ，则放弃 NM 搜索步骤。

Mads-NM 算法将遍历并尝试替换 Y 中的 y^n ，生成与标准 NM 方法一样，这些新的暂定点可以是中心点、反射点、外收缩点和/或内收缩点。然而，为了满足 Mads 关于每个试验点都属于当前网格的要求，以下符号将这些新的暂定点舍入网格：

x_{\oplus}^r 是按网格四舍五入的反射 x^r ；

x_{\oplus}^e 是按网格四舍五入的扩展 x^e ；

x_{\oplus}^{oc} 是网格的外收缩 x^{oc} ；

x_{\oplus}^{ic} 是内收缩 x^{ic} ，按网格四舍五入。

请注意，中心点 x^c 不需要对网格进行舍入，因为

NM 算法从不在中心点评估函数值。

在无约束 NM 算法中，图 1 展示了用于比较试验点的四个区域。每个区域的端点由最好的单纯形顶点 y^0 或最差的单纯形顶点 y^n 限定。现在，在有约束条件的情况下，我们重新定义这些区域，使它们同时反映目标函数 f 和违反约束条件函数 h 的贡献。为了确定在这些试验点中的哪个点评估函数值，我们引入了单纯形 Y 的以下子集：

$$Y^0 = \{y \in Y : \nexists x \in Y \text{ with } x < y\}$$

$$Y^n = \{y \in Y : \nexists x \in Y \text{ with } y \prec x\} \circ \quad (3)$$

集合 Y^0 包含不被任何其他顶点支配的所有顶点。集合 Y^n 包含不被任何其他顶点支配的所有顶点。这些集合都不可能是空的，也不一定互不相交。同样的逻辑也可以应用于无约束优化的 NM：点 y^0 是无支配的，而 y^n 是不支配任何其他顶点的单纯形点。

下面是定义 2.4 的对应部分，但针对的是约束优化。

它采用了支配地位的新定义（定义 4.1）。

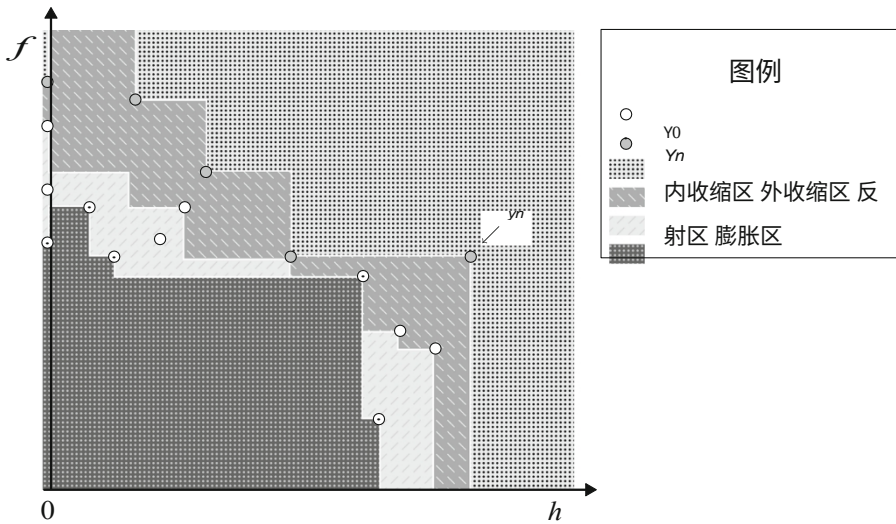


图 2 相对于 16 点单纯形的区域

定义 4.4 (约束优化的区域) 设 $Y = \{y, y^{01}, \dots, y^n\}$ 是 R^n 中的有序单纯形，设 Y^0 和 Y^n 是方程 (3) 中 Y 的子集。试验点 $x \in R^n$

- 如果 $y^i < x$ 对于某些 $y^i \in Y^n$ ，或者 $h(x) > 0$ ，则属于内收缩区。
 $h(y^n)$ ；
- 否则，如果对于某个 $y^i \in Y^0$ ， $x < y^i$ ，它就属于扩展区；
- 否则，如果 x 至少支配 Y 的 2 个点，它就属于反射区；
- 否则，它属于外收缩区， x 主导 Y 的 0 或 1 点。

图 2 是图 1 在受约束情况下的对应图。它不再只在实半线上描述目标函数，而是考虑到了半平面，在半平面上，横坐标代表非负的违反约束条件函数 h ，纵坐标代表

是目标函数 f 。图 2 展示了有 16 个元素的单纯形上的集合 Y^0 和 Y^n 。用符号 \odot 表示的 6 个点是不占优势的顶点，用符号 \bullet 表示的 5 个点是不占优势的顶点。

另一个

算法决定反射点 x^r (图 2 中的扩展区)， x^r 由 Y^n 的一个点支配 (图 2 中的内侧)。收缩区)， x^r 主导 2 个或更多点 (反射区)，或 x^r 主导

0 或 1 点 (外部收缩区域)。请注意，定义 4.1 并不比较可行点和不可行点，因此图被分为两部分。图的左边部分表示可行点的区域，即 h 等于零的点。图中其余部分为不可行点。

NM 搜索步骤可简洁地写成算法 4。

基于网格的Nelder-Mead不等式算法³⁴¹之间是对称的，但三个重要区别。首先，由于对网格进行舍入操作，在替换顶点 y^n 后， Y 有可能不再是一个单纯形。如果出现这种情况

