

约束优化模式搜索法研究进展

黄天云

(西南民族大学计算机科学与技术学院 成都 610041)

摘 要 实际工程应用中的优化问题通常包含复杂的约束条件,其目标函数可能是非线性、非连续、不可微甚至随机函数;而约束函数可能是线性、非线性、离散变量集,甚至黑盒函数(例如,由程序代码生成的值集合);约束变量也可能是包含连续、离散或分类值的混合变量.这些复杂的情况,使得没有任何导数/梯度信息可供利用,因此无法利用现有的凸优化技术求解.文中研究如何利用模式搜索法对常见的混合变量约束优化模型进行求解.首先对直接搜索法的发展历史进行概述;特别地,给出模式搜索法统一的数学描述和收敛性分析.对约束条件从无约束(一般模式搜索)到线性约束(广义模式搜索 GPS)、非线性约束(GPS 过滤法)和混合约束(广义混合变量规划 GMVP)的推广以及在边界约束条件下,算法搜索方向从有限集向紧致集的扩展(网格自适应直接搜索 MADS),进行了全面分析;在此基础上对该领域尚存在的问题及进一步的研究方向进行了总结.

关键词 约束优化;直接搜索法;模式搜索法;广义模式搜索 GPS;GPS 过滤法;广义混合变量规划 GMVP;网格自适应直接搜索 MADS

中图法分类号 TP301

Research Advances on Pattern Searches in Constrained Optimization

HUANG Tian-Yun

(School of Computer Science and Technology, Southwest University for Nationalities, Chengdu 610041)

Abstract Many complicated optimization models in engineering are mixed-variable constrained, containing high-dimensional continuous, discrete or even categorical variables. The objectives may be non-differentiable and contaminated by random noises, i. e., the problem is non-smooth. No information of the gradient is available or trustworthy. The constraints may be linear or non-linear, discrete sets, or even black-box functions generated from stochastic simulations or codes. Hence, direct searches that need no recourse to explicit derivatives are revived and become popular since the new century. A historical overview of the direct searches is given in this paper. Specially, the unified mathematical description and convergence analysis of pattern searches in recent years are discussed. Pattern searches in constrained optimization are thoroughly analyzed, from bound and linear constrained problems (Generalized Pattern Searches, GPS), to non-linear constrained (GPS Filter) and mixed variables constrained programming (GMVP). The new extension on the poll directions to a dense space in Mesh Adaptive Direct Search (MADS) is also identified. Some existing problems and future directions in this field are pointed out with thoughtful discussions.

Keywords constrained optimization; direct search; pattern search; generalized pattern search; GPS filter; generalized mixed variables programming; mesh adaptive direct search

1 引言

人们对约束优化问题已进行了多年研究,提出了诸如模拟退火、遗传和进化、神经网络等新算法。然而这些算法都只适合于特定的应用场景,对复杂的混合约束优化问题,尚无统一的方法进行求解,甚至很多问题只能通过随机搜索来寻求次优解。

注意到实际工程应用中的很多优化问题通常包含复杂的约束条件,其目标函数可能是非线性、非连续、不可微或随机函数(因环境噪声而导致的目标函数值的不确定性,即使在相同的约束条件下,每次得到的测量值也可能不同),因此需要进行多重优化;而约束条件既可以是线性或非线性函数,也可以是离散变量集,甚至是黑盒函数(例如,由程序代码生成的值集合);约束变量也可能是包含连续、离散或分类值的混合变量。这些复杂的情况,使得没有任何导数/梯度信息可供利用(因为随机噪声的影响,甚至连对梯度的估计也可能是不正确的),因而无法利用现有的凸优化技术进行求解。

因此,问题回到最初的求解思路:利用直接搜索法(direct searches)^[1],但不是单纯的随机搜索。诚然,直接搜索法是一种古老的算法。然而,近 10 年来对直接搜索法、特别是模式搜索法的再次深入研究和推广,使其焕发了新的生机。更重要的是,以前大多数人一直认为直接搜索法仅仅是一种启发式搜索算法,没有理论依据,因而问题求解存在不确定性(可能得到解,也可能得不到解;得到的解可能正确,也可能不正确)。然而通过近年的研究,人们已经完善了其理论依据和收敛性分析(GPS、GMVP 和 MADS 是 3 种对模式搜索法最重要的推广),因此这也为实际应用提供了理论基础。

目前,国内绝大多数研究都已经转向遗传和进化、神经网络、人工免疫等方向。然而,直接搜索法在很多大型工程问题中有着重要应用,仍需要我们进一步深入研究。

本文对直接搜索法、特别是模式搜索法在约束优化问题中的研究进展进行分析对比,目的是使广大研究者对直接搜索法进行透彻而深入的理解,并且推动其在理论研究和实际应用中的进一步发展。

本文第 2 节对直接搜索法从 1960 年代到目前的发展历史进行概述,主要包括单纯型法、模式搜索法和自适应直接搜索;第 3 节给出模式搜索法统一的数学描述和收敛性分析,对约束条件从无约束(一般模

式搜索)到线性约束(广义模式搜索 GPS)、非线性约束(GPS 过滤法)和混合约束(广义混合变量规划 GMVP)的推广以及在边界约束条件下,算法搜索方向从有限集向紧致集的扩展(网格自适应直接搜索 MADS)进行全面分析,最后对该领域尚存在的问题及进一步的研究方向进行总结。

2 发展历史概述

传统非线性优化分析,通常依赖于对 n 维空间目标函数 f 的 Taylor 展式逼近。如拟牛顿法,假设一阶和二阶导存在,用二阶 Taylor 展式对 f 局部二次逼近;最速下降法,假设一阶导存在,用一阶 Taylor 展式对 f 线性逼近。0 阶方法如直接搜索法,则不需要导数计算和对目标函数 f 的逼近。

定义 1. 所有导数无关法(derivative free)均不依赖于 Taylor 展式。然而,仅当连续迭代过程仅需根据目标函数值相对大小来选择新的迭代点,而不必保证充分下降条件(Armijo-Goldstein-Wolfe)^[2-4]时,才是直接搜索法。

相比而言,直接搜索法有以下优点:(1)启发式搜索,便于实施;(2)不需计算方向导数或对梯度的有限差分近似;(3)可以解决传统方法不能求解的大多数非线性规划问题;(4)可以作为传统方法如拟牛顿法等的热起点(hot start),以保证算法的局部收敛性。

直接搜索法^[1]有如下划分:(1)单纯型法(Simplex)及其扩展^[5];(2)模式搜索法(Pattern Search)如多向搜索 MDS^[6]、广义模式搜索 GPS^[7-8]、广义混合变量规划 GMVP^[9];(3)网格自适应直接搜索如 MADS^[10]等。

尽管直接搜索法在很多工程问题中应用很好,然而对算法能否保证迭代过程的全局收敛性一直存在疑问,后文将不加证明地引入一些结论。

以下按上述分类对直接搜索法的发展过程进行概述,主要研究集中在 1960~1970 年代的启发式算法,同时讨论 1990 年代到目前的理论分析和相关扩展算法(由此奠定实际应用的理论基础),对算法的收敛性分析也将简单介绍。

2.1 单纯型法

1957 年,Box 提出基于目标函数值比较的进化操作算法(evolutionary operation)^[11]。

1962 年,Spendly、Hext 和 Himsforth 提出单纯型搜索法^[5]。与早期的直接搜索法每次迭代需要

$2n \sim 2^n$ 个函数值估计不同,单纯型法仅需最多 $n+1$ 个函数值估计,就可确定下降(downhill)方向.显然, n 维空间的 $n+1$ 个点可确定一个平面,且仅需 f 在 $n+1$ 个点的值,就可以通过有限差分来估计 $\nabla f(\mathbf{x})$;由此也得到 $n+1$ 个点的单纯型.单纯型搜索的基本思想是:在 \mathfrak{R}^n 上构造非退化的单纯型,并在当前点所在的单纯型各顶点方向搜索.“非退化”保证与任意顶点相邻的边集构成 \mathfrak{R}^n 的线性基.

单纯型迭代中最重要的变换是反射(Reflection).参见图 1 中“①”,假设点 1 是最优点,算法将最坏点 2 沿其对应面中心点 4 反射,得反射点 5;若点 5 仍然是最坏点,选择次坏点 3,重复上述过程.显然,最坏点将沿着其余顶点的中心方向移动,以期获得目标函数值有所改善(相对于当前最优点严格下降)的新迭代点.