算法 4：内尔德-梅德搜索步骤（NM-搜索）

给定初始有序单纯形 $Y = \{y^0, y^1, \dots, y^n\} \subseteq V^k$ 的顶点。

- 更新单纯形：
 - 如果 Y 不是单项式，则转到 4；
 - 否则对 Y 进行重新排序，构建 Y^0 和 Y_n 集。
- 确定一个新的候选顶点 t ：
 - 如果 x^r 属于内收缩区
 - 如果 x^{ic} 属于内收缩区，则转到 4、
 - 否则设 $t = x^{ic}$ ；
 - 否则，如果 x^r 属于扩展区，则设 $t = \text{Best}(x^r, x^e)$ ；
 - 否则，如果 x^r 属于反射区，则设置 $t = x^r$ ；
 - 否则，如果 x^r 属于外部收缩区，则设置 $t = \text{Best}(x^r, x^{oc})$ 。
- 测试并更换最差点：
 - 如果 $t \in V^k$ ，则转到 4、
 - 否则设置 $y^n \leftarrow t$ ，并转到 1。
- 终止：
 - 返回在步骤 2 中访问过的试验点集合。

则 NM 搜索终止。其次，替代顶点 y^n 的建议值有可能是之前访问过的网格点 t ($t \in V^k$)。为防止出现循环问题，将终止 NM 搜索。第三，不需要的收缩参数 γ ，因为 NM 搜索步骤终止而不是收缩单纯形。不缩小单纯形有两个原因：-i 可能会引入多达 n 个新的试验点，这将大大增加搜索步骤的成本；
-经常会生成不在当前网格上的点。

当 NM 搜索步骤结束后，Mads 算法将进入轮询步骤，详见算法 2。在下次 Mads 迭代中，NM-搜索步骤将从算法 3 确定的新单纯形开始，其中包含投票步骤中产生的额外点。

除了参数 π_{radius} 和 π_{svd} 之外，我们还引入了第三个算法参数，称为 $\pi_{eval} \in \mathbb{N}$ ，它限制了每个 NM 搜索步骤可以执行的函数求值次数。实际上，这意味着算法的第 4 步是

π_{eval} 函数求值后立即调用。每个新的 NM 搜索步骤都会重置计数器。为便于阅读，算法 4 中未介绍该参数。在下一节的数值实验中，常规 NM 参数设定值为 $\delta^e = 2$, $\delta^{ic} = -1$, $\delta^{oc} = 1$ ，Mads 算法与算法 4 的 NM 搜索步骤用 Mads-NM 表示。

5 计算实验

NOMAD[29]软件包可从 www.gerad.ca/nomad 免费获取。所有测试都使用了具有 $n + 1$ 轮询方向的 Mads 策略[6]，无论是否使用二次模型。当同时启用二次模型和 NM 搜索时，结果如下

首先执行前者，如果迭代成功，则伺机终止迭代。

下文将介绍数据概况[36]，以评估算法是否能成功生成接近最佳目标函数值的解值。识别成功运行需要进行收敛测试。我们将 x_e 表示为算法对其中一个问题进行 e 次评估后得到的最佳可行迭代值。

当出现以下情况时，问题可在收敛容差 τ 内解决

$$f_{\text{fea}} - f(x_e) \geq (1 - \tau) f_{\text{fea}} - f^*$$

其中， f^* 是所有测试算法在该问题所有运行实例上获得的最佳可行点的目标函数值。 f_{fea} 的值是一个常见的

对某一问题的参考值，是在该问题的所有运行实例中，通过对所有算法的第一个可行目标函数值取平均值而得到的。如果没有获得可行的迭代，则收敛测试失败。

通过使用不同的伪随机生成器种子复制算法运行，可以获得不同的实例。在本研究中，我们认为不同的初始点构成了不同的问题。

数据剖面的横轴代表固定维度问题的评价次数，当涉及不同维度的问题时，横轴代表 $n+1$ 个评价组。纵轴对应于在给定容差 τ 内解决的问题的数量。每种算法都有自己的曲线，以便比较算法收敛到最佳目标函数值的能力。

数值实验的方法如下。

第 5.1 节对分析问题进行了测试，以调整算法参数。第 5.2 节比较了在无约束测试问题上的 NM 算法与几种著名的 NM 算法的实施情况。最后，第 5.3、5.4 和 5.5 节对算法进行了测试。最后，在第 5.3、5.4 和 5.5 节中对该算法在三个受限黑箱工程测试问题上进行了测试。

5.1 校准参数的初步实验

对分析测试问题进行了数值实验，以设定三个算法参数的默认值：采样半径 π_{radius} ，奇异值

每个 NM 调用的阈值 π_{svd} 和评估预算 π_{eval} 。在下文所有图表中，这三个参数被紧凑地写成三元组 $\pi = (\pi_{\text{radius}}, \pi_{\text{svd}}, \pi_{\text{eval}})$ 。

对 87 个分析问题进行了测试。

优化文献。表 1 总结了这些问题的特点和来源。变量数量从 2 个到 20 个不等

；19 个问题有约束条件以外的其他约束条件。

第一组运行仅通过改变奇异值阈值 $\pi_{\text{svd}} \in \{0.001, 0.005, 0.01, 0.05, 0.1\}$ ，同时固定 $\pi_{\text{eval}} = 20n$ 和 $\pi_{\text{radius}} = 4$ 。参数 π_{svd} 对 Mads-NM 的性能有重要影响、

$\pi_{\text{svd}} = 0.01$ 时性能最佳。在第二组运行中， $\pi_{\text{eval}} \in \{5n, 10n, 20n, 40n, 80n, 160n\}$ 的值固定不变，而评估预算参数则有所变化。当数值大于 $20n$ 时，该参数对性能的影响有限， $\pi_{\text{eval}} = 80n$ 是最佳值。在第三组运行中，奇异的

表 1 87 个分析问题集的说明

不	名称	资料来源	<i>n</i>	<i>m</i>	Bnds	不	名称	资料来源	<i>n</i>	<i>m</i>	Bnds
1	ARWHEAD10	[21]	10	0	没有	45	PENALTY1_4	[21]	4	0	没有
2	ARWHEAD20	[21]	20	0	没有	46	PENALTY1_10	[21]	10	0	没有
3	巴德	[35]	3	0	没有	47	处罚1_20	[21]	20	0	没有
4	BDQRTIC10	[21]	10	0	没有	48	处罚2_4	[21]	4	0	没有
5	BDQRTIC20	[21]	20	0	没有	49	PENALTY2_10	[21]	10	0	没有
6	贝勒	[35]	2	0	没有	50	判罚2_20	[21]	20	0	没有
7	BIGGS	[21]	6	0	没有	51	五角大楼	[30]	6	15	没有
8	BOX	[35]	3	0	没有	52	PIGACHE	[39]	4	11	是
9	BRANIN	[22]	2	0	是	53	波拉克2	[30]	10	0	没有
10	BROWNAL5	[21]	5	0	没有	54	POWELL_BS	[35]	2	0	没有
11	BROWNAL7	[21]	7	0	没有	55	POWELLSG4	[21]	4	0	没有
12	BROWNAL10	[21]	10	0	没有	56	POWELLSG8	[21]	8	0	没有
13	BROWNAL20	[21]	20	0	没有	57	POWELLSG12	[21]	12	0	没有
14	布朗登尼斯	[35]	4	0	没有	58	POWELLSG20	[21]	20	0	没有
15	BROWN_BS	[35]	2	0	没有	59	雷达	[34]	7	0	是
16	CHENWANG_F2	[12]	8	6	是	60	拉纳	[25]	2	0	是
17	CHENWANG_F3	[12]	10	8	是	61	拉斯提金	[22]	2	0	是
18	CRESCENT	[3]	10	2	没有	62	罗斯布洛克	[35]	2	0	是
19	磁盘	[3]	10	1	没有	63	SHOR	[30]	5	0	没有
20	ELATTAR	[30]	6	0	没有	64	蛇	[3]	2	2	没有

表 1 继续

不	名称	资料来源	n	m	Bnds	不	名称	资料来源	n	m	Bnds
21	EVD61	[30]	6	0	没有	65	春季	[41]	3	4	是
22	过滤器	[30]	9	0	没有	66	SROSENBR6	[21]	6	0	没有
23	Freudensteinroth	[35]	2	0	没有	67	SROSENBR8	[21]	8	0	没有
24	高加索语	[35]	3	0	没有	68	SROSENBR10	[21]	10	0	没有
25	G210	[4]	10	2	是	69	SROSENBR20	[21]	20	0	没有
26	G220	[4]	20	2	是	70	特雷费亨	[25]	2	0	是
27	格里旺克	[22]	10	0	是	71	TRIDIA10	[21]	10	0	没有
28	海湾区	[35]	3	0	没有	72	TRIDIA20	[21]	20	0	没有
29	鹤立谷	[35]	3	0	没有	73	触发式	[35]	10	0	没有
30	HS19	[23]	2	2	是	74	VARDIM8	[21]	8	0	没有
31	HS78	[30]	5	0	没有	75	VARDIM10	[21]	10	0	没有
32	HS83	[23]	5	6	是	76	VARDIM20	[21]	20	0	没有
33	HS114	[30]	9	6	是	77	WANGWANG_F3	[46]	2	0	是
34	jennrichsampsom	[35]	2	0	没有	78	WANGWANG_F5	[46]	2	0	是
35	科瓦利科斯堡	[35]	4	0	没有	79	瓦特森9	[35]	9	0	没有
36	MAD6	[30]	5	7	没有	80	WATSON12	[35]	12	0	是
37	麦金农	[32]	2	0	没有	81	黄1	[30]	7	0	没有
38	MDO	[44]	10	10	是	82	黄2	[30]	10	0	没有
39	梅兹蒙特斯	[33]	2	2	是	83	木材4	[21]	4	0	没有
40	梅耶尔	[35]	3	0	没有	84	WOODS12	[21]	12	0	没有

41	OPTENG_RBF	[27]	3	4	是	85	WOODS20	[21]	20	0	没有
42	OSBORNE1	[35]	5	0	没有	86	TAOWANG_F2	[43]	7	4	是
43	OSBORNE2	[30]	11	0	没有	87	ZHAOWANG_F5	[47]	13	9	是
44	PBC1	[30]	5	0	没有						