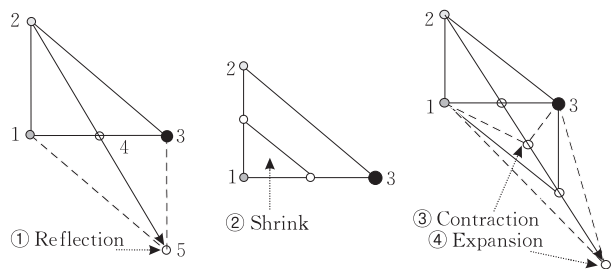


图 1 单纯型及其改进算法的主要变换示意图

上述迭代过程得到的反射序列,若最终返回起始点,但没有替换原始最优点 \mathbf{x}_k ,则 \mathbf{x}_k 是一个局部平稳点.此时,迭代易陷入无限循环.需要缩短最优点的相邻边长度,再次搜索;或借助于其它高阶方法以加快局部收敛.

改进单纯型法由 Nelder 和 Mead 于 1965 年提出(图 1②~④)^[12].新的变换如膨胀(Expansion)、收缩(Contraction 或 Shrink),可以加速迭代收敛,并且避免陷入局部最优点. Nelder-Mead 单纯型法是典型的直接搜索法,在实际问题中应用广泛.然而,该方法也可能失效(收敛到非平稳点)^[13];特别是针对高维空间问题,难以证明其全局收敛性^[14],因此无法当作一般化的方法使用.甚至此后 20 年,对该问题的收敛性分析也一直未能得到进展^[6,13-14].

在 Nelder-Mead 单纯型搜索基础上, Torczon 于 1989 年提出多向搜索 MDS^[6].在每次迭代完成后,通过调整尺度参数(Scale)以排除反射和膨胀所造成的单纯型形状的改变,并更新基矩阵 \mathbf{B}_{k+1} (从而更新搜索模式).MDS 能保证迭代的全局收敛.对 MDS 的收敛性已进行了详细的分析和研究^[15-16].

2.2 模式搜索法

1959 年,Davidon 提出坐标搜索^[17],也称轴松弛(axial relaxation)或局部扰变(local variation)等.算法从当前迭代中心点 \mathbf{x}_k 开始,沿坐标方向的 $2n$ 个试探点搜索.只要找到改善点($f(\mathbf{x}_{k+1}) < f(\mathbf{x}_k)$),则以 \mathbf{x}_{k+1} 为新中心点,重复迭代.最坏情况下, $2n$ 个试探点都已搜索完毕,尚未找到改善点;则缩短步长 Δ_k (step size),在当前中心点 \mathbf{x}_k 继续搜索.

1961 年,Hooke 和 Jeeves 引入直接搜索^[18],提出基于坐标搜索法变种的模式搜索,目的是通过成功迭代的步长模式(pattern)改进,加速算法收敛.

定义 2. 模式搜索法是对当前搜索点按固定模式和步长 Δ_k 探索移动(exploratory moves),以寻求可行下降方向(非最速下降方向)的直接搜索法.迭代过程只要找到相对于当前点的改善点,则步长递增,并从该点开始进入下一次迭代;否则步长递减,在当前点继续搜索.

参见图 2,若 \mathbf{x}_k 迭代成功,则下次迭代从临时点 $\mathbf{x}' = \mathbf{x}_k + (\mathbf{x}_k - \mathbf{x}_{k-1})$ 开始,其中 $(\mathbf{x}_k - \mathbf{x}_{k-1})$ 为模式步长;以期望着顺着方向 $(\mathbf{x}_k - \mathbf{x}_{k-1})$ 能获得比在 \mathbf{x}_k 更好的解.不管是否有 $f(\mathbf{x}') \leq f(\mathbf{x})$,坐标搜索都在 \mathbf{x}' 展开.若 \mathbf{x}' 坐标搜索成功(Success),置 $\mathbf{x}_{k+1} = \mathbf{x}'$,并从 \mathbf{x}_{k+1} 开始新的迭代;否则,坐标搜索在 \mathbf{x}_k 展开.若在 \mathbf{x}_k 不成功(Unsuccess1),则坐标搜索在 \mathbf{x}_{k-1} 展开,并且步长减半($\theta = 1/2$).若在 \mathbf{x}_{k-1} 仍然不成功(Unsuccess2),回溯并重复上述过程.

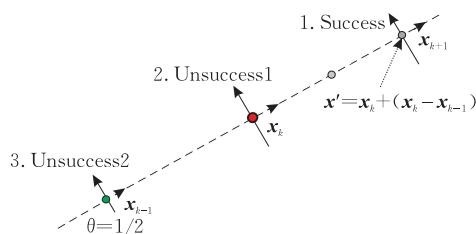


图 2 Hooke 和 Jeeves 步长模式法(\mathfrak{R}^2)

类似地,Rosenbrock 的自适应坐标搜索(1960 年)则分阶段进行,利用连续两阶段的迭代点下降方向(descend direction)来进行坐标旋转^[19].该算法阶段内采取坐标搜索,当在每个搜索方向上都至少出现一次成功和失败步时,转入下一阶段.此时,以连续两阶段的起始中心点所在方向为坐标方向(即坐标旋转),开始新的阶段搜索.坐标旋转可以通过简单的 Gram-Schmidt 正交化过程实现.显然,阶段间的方向选择代表搜索过程最可能的下降方向,因此更可能找到最终的收敛点.图 3 给出该

算法在 Rosenbrock 香蕉函数上的应用, 搜索路径首先从初始点 $(-1.2, 1)$ 垂直向上(取决于初始方向选择, 也可能水平进入), 沿着山谷先下降到谷底, 再逐步上升, 前进到最优点 $(1, 1)$ (测试算法在点 $(0.9999, 1.0000)$ 陷入局部平稳, 迭代终止; 最优值为 $8.660966\text{e}-007 \rightarrow 0$). 谷底附近的两个坐标清晰地标记了坐标旋转的过程.

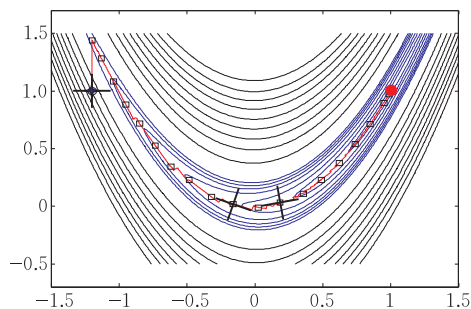


图3 自适应坐标旋转(Rosenbrock 香蕉函数)

1964 年 Davies、Swann 和 Campey^[20-21] 改进了自适应坐标搜索, 在每个方向采用一维线性搜索; 当找到一个一维极小点时, 在该点进行二次插值以预测可能下降方向, 寻找更接近的极小点位置. Powell 方法^[22] 则是导数无关的非线性共轭梯度法, 采用基于模型的凸二次插值以逼近目标函数. 因此严格来说, Powell 方法仅是导数无关法, 而非直接搜索法.

模式搜索的收敛分析早期由 Polak、Céa 和 Berman 进行了研究^[23-25]. Polak 的研究^[23] 表明: 模式搜索法在步长递减之前, 仅可能产生迭代序列 $\{x_k\}$ 的有限个中间点. 因此, 迭代过程不会阻塞于任何一个中间点, 序列 $\{x_k\}$ 最终必然收敛到全局平稳点 x' ; $f \in C^1$ 时有 $\nabla f(x') = 0$. Céa^[24] 则给出 Hooke 和 Jevees 模式搜索法^[18] 的全局收敛性证明. Berman^[25] 讨论了迭代序列 $\{x_k\}$ 的有理网格(rational lattice)结构:

$$L_k = L(x_k, \Delta_k) = \{x \mid x = x_k + \Delta_k \lambda, \lambda \in \Lambda\}.$$

此后, 对模式搜索的研究一直处于停滞状态. 直到 1990 年代, 对模式搜索的收敛性分析才引起广泛关注, 并由此建立了完整的理论依据^[6-9, 16].