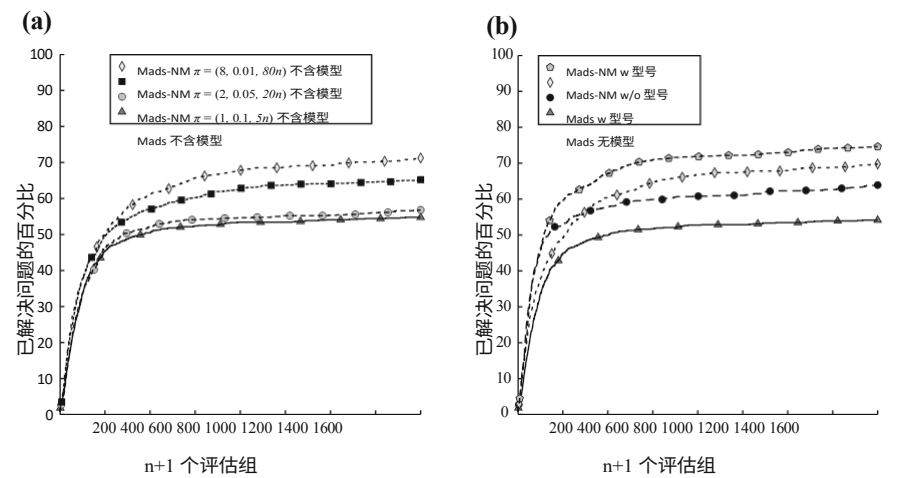


图 3 在收敛容差 $\tau = 10^{-5}$ 的条件下，对 87 个分析问题进行 10 次重复，采用不同的 NM 设置（左）和二次方模型（右）获得的数据剖面。
有模型和无模型的 π 和 \mathbf{b} 比较

值阈值和评估预算参数固定为： $\pi_{\text{svd}} = 0.01$ 和 $\pi_{\text{eval}} = 80n$ ，采样半径可变： $\pi_{\text{radius}} \in \{1, 2, 4, 8, 16\}$ 。 $\pi_{\text{半径}} \in \{8, 16\}$ 的性能相似。我们选择 $\pi_{\text{radius}} = 8$ 。最后，分别改变 π_{svd} 和 π_{eval} 的结果也没有改善。这些密集的计算测试

由于可以获得 87 个测试问题的分析公式，因此可以对这些问题进行分析。图 3a 显示了针对 Mads-NM 的三组不同参数获得的具有代表性的数据曲线 $\diamond\diamond\diamond\diamond$ 。在 $\tau = 10^{-3}$ 和 $\tau = 10$ 时，也得到了类似的曲线图。 10^{-7} ，但为便于阅读，不一一列出。通过这些实验，我们可以得出以下结论

$$\pi = (\pi, \pi_{\text{svdeval}}, \pi_{\text{radius}}) = (0.01, 80n, 8)$$

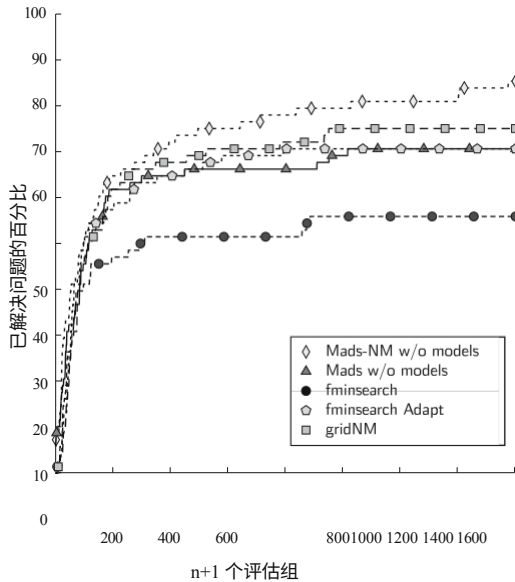
对于当前的问题来说，这是一组令人满意的参数，因此我们在其余所有测试中都使用这些参数。

图 3b 展示了第二系列测试，以评估使用和不使用二次模型的 Mads- NM 性能。结果表明，Mads- NM 的性能优于 Mads。令人惊讶的是，当函数求值次数为 0 时，不使用二次模型的 Mads-NM 比使用二次模型的 Mads 性能更好。

超过 $\sim 350 \times (n + 1)$ 。在这些分析问题上，结合使用二次模型和 NM 搜索在所有有其他算法变体中都占优势。

Mads-NM 算法可用于无约束优化问题，并可与 Matlab 的 `fminsearch` 算法进行比较，后者是 NM 算法的一种实现 [28]，`fminsearch Adapt`（与 `fminsearch` 的 Matlab 代码相同，但扩展、收缩和缩小参数取决于问题的维度 n ，如图所示

图 4 使用 Mads-NM、Mads、fminsearch、fminsearch Adapt 和 gridNM 的数据曲线，收敛容差为 $\tau = 10^{-5}$ ，对一个重复的 68 个测试问题没有非约束条件



[20]) 和 gridNM (一种网格约束 NM 算法的实现 [10])。为了进行比较，我们从之前的 87 个分析问题中提取了除边界 (表 1 中 $m = 0$) 外没有其他约束条件的问题。这得出 68 个测试问题的子集。

fminsearch 算法只考虑一个初始点，并自动从中创建初始单纯形，该算法不依赖于随机数生成器。gridNM 算法和 fminsearch Adapt 算法的初始单纯形选择与 fminsearch 算法相同。为了进行公平比较，所有测试问题只运行一次 Mads 算法和 Mads- NM 算法。

图 4 中的数据曲线表明，gridNM、fminsearch Adapt 以及不带二次模型的 Mads 和 Mads-NM 均优于 Matlab 的 fminsearch。同样，Mads-NM 是最主要的算法。