所有模式搜索的收敛性分析均基于以下假设: (1) 当前迭代点的模式选择必须包含足够多的搜索方向, 以保证若 x_k 不是 f 的平稳点, 则必然至少存在一个可行下降方向. 就 Polak 和 Céa 而言, $x_{k+1} = x_k + \Delta_k e_i$, $i \in \{1, 2, \dots, n\}$, 其中 e_i 是单位坐标向量. 就 Berman 而言, 则要求 $\Lambda \in \mathfrak{R}^n$ 是有理网格整数点,

因此有理网格的基(basis)是单位矩阵 $I \in \mathfrak{R}^{n \times n}$; (2) 推广开来, 允许基为任意非奇异矩阵 $B \in \mathfrak{R}^{n \times n}$, $x_{k+1} = x_k + \Delta_k B \gamma'_k$ ($\gamma'_k \in Z$)^[7], 以扩展搜索方向和搜索步长; 此时 Polak 和 Céa 的分析只是其特殊情况; (3) 若能够从当前点 x_k 找到下降点 x_{k+1} , 则不允许步长 Δ_k 递减, 以避免过早收敛到非平稳点; (4) 搜索步长 $\Delta_{k+1} = \tau^w \Delta_k$ ($\tau > 1$), 其中 w 从事先设定的有限集取得. 因此, $w < 0$ 时, Δ_k 递减, $L_k \subset L_{k+1}$; $w = 0$ 时, Δ_k 保持不变, $L_k = L_{k+1}$; $w > 0$ 时, Δ_k 递增, $L_{k+1} \subset L_k$.

在对单纯型法进行改进并提出多向搜索 MDS^[6] 的基础上, Torczon 于 1991 年建立了 MDS 的收敛性分析^[16] (MDS 也是一种模式搜索法). 同时, 对模式搜索法的完整定义及全局收敛分析也于 1997 年给出^[7]. 考虑到工程应用中分类变量的存在, 传统对离散变量的连续松弛条件无法应用, Audet(2000 年)在 Torczon 基础上, 提出广义混合变量模式搜索算法 GMVP^[9], 以在必要的局部优化条件下寻求最优解. Audet(2003 年)对 Torczon 的模式搜索进一步推广, 提出广义模式搜索 GPS^[8], 以解决无约束和线性约束优化问题^[26-27]. 在强化的搜索方向和目标函数局部平滑假设下, 给出了简化的收敛分析.

2.3 自适应直接搜索

尽管模式搜索如 GPS 或 GMVP, 可以在每个迭代点通过足够多的方向进行搜索, 解决很多非线性优化问题, 然而针对混合变量和多约束条件优化, 在临近约束边界时往往无法保证一定能找到合适的下降方向. 网格自适应直接搜索 MADS^[10] 对 GPS 类算法进行扩展, 以允许局部探索(local exploration)或方向筛选(poll). MADS 是介于 GPS 和帧(frame)方法^[28-29] 之间的一类直接搜索算法.

GPS 类算法仅允许有限搜索方向(取决于基矩阵的选择). MADS 的局部探索则允许优化变量在限制约束空间 Ω^n 的一个紧致集(dense set)进行方向选择, 因此避免了 GPS 类算法陷入边界局部极小的问题. 这也是 MADS 和 GPS 的最大不同之处. MADS 解决了迭代搜索过程的两大基本问题: (1) 步长筛选(poll size); (2) 方向筛选(poll direction). 因此 MADS 更可能得到全局最优解. 图 4 给出直接搜索法的主要发展历史过程示意图(其它与传统优化技术相结合的方法如参数化 Lagrange 法、罚函数法(penalization)等没有列出, 仅就直接搜索和模式搜索进行概括).

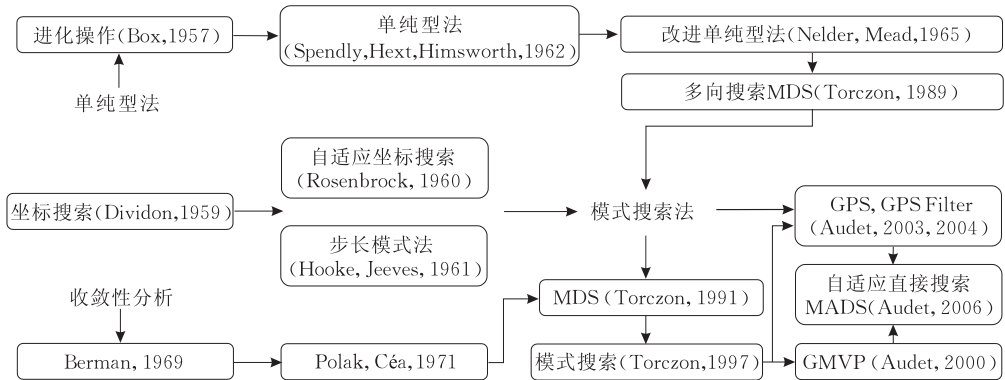


图 4 直接搜索法及其主要发展历史过程示意图

3 算法分析

对直接搜索法发展过程的概述,使得我们可以主要集中到对模式搜索法的研究. 鉴于模式搜索法是与方向导数或下降梯度无关的一类最优搜索算法,搜索模式和搜索步长的选择对迭代过程能否达到收敛起着决定性作用,因此首先引入基本定义,在此基础上对模式搜索法(包括 GPS、GMVP、MADS)进行统一描述并给出收敛性分析的主要结论. 对约束条件从无约束到线性约束、非线性约束以及混合约束的推广;以及边界约束条件下算法搜索方向从有限集向紧致集的扩展,也将详细分析并通过相关实例加以说明.

以下分析均在 \Re^n 上进行. 为简化起见,向量 α 和矩阵 A 的转置分别记为 α', A' . 若未特别声明,所有矩阵均按列向量表示.

3.1 基本定义

定义 3. 给定线性独立集 a_1, a_2, \dots, a_r ,若对 $\forall a \in \Re^n, \exists c_i \geq 0, i=1, 2, \dots, r, \ni a = c_1 a_1 + c_2 a_2 + \dots + c_r a_r$,则称 $\{a_1, a_2, \dots, a_r\}$ 是 \Re^n 的正支撑集(positive spanning set).

定义 4. \Re^n 的正支撑集 $\{a_1, a_2, \dots, a_r\}$ 必正线性独立(positive linear independent)^[30-31]. 简称 \Re^n 的正基(positive basis).

定理 1. 若 $\{a_1, a_2, \dots, a_r\}$ 是 \Re^n 的正支撑集,则 $\{a_2, \dots, a_r\}$ 是 \Re^n 的线性支撑集.

定理 2. 若 $\{a_1, a_2, \dots, a_r\}, a_i \neq 0$ 是 \Re^n 的线性支撑集,则如下命题等价^[30]:

- (1) $\{a_1, a_2, \dots, a_r\}$ 是 \Re^n 的正支撑集;
- (2) $\forall b \neq 0, \exists i, \ni b' a_i > 0$;
- (3) $\forall i = 1, 2, \dots, r, -a_i$ 在 $\{a_1, a_2, \dots, a_{i-1}, a_{i+1}, \dots, a_r\}$ 所构成的凸锥(convex cone)内.

\Re^n 的正基至少包含 $n+1$ 个元素(最小正基),至多包含 $2n$ 个元素(最大正基). 最大正基通常由 \Re^n 的 n 个线性基及其负值构成.

命题 1. 若 $\{a_1, a_2, \dots, a_r\}$ 是 \Re^n 的正基,则对任意非奇异矩阵 $B \in \Re^{n \times n}, \{Ba_1, Ba_2, \dots, Ba_r\}$ 亦是 \Re^n 的正基^[32].

命题 2. 若 $B \in \Re^{n \times n}$ 非奇异,则矩阵 $[B \quad -B]$ 构成 \Re^n 最大正基^[32].

3.2 统一描述和收敛性分析

所有模式搜索法均包括 3 个部分:(1) 模式定义;(2) 探索算法;(3) 步长更新算法. 因此以下按此对模式搜索法进行统一描述^①.

模式定义:模式由基矩阵 B 和生成矩阵 C_k 唯一确定. $B \in \Re^{n \times n}$ 非奇异,在迭代过程中通常保持不变; $C_k \in Z^{n \times p} (p > 2n)$ 按如下方式排列:

$$C_k = [M_k \quad -M_k \quad L_k] = [\Gamma_k \quad L_k],$$

其中 $M_k \in M \subset Z^{n \times n}$ (M 是有限的非奇异矩阵集合). $L_k \in Z^{n \times (p-2n)}$ 至少包含一列全 0(这使得搜索失败时可以快速回到起始点). 定义模式:

$$P_k = BC_k = [BM_k \quad -BM_k \quad BL_k] = [B\Gamma_k \quad BL_k].$$

显然 P_k 构成 \Re^n 的正基. 由此,在任意迭代步 k ,定义以 x_k 为中心的试探点(trial points): $x_k^i = x_k + s_k^i = x_k + \Delta_k Bc_k^i$. 其中,搜索步长 $\Delta_k \in \Re_+$;搜索方向 $Bc_k^i (C_k = [c_k^1 \dots c_k^p])$.

命题 3. 若 $B = \frac{1}{n} [b_1 \dots b_n]$ 非奇异,记

$$b_{n+1} = -\frac{1}{n} \sum_{i=1}^n b_i, \quad e = (1, 1, \dots, 1)',$$

① Audet 采用方向矩阵 $D_k \subset D$ 对 GPS 算法进行描述^[8]. 生成矩阵 G 在迭代过程中保持不变,仅从 D 中选择有限的方向集 D_k 来构造 \Re^n 的正基. 通过在网格 M_k 可选的搜索步和局部筛选(local poll),以寻求网格改善点(improved mesh points)或局部最优点(mesh local optimizers),并以此来调整下次迭代的步长参数和方向集.

令 $\Gamma_k = [nI \quad -e]$, 则 Γ_k 构成 \mathcal{R}^n 最小正基, 且 $B\Gamma_k = [b_1 \cdots b_n \quad b_{n+1}]$ 构成 \mathcal{R}^n 的最小搜索模式^[31].

命题 4. 若 $B = [b_1 \cdots b_n]$ 非奇异, 记

$$b_{n+1} = -\sum_{i=1}^n b_i, \quad e = (1, 1, \dots, 1)',$$

令 $\Gamma_k = [I \quad -e]$, 则 Γ_k 构成 \mathcal{R}^n 最小正基, 且 $B\Gamma_k = [b_1 \cdots b_n \quad b_{n+1}]$ 构成 \mathcal{R}^n 的最小搜索模式^[31].

对探索移动有如下两个基本假设:

(1) $s_k \in \Delta_k P_k \equiv \Delta_k BC \equiv \Delta_k [B\Gamma_k \quad BL_k]$;

(2) 若 $\min\{f(x_k + y), \forall y \in \Delta_k B\Gamma_k\} < f(x_k)$, 则有 $f(x_k + s_k) < f(x_k)$.

探索算法.

1. 给定初始点 $x_0 \in \mathcal{R}^n$, 初始步长 $\Delta_0 > 0$;

2. $k = 0, 1, \dots$;

2.1. 计算 $f(x_k)$, 按一定模式确定探索步 s_k ;

2.2. 计算 $\rho_k = f(x_k) - f(x_k + s_k)$. 若 $\rho_k > 0$, 置 $x_{k+1} = x_k + s_k$; 否则 $x_{k+1} = x_k$;

2.3. 更新生成矩阵 C_k 和步长参数 Δ_k .

步长更新算法.

给定 $\tau \in Q, \tau > 1; \{\omega_0, \omega_1, \dots, \omega_L\}: \omega_0 < 0, \theta = \tau^{\omega_0}, \omega_i \geq 0, i = 1, 2, \dots, L$;

$\lambda_k \in \Lambda = \{\tau^{\omega_1}, \tau^{\omega_2}, \dots, \tau^{\omega_L}\}, L = |\Lambda| < +\infty$; 按如下规则更新参数 Δ_k :

(1) $\rho_k \leq 0, \Delta_{k+1} = \theta \Delta_k$ (失败);

(2) $\rho_k > 0, \Delta_{k+1} = \lambda_k \Delta_k$ (成功).

以上算法保证得到的迭代点序列 $\{x_k\}$, 只能出现在拉伸平移后的整数网格点 (文献[7]定理 3.2), 因此推广了 Berman 的研究^[25].

并且若定义 $L(y) = \{x: f(x) < f(y)\}$, 有以下定理.

定理 3. 若 $L(x_0)$ 紧致(compact), 则

$$\liminf_{k \rightarrow +\infty} \Delta_k = 0 \text{ (文献[7]定理 3.3)}.$$

反过来, 若 $\liminf_{k \rightarrow +\infty} \|\nabla f(x_k)\| \neq 0$, 即梯度非 0, 则必然可以继续搜索, 并且在有限步 (文献[7]引理 6.1) 后找到下降方向 (文献[7]命题 3.4).

强化前面假设(2): $\min\{f(x_k + y), y \in \Delta_k B\Gamma_k\} < f(x_k) \Rightarrow f(x_k + s_k) \leq \min\{f(x_k + y), \forall y \in \Delta_k B\Gamma_k\}$; 即 $f(x_k + s_k)$ 是所有搜索方向最小值, 有以下定理.

定理 4. 若 $L(x_0)$ 紧致, f 在 $L(x_0)$ 的邻域连续可微, 生成矩阵 C_k 列范数有界, 则 $\lim_{k \rightarrow +\infty} \Delta_k = 0$; 并且有 $\lim_{k \rightarrow +\infty} \|\nabla f(x_k)\| = 0$ (文献[7]定理 3.7).

由此, 当步长控制参数 Δ_k 足够小时, 可以认为上述算法已经收敛 (此时所有方向上的梯度均为 0, 无法再找到可行的下降方向), 算法终止.

3.3 推广到边界和线性约束条件

从历史来看, 模式搜索向约束优化如边界、线性和非线性问题的推广是非常困难的. 类似在非约束模式搜索中, 对每个搜索方向, 足够长 (但不是非常长, 否则可能导致过早收敛到非成熟点) 的步长必须被保证. 然而应用到约束优化时会产生如下问题: 如何保证这些搜索方向仍然在可行域内?

为此, Lewis 对 GPS 算法在边界和线性约束条件下的推广进行了研究^[26-27], Audet 则给出其统一框架^[8]. 现考虑一般线性约束条件下的优化模型:

$$\min_{x \in \Omega} f(x), \quad f: \mathcal{R}^n \rightarrow \mathcal{R} \cup \{\infty\},$$

其中, 可行区域

$$\Omega = \{x \in \mathcal{R}^n: \ell \leq Ax \leq u\},$$

$$A = [a_1 \quad a_2 \quad \cdots \quad a_m]' \in Q^{m \times n},$$

$$\ell, u \in \{\mathcal{R} \cup \{\pm\infty\}\}^m, \ell \leq u.$$

当 $A = I$ 时, 模型为边界约束问题.

如前所述, 对非约束模式搜索, 只要保证沿着与最速下降在 90° 范围内的方向进行搜索即可 (这也是模式搜索采用正支撑集的原因). 然而, 约束条件可能会阻止某些虽然可以减少目标函数值, 但是超出约束边界的搜索步长, 特别是当采用可行迭代法(feasible iterates)时, 超出边界的不可行迭代是不允许的.

为此, 约束问题的迭代搜索过程要求: (1) $x_k \in \Omega$; (2) $x_k + s_k \in \Omega$; (3) $\forall y \in \Delta_k B\Gamma_k, x_k + y \in \Omega$. 其中 $\Gamma_k \in \Gamma$ 为核模式(core pattern) (Γ 是有限的矩阵集合). 同时, 在可行区域 Ω 边界要考虑模式的可行方向: 必须保证有足够多的可行方向供搜索, 否则算法将陷入边界局部极小. 此时, 模式 P_k 的选择必须反映可行区域 Ω 的边界几何约束. 为简化起见, 忽略基矩阵, 令 $B = I$, 则模式 $P_k = [\Gamma_k \quad L_k]$.

定义 5. 若 x 是满足上述边界和线性约束条件的可行点, 则 $\forall z \in \mathcal{R}^n, (\nabla f(x), z - x) \geq 0$. 即 x 是一个 KKT 点(Karush-Kuhn-Tucker).

以下首先引入凸分析的一些基本概念^[33].