5.3 洛克伍德的问题

测试了带有二次模型的 Mads 算法和 Mads-NM 算法，以解决蒙大拿州洛克伍德溶剂地下水羽流遗址[31]的抽水处理地下水修复问题的基本版本。每个初始点都定义了一个 LOCK- WOOD 问题。要解决这个问题，就必须使运行成

件是：两个污染物羽流的通量，这两个污染物羽流的通量来自于

蓝鸟模拟器 [16]。该问题有 6 个设计变量，边界为 $[0; 20000]^6$ 。图 5 显示了

解决 20 个 LOCKWOOD 问题时的优化历程

从超矩形 $[0; 20000]^6$ 中随机选择的不同初始点出发。图中绘制了现存解值随

功能评估。右侧图放大了目标函数值的较低值。对于 5 个初始点，Mads-NM

算法未能到达可行区域，而 Mads 算法则失败了 4 次。不过，从图中可以看

出，Mads-NM 算法找到了更多的可行区域。

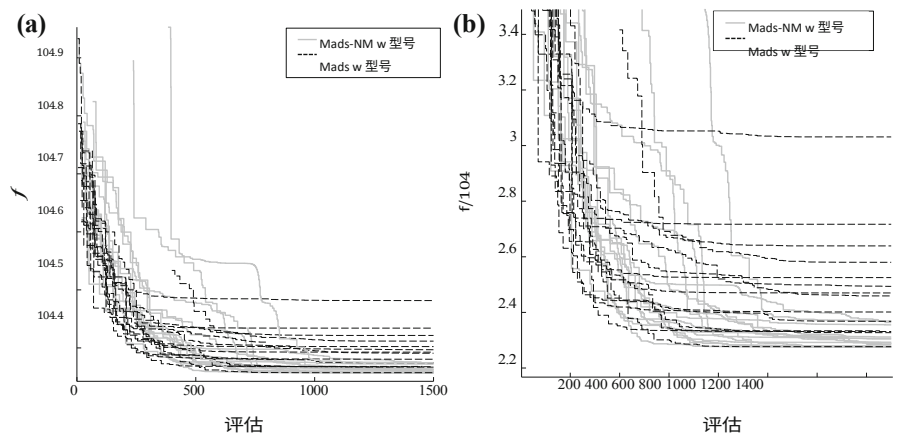


图 5 20 个 LOCKWOOD 问题的优化历史 (右图为低目标值放大图)

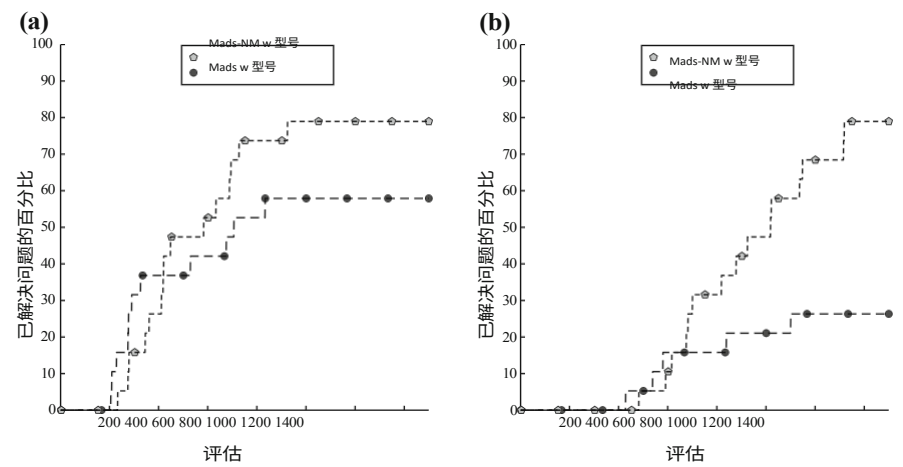


图 6 在 20 个 LOCKWOOD 问题上以收敛容差 τ 获得的数据曲线。a $\tau = 10^{-1}$ 和 b $\tau = 10^{-2}$

目标函数值低于 Mads 的解决方案。从图 6 的数据曲线中也可以明显看出这一点。

5.4 MDO 的问题

在解决一个简单的多学科机翼设计优化问题时，对具有二次模型的 Mads 算法和 Mads-NM 算法进行了测试 [44]。每个初始点都定义了一个 MDO 问题。该问题的求解包括在 10 个约束条件下最大化飞机的航程。该问题有 10 个按

比例分配的设计变量，它们的边界是

在超矩形 $[0; 100]^{10}$ 中。

图 7a 显示了解决 20 个 MDO 问题时的优化历程。

使用 1500 次或更少的函数求值来输入初始点。初始点为整数

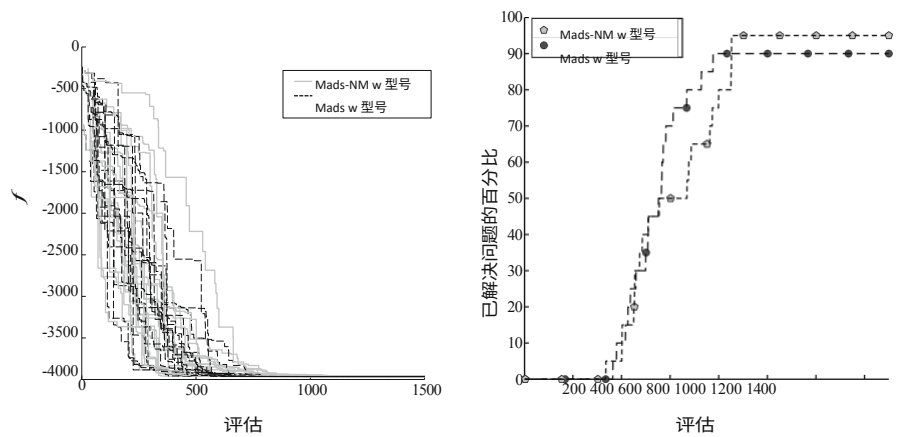


图 7 在 20 个 MDO 问题上使用二次模型的 Mads 和 Mads-NM a 优化历史和 b $\tau = 10^{-2}$ 的数据剖面

在边界内随机选择。对于 1 个初始点，Mads-NM 算法未能达到可行区域，而 Mads 算法则失败了 2 次。图 7 中的优化历程图和数据剖面图显示，Mads-NM 算法和 Mads 算法的行为相似。在评估次数较多的情况下，Mads-NM 解决的问题略多于 Mads。

与前几小节不同的是，对 MDO 问题的数值实验并没有发现一种占主导地位的算法。

5.5 STYRENE 的问题

对带有二次模型的 Mads 算法和 Mads-NM 算法进行了测试，以优化苯乙烯生产过程[1]。化学过程的模拟依赖于一系列相互依存的计算模块，使用的是常见的数值方法，如 Runge-Kutta、牛顿、定点以及与化学相关的求解器。一旦工艺模拟成功完成，就可以在后处理过程中对约束条件和目标进行评估。目标是使苯乙烯生产过程的净现值最大化，同时受到 9 项工业和环境法规的限制。模拟和后处理结合在一个黑盒中。如果化学过程的模拟不能成功迭代，则返回黑盒评估失败。

在这项工作中，STYRENE 问题包含 8 个影响苯乙烯生产过程的独立变量。优化过程中考虑的变量都在 $X = [0; 100]^8$ 中按比例分配并有边界。作为参

x^{ref} 。从这一参考点出发, 已完成了 20 个 STYRENE 问题

在以 x^{ref} 为圆心、半径为 10 的超矩形中随机生成整数个初始点 (如有必要, 可将这些点投影到 X 上)。对于所有初始点, Mads 算法和 Mads-NM 算法都能从每个初始点到达可行区域。Mads 算法和 Mads-NM 算法的运行历史图均使用了 1500 次评估或

图 8 20 个 STYRENE 问题的
优化历程

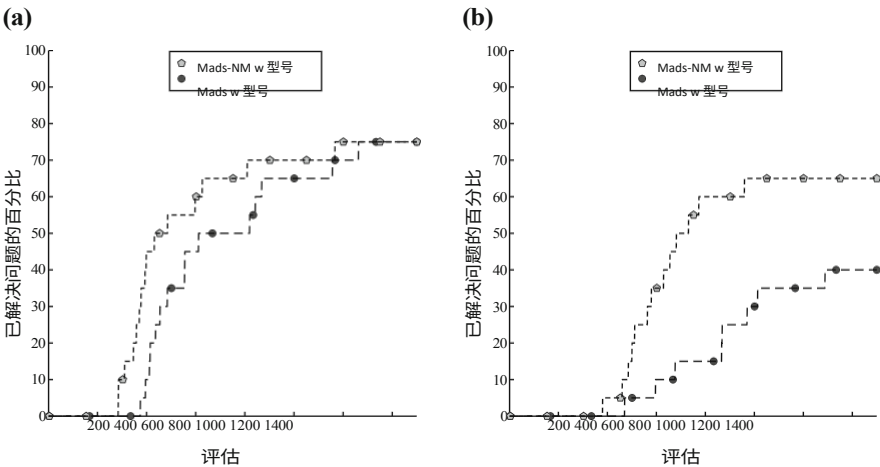
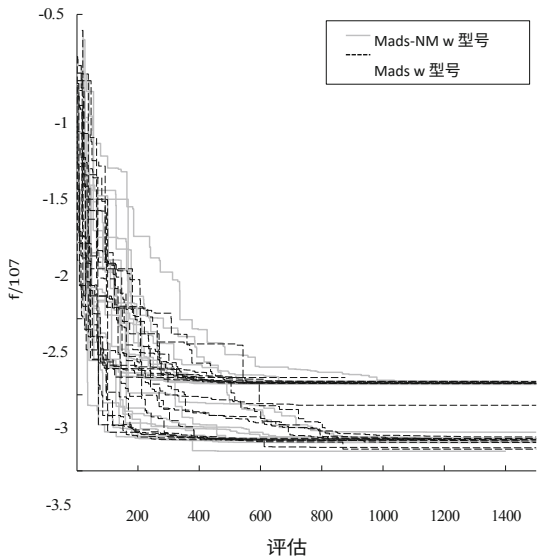


图 9 对 20 个 STYRENE 问题采用两种收敛公差获得的数据曲线。a $\tau = 10^{-2}$ 和 b $\tau = 10^{-3}$

图 8 显示了不同算法的局部最小值。各种算法都能达到不同的局部最小值，但图中并没有明确指出哪种算法更优。

图 9 显示了收敛公差为 $\tau = 10^{-2}$ 和 10^{-3} 的数据曲线。 $\tau = 10^{-3}$ 时的剖面图显示，Mads-NM 算法比 Mads 算法占优势。