定义 6. 凸区域: 若 \bar{x} 是 Ω 的极小点, 则 X 是凸区域 $\Leftrightarrow \forall x \in \Omega, \nabla f(x)'(x - \bar{x}) \geq 0$.

定义 7. 锥(cone) C 是多面体, 若 $C = \{x | a_j'x \leq 0, j = 1, 2, \dots, r\}, x, a_1, a_2, \dots, a_r \in \mathcal{R}^n$.

定义 8. 锥 C 有限生成(finitely generated), 若

$$C = \{x | x = \sum_{j=1}^r u_j a_j, u_j \geq 0, j = 1, 2, \dots, r\} \\ = \text{Cone}(\{a_1, a_2, \dots, a_r\}),$$

$a_j (j=1,2,\dots,r)$ 为锥 C 的生成因子 (generators).

定义 9. 若 $y=0$, 或 $\exists \{x_k\} \subset \Omega, \exists x_k \rightarrow x; \frac{(x_k-x)}{\|x_k-x\|} \rightarrow \frac{y}{\|y\|}$, 则向量 y 和 Ω 在点 x 正切. 称点 x 所有切线的集合为切锥 (tangent cone) (图 5), 记为

$$T_{\Omega}(x) = cl\{\mu(w-x) : \mu \geq 0, w \in \Omega\}.$$

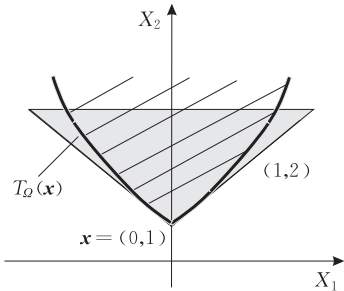


图 5 凸区域 Ω 的切锥 $T_{\Omega}(x)$

定义 10. $N_{\Omega}(x) = \{v \in \mathbb{R}^n : \forall w \in T_{\Omega}(x), v'w \leq 0\}$ 为 Ω 在点 x 的法锥 (normal cone) 或 $T_{\Omega}(x)$ 的极锥 (polar cone).

现针对约束矩阵 A , 定义可行区域 Ω 的边界几何特征.

定义 11. 定义可行区域 Ω 的左右边界点所在的半空间 (Half-Space):

$$A_{\ell_i} = \{x | a'_i x = \ell_i\}, A_{u_i} = \{x | a'_i x = u_i\}.$$

定义 12. 定义可行区域 Ω 的左右边界所在的 ϵ 半径区域:

$$\partial\Omega_{\ell_i} = \{x \in \Omega | dist(x, A_{\ell_i}) \leq \epsilon\};$$

$$\partial\Omega_{u_i} = \{x \in \Omega | dist(x, A_{u_i}) \leq \epsilon\}.$$

定义 13. 定义可行区域 Ω 的左右边界索引集: $I_{\ell}(x, \epsilon) = \{i | x \in \partial\Omega_{\ell_i}\}, I_u(x, \epsilon) = \{i | x \in \partial\Omega_{u_i}\}.$

定义 14. 定义可行区域 Ω 左右边界所在面的外法向量:

$$v_{\ell_i}(x, \epsilon) = -a_i, i \in I_{\ell}(x, \epsilon);$$

$$v_{u_i}(x, \epsilon) = a_i, i \in I_u(x, \epsilon).$$

定义 15. 定义 $K(x, \epsilon)$ 是由 $v_{\ell_i}(x, \epsilon)$ 和 $v_{u_i}(x, \epsilon)$ 有限生成的凸锥, 则 $K(x, 0) = N_{\Omega}(x)$; 记 $K(x, \epsilon)$ 的极锥为 $K^*(x, \epsilon)$, 则 $K^*(x, 0) = T_{\Omega}(x)$ (图 6).

锥 $K(x_k, \epsilon)$ 由点 x_k 与边界距离为 ϵ 的各面法向量生成. 它的极锥 $K^*(x_k, \epsilon)$ 非常重要. 因为当 $\epsilon > 0$ 足够小时, 我们仍然可以从 x_k 沿着 $K^*(x_k, \epsilon)$ 的所有方向至少搜索前进步长 $\delta \geq \epsilon$, 且仅依赖于参数 ϵ , 但仍能保证在可行域内 (这对点 x_k 的切锥不成立, 因为切锥不能反映某个点在靠近边界时, 对边界形状的逼近). 此时, 需要选择新的核模式 Γ_k , 使得其所

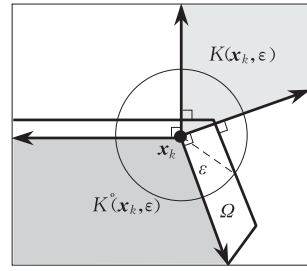


图 6 约束边界的几何特征

有生成因子从 $K^*(x_k, \epsilon), 0 \leq \epsilon \leq \epsilon^*$ 取得 ($\epsilon^* > 0$ 为事先设定的值). 注意当 ϵ 从 0 变化到 ϵ^* 时, 仅存在有限个完全不同的锥 $K(x_k, \epsilon)$, 因为区域 Ω 在该点仅包含有限个面.

当 x_k 远离边界时, $K(x_k, \epsilon) = \phi, K^*(x_k, \epsilon) = \mathbb{R}^n$, 此时 \mathbb{R}^n 的任意正基都可以用来生成 $K^*(x_k, \epsilon)$; 当 x_k 接近边界时, $K(x_k, \epsilon) \neq \phi$, 搜索模式的选择必须与边界几何特征一致. 对此, 要求核模式 Γ_k 包含所有极锥 $K^*(x_k, \epsilon) (0 \leq \epsilon \leq \epsilon^*)$ 的生成因子, 使得即使 x_k 趋近边界 ($\epsilon \rightarrow 0$), 在 $K^*(x, \epsilon)$ 内仍然存在可行的搜索方向和足够大的步长 (文献 [27] 命题 6.5) ①.

以下步骤给出如何在线性约束条件下构造核模式 Γ_k (文献 [27] 命题 8.2).

对任意迭代步 k , 设定 ϵ 上界 ϵ^* , 下界 ϵ_* ②. 选择 $\epsilon_k > \epsilon_*$:

1. 定义凸锥 $K(x_k, \epsilon_k)$;
2. 以 $K(x_k, \epsilon_k)$ 的生成因子 $v_{\ell_i}(x_k, \epsilon_k), v_{u_i}(x_k, \epsilon_k)$ 构造列矩阵 V . 若 V 列满秩, 转步 3; 否则, 递减 ϵ_k 直至 $|I_{\ell}(x_k, \epsilon_k)| + |I_u(x_k, \epsilon_k)|$ 减小, 转步 1;
3. 由矩阵 $\pm(I - V(V'V)^{-1}V')$ 列构造有理正基 N ;
4. $\Gamma_k = [N \quad V(V'V)^{-1} \quad -V(V'V)^{-1}]$.

定理 5. 若迭代子序列 $\{x_k\}_{k \in K} \rightarrow \hat{x}, f$ 在 \hat{x} 严格可微; 且模式的选择符合上述边界几何约束, 则对 $\forall w \in T_{\Omega}(\hat{x}), \nabla f(\hat{x})w \geq 0; -\nabla f(\hat{x}) \in N_{\Omega}(\hat{x})$. 因此, \hat{x} 是一个 KKT 点 (文献 [8] 定理 3.14).

3.4 向非线性约束条件的推广

考虑如下形式的非线性约束优化问题:

$$\min_{x \in X} f(x), \quad \text{s. t. } C(x) \leq 0,$$

其中, $f: X \rightarrow \mathbb{R} \cup \{\infty\}, C: X \rightarrow (\mathbb{R} \cup \{\infty\})^m, C = (c_1, c_2, \dots, c_m)'$. 定义域 X 给出 \mathbb{R}^n 上的边界或线性

① 文献 [34] 的第 453~463 页详细分析了线性约束条件为什么必须考虑边界几何约束以及当 ϵ 足够小时, 如何寻求符合边界约束的可行搜索方向和搜索步长.

② 过长或过短的步长都可能导致提前收敛到非成熟点或病态收敛. 基于梯度的策略通过充分下降以避免过长步长, 通过函数的凸性以避免过短步长 (Armijo-Goldstein-Wolfe 条件). 模式搜索法, 不管采用充分下降还是简单下降 (有理网格 [7]、移动网格 [29]), 均提供了内置机制以避免这些情况出现.

约束多面体;而 $\mathbf{C}(\mathbf{x}) \leq \mathbf{0}$ 给出相应的非线性约束区域 Ω , 因此可行区域为 $X \cap \Omega$. 在约束函数以黑盒函数形式给出时(如随机系统仿真或程序代码调用, 无法确定是线性还是非线性约束), 通常将所有约束看成是非线性的.

通常, 如果非线性约束条件的梯度可以较容易和较准确地计算, 或者目标函数和约束在接近边界时可以通过线性逼近的话, 则约束导数在可行方向法中就可以得到应用, 如文献[35-36]的工作. 这些方法引入了对约束导数的显式使用, 通过约束导数来计算当迭代点逼近约束边界时的可行方向.

其它进展包括参数化 Lagrange 法^[37-38], 罚函数法和屏障法(Barrier). 参数化 Lagrange 法通过引入松弛变量来将非线性约束转换为等式约束, 并由此估计 Lagrange 乘子. 不确定(inexact) 惩罚对目标函数进行修改, 以使得靠近约束边界的点和所有不可行点取非常大的目标函数值;而确定(exact)的惩罚通过对每个约束条件引入足够大的加权, 将不等式约束看成确定的、非光滑惩罚函数. 该思想也在屏障法中得到应用, 此时对可行区域外的所有点, 都将其目标函数值置为无穷.

一个处理非线性约束的典型思想来源于 Fletcher 和 Leyffer^[39]——文献[40]的 GPS 过滤法(Filter). 不同于上述提及的方法, GPS 过滤法不是将约束条件和目标函数综合起来进行极小化, 而是通过引入约束违背函数 h , 将模型转化为二目标问题(bi-objective)进行求解. GPS 过滤法试图同时减少目标函数值和约束违背函数值. 然而优先考虑 h , 因为在这些极小不可行点的筛选, 不仅可能跳出局部极小点, 也允许搜索其相邻区域, 从而更可能获得全局最优解.

定义 16. 定义具备如下属性的函数 $h(\mathbf{x}) \geq 0$ 为约束违背函数:

- (1) $h(\mathbf{x}) = 0$, 当且仅当 $\mathbf{C}(\mathbf{x}) \leq \mathbf{0}$;
- (2) $h(\mathbf{x}) > 0$, 当且仅当 $\mathbf{C}(\mathbf{x}) \not\leq \mathbf{0}$;
- (3) $h(\mathbf{x}) = \infty$, 若存在 $\mathbf{C}(\mathbf{x})$ 的约束 $c_i(\mathbf{x}) \rightarrow \infty$.

例如, 通常设 $h(\mathbf{x}) = \|\mathbf{C}(\mathbf{x})_+\|_2^2$, 其中 $\|\cdot\|$ 为向量范数. 当 $c_i(\mathbf{x}) \leq 0$ 时, $(\mathbf{C}(\mathbf{x})_+)_i = 0$; 否则 $(\mathbf{C}(\mathbf{x})_+)_i = c_i(\mathbf{x})$.

类似 GPS Barrier 法, 可以将多面体 X 区域外的 h 值置 $+\infty$, 并设 ψ_X 为当前迭代点是否在可行区域的指示函数(在可行区域时 $\psi_X = 0$; 否则 $\psi_X = +\infty$). 因此得到新的约束违背函数 $h_X = h + \psi_X$. 注意到可行域的迭代收敛结果仅依赖于 h , 而非 h_X .

定义 17. 定义向量 \mathbf{w} 支配(dominates)向量 \mathbf{w}' , 若对 $\forall i, w_i \leq w'_i$, 且 $\mathbf{w} \neq \mathbf{w}'$, 记为 $\mathbf{w} < \mathbf{w}'$; 同样有 $\mathbf{w} \leq \mathbf{w}'$, 若 $\mathbf{w} < \mathbf{w}'$ 或 $\mathbf{w} = \mathbf{w}'$.

定义 18. 由上述问题描述及定义 17, 对 $\forall \mathbf{x}_k, \mathbf{x} \in \mathcal{R}^n$, 定义支配关系: $\mathbf{x}_k < \mathbf{x} \Leftrightarrow ((h_X(\mathbf{x}_k), f(\mathbf{x}_k)))' < ((h_X(\mathbf{x}), f(\mathbf{x})))'$; 同样可推广到 $\mathbf{x}_k \leq \mathbf{x}$.

令 \mathcal{F} 为 \mathcal{R}^n 上所有不可行点的过滤集, 并且该过滤集中任意两个点都不存在支配关系. 令 h_{\max} 为最大允许的约束违背上限, f^x 为迭代到目前为止寻找到的最佳可行目标函数值. 满足如下条件的点 \mathbf{x}' 将被集 \mathcal{F} 过滤: (1) 点 \mathbf{x}' 被集 \mathcal{F} 的某个点 \mathbf{x} 支配; (2) 点 \mathbf{x}' 的约束违背值超过上限 h_{\max} ; (3) 点 \mathbf{x}' 虽可行但非最佳可行. 由此, 有如下定义的被过滤点集 $\bar{\mathcal{F}}$.

定义 19. $\bar{\mathcal{F}} = \bigcup_{\mathbf{x} \in \mathcal{F}} \{\mathbf{x}': \mathbf{x}' \geq \mathbf{x}\} \cup \{\mathbf{x}': h_X(\mathbf{x}') \geq h_{\max}\} \cup \{\mathbf{x}': h_X(\mathbf{x}') = 0, f(\mathbf{x}') \geq f^x\}$.

GPS 过滤法试图同时降低目标函数 f 和约束违背函数 h 的值, 即寻求最佳可行点(best feasible points)和最佳不可行点(least infeasible points), 因此迭代点的选择同时包括可行覆盖解和最佳不可行覆盖解(有最小的 h 值). 但显然应该优先降低 h 的值, 因为在这些不可行点的筛选不仅可能跳出局部极小, 也允许搜索其它区域以寻求全局最优解^①.

定义 20. 令 f_k^x 为直到第 k 次迭代的可行覆盖值(incumbent), 即到目前为止的最佳可行目标函数值. 当没有寻求到任何可行点时, 令 $f_k^x = +\infty$. 令 $h_k^l > 0$ 为直到第 k 次迭代的最小约束违背值, 其对应的目标函数值为 f_k^l ; 并且若该点不存在或 $h_k^l > h_{\max}$ 时, 令 $h_k^l = h_{\max}$; $f_k^l = +\infty$. 令 S_k 为第 k 次迭代需要估计的所有点的覆盖集.

与无约束或线性约束优化不同, 因为要同时考虑两种情况的点集筛选, 在 GPS 过滤法中将网格改善点称为网格未过滤点(unfiltered mesh points), 将网格局部极小点称为网格孤立过滤点(mesh isolated filter points). 注意初始覆盖集 S_0 至少应包含一个可行解, 并落在其中某个点 \mathbf{x} 所在的 Δ_0 网格上; 而 k 次迭代的覆盖网格则是覆盖集所有点所在网格的并集, 这也使得每次迭代的搜索步有更多的点可供选择, 即

$$M(\mathbf{S}_k, \Delta_k) = \bigcup_{\mathbf{x} \in S_k} M(\mathbf{x}, \Delta_k).$$

① 对包含黑盒函数的优化问题, 无法保证全局收敛性; 即使得到的是全局最优解, 也难以证明. 模式搜索及其扩展算法的目的是在有限的计算资源或迭代次数下寻求尽可能好的解.

同样,筛选中心集 P_k 的选择包含最佳可行和最佳不可行点以及和几何约束 X 相一致的正支撑矩阵伸展得到的所有方向集,即当前筛选中心 \mathbf{p}_k 满足 $(h_X(\mathbf{p}_k), f(\mathbf{p}_k)) = (0, f_k^f)$ 或 $(h_X(\mathbf{p}_k), f(\mathbf{p}_k)) = (h_k^l, f_k^l)$, 且有 $P_k = \{\mathbf{p}_k\} \cup \{\mathbf{p}_k + \Delta_k d : d \in D_k\}$.