6 讨论

约束。具体方法是在

Mads 算法。NM 搜索点构成简约，由涉及目标函数值和违反约束函数值的传递关系排序。简图的反射、扩展和收缩与原始算法相同，只是试验点被舍入到当前网格中。为了继承丰富的 Mads 分层收敛分析，最后一项操作是必要的。一旦顶点无法形成单纯形，或者顶点被之前访问过的点取代，就会中断每个 NM 搜索步骤。由此产生的算法适用于无约束和不等式约束的黑箱优化问题。

对无约束优化问题的数值实验表明，由此产生的 Mads-NM 算法优于 Matlab 实现的 NM 算法（fmin-search 和 fminsearch Adapt）、gridNM 算法以及之前使用或不使用二次模型的 Mads 算法。在三个受限工程问题上的实验表明，Mads-NM 算法和 Mads 算法在最初的数百次函数求值过程中表现相当。但随着函数求值次数的增加，Mads-NM 算法通常比 Mads 算法能找到更好的可行解。解决问题百分比的差异

对于 MDO 问题（ $\tau = 10^{-2}$ ），数据剖面的误差约为 5%；对于 STYRENE 问题（ $\tau = 10^{-3}$ ），误差为 25%；对于 LOCK- WOOD 问题（ $\tau = 10^{-2}$ ），误差超过 50%。

测试版 3.8.2 NOMAD 黑盒优化软件使用 NM

本文所述的搜索步骤，并包含所有考虑过的测试问题 及其参考初始点。

致谢 本研究得到了国家科学研究中心 CRD 补助金（#RDCPJ 490744 - 15）的支持，合作方包括魁北克水电公司（Hydro-Québec）和力拓集团（Rio Tinto）。作者感谢 Shawn Mattot、Genetha Gray 和 Stefan Wild 提供 LOCKWOOD 问题。

参考资料

1. Audet, C., Béchard, V., Le Digabel, S.: Nonsmooth optimization through mesh adaptive direct search and variable neighborhood search. *J. Global Optim.* **41**(2), 299-318 (2008)
2. Audet, C., Dennis Jr., J.E.: 约束优化的网格自适应直接搜索算法. *SIAM J. Optim.* **17**(1), 188-217 (2006)
3. Audet, C., Dennis Jr., J.E.: A progressive barrier for derivative-free nonlinear programming. *SIAM J. Optim.* **20**(1), 445-472 (2009)
4. Audet, C., Dennis Jr., J.E., Le Digabel, S.: 网格自适应直接搜索算法的并行空间分解. *SIAM J. Optim.* **19**(3), 1150-1170 (2008)
5. Audet, C., Hare, W.: 无衍生和黑箱优化. *Springer Series in Operations Research and Financial Engineering*. Springer, Berlin (2017)
6. Audet, C., Ianni, A., Le Digabel, S., Tribes, C.: 减少网格自适应直接搜索算法中的函数评估次

- 数。SIAM J. Optim.**24**(2), 621-642 (2014)
7. Audet, C., Kokkolaras, M., Le Digabel, S., Talgorn, B.: 在无衍生优化中管理代理集合的基于阶次的误差。J. Global Optim.**70**(3), 645-675 (2018)
 8. Audet, C., Le Digabel, S., Tribes, C.: 黑箱优化的 网格自适应直接搜索算法中的动态缩放。Optim.Eng.**17**(2), 333-358 (2016)
 9. Brea, E.: Nelder-Mead 方法在非线性混合整数优化问题上的扩展。Rev. Int.Métod.Númer.**29**(3), 163-174 (2013)
 10. Buřmen, Ā., Puhan, J., Tuma, T.: Grid restrained Nelder-Mead algorithm.Comput.Optim.**34**(3), 359-375 (2006)