图 7 给出上述概念的示意图,可行域为 $X \cap \Omega = \{\mathbf{x} \in \mathbb{R}^n : h_X = 0\}$, 当前迭代试探点为 $\mathbf{t}_k \in M(\mathbf{x}_k, \Delta_k)$, 任意点 $\mathbf{t}_k \notin \bar{\mathcal{F}}_k$ 均为网格未过滤点(反之,任意点 $\mathbf{t}_k \in \bar{\mathcal{F}}_k$ 均被过滤掉),任意点集 $P_k \subset \bar{\mathcal{F}}_k$ 均为网格孤立过滤点. 可行覆盖解集为 $\{\mathbf{x} \in X \cap \Omega \cap S_k : f(\mathbf{x}) = f_k^f\}$, 不可行覆盖解集为 $\{\mathbf{x} \in X \cap S_k : h(\mathbf{x}) = h_k^l, f(\mathbf{x}) = f_k^l\}$.

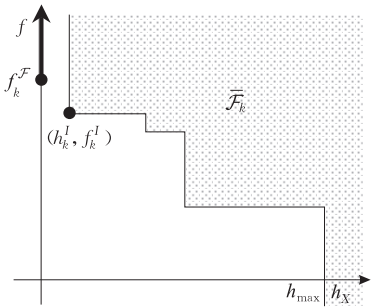


图 7 GPS Filter 的二目标优化^[40]

与一般 GPS 算法类似, GPS 过滤法首先在某个点集上选择有最小 h_X 或 f 的初始点(在搜索阶段可以采用多种搜索策略), 并构造初始覆盖集 S_0 及相应的 \mathcal{F}_0 . 对当前网格 $M(S_k, \Delta_k)$ 的搜索和筛选结果

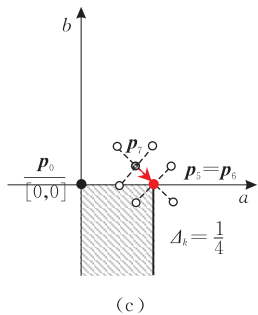
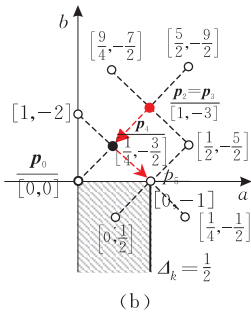
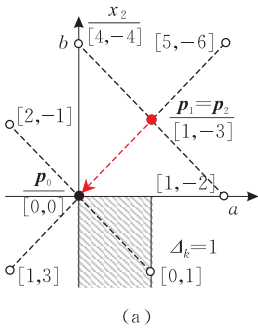


图 8 GPS Filter 算法对上述问题的迭代搜索和筛选示意图^[40]

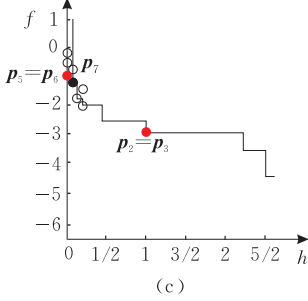
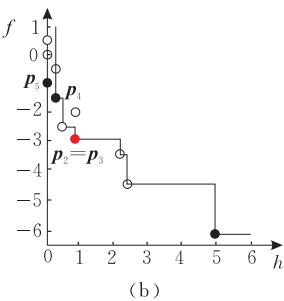
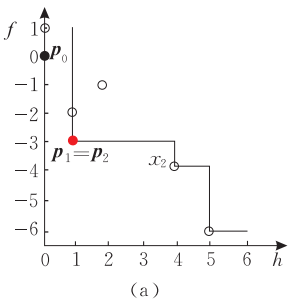


图 9 GPS Filter 算法连续迭代过程中约束违背值 h 和目标函数值 f 的变化趋势^[40]

可能包含以下 3 种情况: (1) 新的覆盖解, 步长增加; (2) 未过滤点, 步长保持不变; (3) 孤立孤立点, 步长减小(然而在减少步长 Δ 之前, 也允许先在过滤点附近区域搜索, 以跳出局部极小, 可以将该方式称为次优搜索). 由 \mathcal{F}_k 和所有不可行未过滤点构造 \mathcal{F}_{k+1} , 并移除 \mathcal{F}_{k+1} 中的被支配点, 重复上述过程直至步长 Δ 收敛或达到规定的迭代次数.

考虑如何利用 GPS 过滤法求解如下简单的线性规划问题^[26, 40]:

$$\min_{\mathbf{x}=(a,b)'} -a-2b, \text{ s. t. } 0 \leq a \leq 1, b \leq 0.$$

已知最优解为 $\hat{\mathbf{x}} = (1, 0)'$. 假设约束条件作为黑盒函数出现, 因此无法确定是线性还是非线性的. 设初始搜索点从 $\mathbf{x}_0 = (0, 0)'$ 开始, 初始步长 $\Delta_0 = 1$, 4 个对角方向 $\pm(1, 1)'$ 和 $\pm(1, -1)'$ 被用于构造正基.

已经表明 GPS Barrier 法从原点开始的 4 个方向均不可行, 因此迭代始终停留在原点^[26].

图 8 给出 GPS Filter 算法最初几次迭代搜索和筛选的示意图, 图中数值按 $[h(\mathbf{x}), f(\mathbf{x})]$ 标记.

图 8(a) 的初始迭代仅找到试探点 $(1, -1)'$ 可行, 然而 $(1, -1)'$ 被 \mathbf{x}_0 支配; 点 $\mathbf{p}_1 = (1, 1)'$ 支配其它两个试探点, 因此是未过滤点. 现在从点 $\mathbf{x}_1 = \mathbf{p}_1 = (1, 1)'$ 开始, 尽管 $(0, 2)'$ 和 $(2, 2)'$ 有更小的 f 值, 但其 h 值均大于点 \mathbf{p}_1 , 因此不能支配点 \mathbf{p}_1 , 故筛选中心保持不变. 在 $\mathbf{p}_2 = \mathbf{p}_1$ 筛选的结果将导致所有相邻点均被过滤掉, 因此 \mathbf{p}_2 是网格孤立过滤点.

图 8(b) 从点 $p_3 = p_2 = (1, 1)'$, 以步长 $\Delta_3 = 1/2$ 开始第 3 次迭代. 经连续两次迭代找到未过滤点 $p_4 = (1/2, 1/2)'$ 和孤立过滤点 $p_5 = (1, 0)'$. 如图 8(c), 最终将在 $p_7 \rightarrow p_6 (= p_5)$ 方向上迭代搜索, 直到 $\Delta_k \rightarrow 0$ (实际上仅需非常少的有限次迭代便可以达到收敛). 图 9 同时给出上述迭代过程 h 和 f 值的变化趋势.

3.5 向混合约束条件的推广

在包含连续、离散(或分类)等混合变量时, 考虑如下划分:

$$\Omega = \Omega^c \times \Omega^d \subset \mathfrak{R}^{n^c} \times Z^{n^d},$$

其中 $\Omega^c = [\ell, u](\ell, u \in \mathfrak{R}^{n^c} \cup \{\pm \infty\}, \ell < u)$ 为连续变量约束空间; Ω^d 为离散变量约束空间(通常将分类变量映射到离散点集). 任意迭代中心点:

$$x_k = (x_k^c, x_k^d) \in \mathfrak{R}^{n^c} \times Z^{n^d}.$$

在混合变量约束条件下, 迭代过程需要在当前局部最优点 x_k 所在网格 M_k 及其邻域扩展搜索, 以寻求新的覆盖解^[9]. 关键在于如何定义搜索网格 M_k 和筛选集(poll set)以进行扩展邻域搜索.

定义 21. 设 $\Delta_k \in \mathfrak{R}_+$ 为网格尺寸; 基矩阵 $B \in \mathfrak{R}^{n^c \times n^c}$, 生成矩阵 $G_j \in Z^{n^c \times n^c}$ ($j=1, 2, \dots, j_{\max} < +\infty$) 是有限的矩阵集合, 模式

$$P_j = BG_j \in \mathfrak{R}^{n^c \times n^c};$$

$$M(\Delta_k) = \bigcup_{j=1}^{j_{\max}} \{x_k^c + \Delta_k P_j z : z \in Z^{n^c}\}.$$

定义网格:

$$M_k = (M(\Delta_k) \times Z^{n^d}) \cap \Omega.$$

定义 22. $\hat{x} = (\hat{x}^c, \hat{x}^d) \in \Omega$ 是 f 的局部最小 $\Leftrightarrow \exists \epsilon > 0, \forall v \in \bigcup_{y \in N(\hat{x})} (([\ell, u] \cap B(\epsilon, y^c)) \times y^d)$: $f(\hat{x}) < f(v)$. 其中 $N(\hat{x}) \subset \Omega$ 是 \hat{x} 邻域的有限点集, $B(\epsilon, y^c)$ 是 y^c 的 ϵ 半径邻域.