11. Chang, K.H.: Stochastic Nelder-Mead simplex method-a new globally convergent direct search method for simulation optimization.Eur.J. Oper.**220**(3), 684-694 (2012)
12. Chen, X., Wang, N.: 基于 DNA 的混合遗传算法优化短时汽油调配问题。Chem.Eng.Process.**49**(10), 1076-1083 (2010)
13. Clarke, F.H.: Optimization and Nonsmooth Analysis.Wiley, New York (1983).(1990 年由费城 SIAM 出版社作为《应用数学经典》系列第 5 卷再版)。
14. Conn, A.R., Le Digabel, S.: Use of quadratic models with mesh-adaptive direct search for constrained black box optimization.Optim.Methods Softw.**28**(1), 139-158 (2013)
15. Coope, I.D., Price, C.J.: 基于框架的无约束优化方法。J. Optim.Theory Appl.**107**(2), 261-274 (2000)
16. Craig, J.: Bluebird developer manual. http://www.civil.uwaterloo.ca/jrcraig/pdf/bluebird_developers_manual.pdf (2002).访问日期: 2018 年 6 月 11 日
17. Dennis Jr., J.E., Woods, D.J.: 微型计算机上的优化: Nelder-Mead 单纯形算法。In: Wouk, A. (ed.) New Computing Environments: 大型计算中的微型计算机, 第 116-122 页。工业与应用数学学会, 宾夕法尼亚州费城 (1987 年)
18. Fletcher, R., Leyffer, S.: 无惩罚函数的非线性编程。Math.Program.Ser.**91**, 239-269 (2002)
19. Fletcher, R., Leyffer, S., Toint, PHL: 论过滤-SQP 算法的全局收敛性。SIAM J. Optim.**13**(1), 44-59 (2002)
20. Gao, F., Han, L.: 利用自适应参数实现 Nelder-Mead 单纯形算法。Comput.Optim.**51**(1), 259-277 (2012)
21. Gould, N.I.M., Orban, D., Toint, PhL: CUTer (and SifDec): a constrained and unconstrained testing environment, revisited.ACM Trans.Math.Softw.**29**(4), 373-394 (2003)
22. Hedar, A.-R.: Global optimization test problems. http://www-optima.amp.i.kyoto-u.ac.jp/member/student/hedar/Hedar_files/TestGO.htm (2017).访问日期: 2018 年 6 月 11 日
23. Hock, W., Schittkowski, K.: Test Examples for Nonlinear Programming Codes, Lecture Notes in Economics and Mathematical Systems, vol. 187.Springer, Berlin (1981)
24. Jahn, J.: Introduction to the Theory of Nonlinear Optimization.柏林施普林格出版社 (1994 年)
25. Jamil, M., Yang, X.-S.: A literature survey of benchmark functions for global optimisation problems.Int.J. Math.Modell.Numer.Optim.**4**(2), 150-194 (2013)
26. Kelley, C.T.: 利用充分下降条件检测和补救奈德-梅德算法中的停滞。SIAM J. Optim.**10**(1), 43-55 (1999)
27. Kitayama, S., Arakawa, M., Yamazaki, K.: Sequential approximate optimization using radial basis function network for engineering optimization.Optim.Optim.**12**(4), 535-557 (2011)
28. Lagarias, J.C., Reeds, J.A., Wright, M.H., Wright, P.E.: Nelder-Mead simplex method in low dimensions.SIAM J. Optim.**9**, 112-147 (1998)
29. Le Digabel, S.: 算法 909: NOMAD: 使用 MADS 算法的非线性优化。ACM Trans.Math.Softw.**37**(4), 44:1-44:15 (2011)
30. Lukšan, L., Vlček, J.: 非光滑无约束和线性约束优化的测试问题。技术报告 V-798, ICS AS CR (2000)
31. Matott, L.S., Leung, K., Sim, J.: 将 MATLAB 和 Python 优化器应用于涉及地下水流和污染物迁移建模的两个案例研究。Comput.Geosci.**37**(11), 1894-1899 (2011)
32. McKinnon, K.I.M.: Nelder-Mead 单纯形法向非稳态点的收敛。SIAM J.Optim.**9**(1), 148-158 (1998)
33. Mezura-Montes, E., Coello, C.A.: 工程优化中有用的不可行解与演化算法。In: 第四届墨西哥人工智能进展国际会议论文集, MICAI'05, 第 652-662 页, 施普林格, 柏林 (2005 年)。
34. Mladenovic, N., Petrovic, J., Kovac'evic'-Vujc'ic', V., C'angalic', M.: Solving spread spectrum radar

- 通过塔布搜索和变量邻域搜索解决多相编码设计问题。Eur.J. Oper.Res.
151(2), 389-399 (2003)
35. Moré, J.J., Garbow, B.S., Hillstom, Kenneth E.: Testing unconstrained optimization software.ACM Trans.Math.Softw.**7**(1), 17-41 (1981)
 36. Moré, J.J., Wild, S.M.: Benchmarking derivative-free optimization algorithms.SIAM J. Optim.**20**(1), 172-191 (2009)
 37. Nazareth, L., Tseng, P.: 锦上添花：基于黄金分割搜索的 Nelder-Mead 算法变体。Comput.Optim.**22**, 133-144 (2002)

38. Nelder, J.A., Mead, R.: A simplex method for function minimization. *Comput.J.* **7**(4), 308-313 (1965)
39. Pigache, F., Messine, F., Nogarede, B.: 压电变压器的优化设计: 基于分析模型和确定性全局优化的合理方法。 *IEEE Trans.Ultrason.Ferroelectr.Freq.控制* **54** (7), 1293-1302 (2007 年)
40. Price, C.J., Coope, I.D., Byatt, D.: Nelder-Mead 算法的收敛变体。 *J. Optim.* **113**(1), 5-19 (2002)
41. Rodríguez, J.F., Renaud, J.E., Watson, L.T.: 用于顺序响应面近似和优化的可信区域增强拉格朗日方法。 *J. Mech.Des.* **120**(1), 58-66 (1998).03
42. Strasser, M.: Übertragung des Optimierungsverfahrens von Nelder und Mead auf restringierte Problem. 德国达姆施塔特工业大学数值数学小组毕业论文 (1994 年)
43. Tao, J., Wang, N.: 基于 DNA 双螺旋混合 GA 的汽油调配配方优化问题。 *Chem.Eng.Technol.* **31**(3), 440-451 (2008)
44. Tribes, C., Dubé, J.-F., Trépanier, J.-Y.: 多学科优化问题的分解: 简化机翼设计的公式和应用。 *Eng.Optim.* **37**(8), 775-796 (2005)
45. Tseng, P.: 强化后裔简约搜索法: 一种通用方法。 *SIAM J. Optim.* **10**(1), 269-288 (1999)
46. Wang, K., Wang, N.: 用于动态系统参数估计的新型 RNA 遗传算法。 *Chem.Eng.Res.Des.* **88**(11), 1485-1493 (2010)
47. Zhao, J., Wang, N.: 基于膜计算的生物启发算法及其在汽油混合调度中的应用。 *Comput.Chem.计算*. **35**(2), 272-283 (2011)