网格 M_k 的搜索步骤可以保证算法的全局收敛性. 当网格搜索失败时, 需要考虑更为严格的筛选步骤, 在筛选集中检查其覆盖解是否是局部极小值, 以保证算法的局部收敛性(取决于对筛选集的定义). 筛选集包括 3 部分: (1) 连续变量 x_k 网格邻域 $N^c(x_k)$ 的常规搜索(典型筛选法); (2) 离散变量邻集 N_k 的搜索(离散值的邻集); (3) 覆盖解邻集 $\epsilon(y)$ 的扩展筛选(extended poll)(当邻集中某些点 y 的值 $f(y)$ 接近覆盖解时). 因此, 从 x_k 的邻域到扩展筛选邻集的搜索, 可能存在更好的候选点, 从而导致下降.

首先构造 \mathfrak{R}^{n^c} 的最大正基矩阵集合:

$$B = \{[b_1 \ \cdots \ b_n \ b_{n+1} \ \cdots \ b_{2n}]\}.$$

由最大正基定义, B 可以分解为两个 $n^c \times n^c$ 的非奇异对角矩阵.

定义 23. 定义连续变量 x_k 的网格邻域:

$$N^c(x_k) = \{x_k + \Delta_k(b, 0) \in \Omega : b \in B_k(x_k) \in B\},$$

其中, $B_k(x_k)$ 的选择同时取决于 k 和 x_k .

显然 $N^c(x_k) \subset M_k$. 同时, 最大正基的选择保证即使点趋近约束边界时, 仍然有足够多的可选下降方向, 以便能再次回到可行区域.

离散变量邻集 N_k 在 Ω^d 中取值. 取决于用户的选择, N_k 通常等于 $N(x_k)$ (定义 22), 并且在每次迭代中可以改变.

若在 $N^c(x_k)$ 和 N_k 均不能找到改善点, 则需对 N_k 邻集进行扩展筛选. 通过对点 x_k 的局部扰动(参数 $\xi > 0$), N_k 的邻集离散点 y 所在连续邻域可能产生更好的解. 然而, 当 N_k 的所有邻集离散点 y 都满足 $f(y) > f(x_k) + \xi$, 表明 x_k 已是局部极小, 需收缩网格并在 x_k 点重新迭代; 或者当 $f(y) \leq f(v) (\forall v \in N^c(y))$ 时, y 是局部极小, 直接从 y 进入下一次迭代.

其它情况下, 假设点 y 的扩展筛选序列为 $\{y^j\}_{j=0}^J, y^0 = y$, 扩展筛选仅须保证 $f(y^j) < f(y^{j-1})$. 当 $f(y^j) \leq f(v) (\forall v \in N^c(y^j))$ 时, y^j 已是局部极小, 筛选终止, 标记该点下标为 J , 即终点 z ; 若在筛选过程中, 存在 $f(y^j) < f(x_k)$, 则直接从该点进入下一次迭代.

以上步骤将产生单调递减的扩展筛选序列 $\{y^0, y^1, \dots, y^J = z\}$. 因此, 从初始点 y^0 开始, 只要能找到下降方向, 使得 $f(y^j) < f(y^{j-1})$ 即可; 除最终点 z , 不必估计 y^j 的所有邻域 $N^c(y^j)$ (为保证 $N^c(z)$ 在约束条件范围内, 可能需要定义点 z 的新邻域集). 因为只在当前点 y^j 的有限点邻域内搜索, 可以保证总的扩展筛选次数 J 必然是有限的.

定义 24. 定义点 x_k 的筛选集:

$$X_k^\xi = N^c(x_k) \cup N_k \bigcup_{\substack{y \in N_k \\ f(x_k) \leq f(y) \leq f(x_k + \xi)}} \epsilon(y).$$

图 10 给出上述的扩展筛选序列产生过程. 图 11 是在混合变量 (x_1^c, x_2^c, x^d) 上得到的点 x_k 的筛选集, y_1^0, y_1^1, y_2^0 为离散变量值, $a \sim l$ 为连续变量值. x_k 的离散点邻集 $N_k = N(x_k) = \{x_k, y_1^0, y_2^0\}$, 并且

$$N^c(x_k) = \{f, g, h\}, \quad N^c(y_1^0) = \{d, e, y_1^1\},$$

$$N^c(y_1^1) = \{a, b, c\}, \quad N^c(y_2^0) = \{i, j, k, l\}.$$

若 $f(\mathbf{x}_k) \leq f(\mathbf{y}_1^0) < f(\mathbf{x}_k) + \xi < f(\mathbf{y}_2^0)$, 则筛选集包括 $N^c(\mathbf{x}_k)$ 和 $N^c(\mathbf{y}_1^0)$. 再设

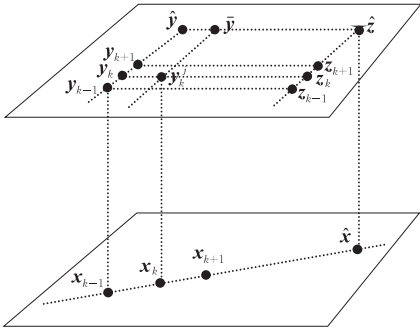
$$f(\mathbf{y}_1^1) < f(\mathbf{y}_1^0), \quad f(a) > f(\mathbf{y}_1^1),$$
$$f(b) > f(\mathbf{y}_1^1), \quad f(c) > f(\mathbf{y}_1^1).$$


图 10 扩展筛选搜索序列^[9]

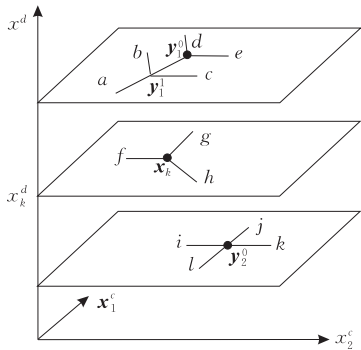


图 11 \mathbf{x}_k 的网格邻域点集^[9]

由定义 24, 得点 \mathbf{x}_k 的筛选集:

$$X_k^\xi = \{f, g, h\} \cup \{\mathbf{x}_k, \mathbf{y}_1^0, \mathbf{y}_2^0\} \cup \varepsilon(\mathbf{y}),$$

其中 $\varepsilon(\mathbf{y}) = \{\mathbf{y}_1^1\} \cup \{a, b, c\}$.

考虑如下形式的混合约束优化问题^[9]:

$$\min f(\mathbf{x}) = f(a, b, x^d)$$
$$= g(a, b) \cdot (1 - x^d) + h(a, b) \cdot x^d,$$
$$g(a, b) = a^2 + b^2; \quad h(a, b) = a^2 b + a(1 - b)$$

s. t. $-2 \leq a, b \leq 2; \quad x^d = 0, 1.$

首先定义 \mathbf{x} 的网格邻域:

$$N^c(\mathbf{x}) = \mathbf{x} + \Delta_k \{(0, 1, 0), (0, -1, 0), (5, 0, 0),$$
$$(-7, 0, 0)\} \text{ (对应最大模式);}$$

当 $\mathbf{x} = (2\Delta_k, 1 - \Delta_k, 1)$ 时:

$$N^c(\mathbf{x}) = \mathbf{x} + \Delta_k \{(0, -1, 0), (5, 1, 0), (-7, 1, 0)\}$$

(对应最小模式).

同时定义 \mathbf{x} 的离散点邻集:

$$N(\mathbf{x}) = \{(a, b, 1 - x^d), (a, b, x^d)\}.$$

初始值 $\xi = 1; \mathbf{x}_0 = (1, 0, 0), \Delta_0 = 1/4$; 若迭代成功, $\Delta_{k+1} = \Delta_k$, 否则 $\Delta_{k+1} = \Delta_k/2$.

参照表 1, 从初始点 $\mathbf{x}_0 = (1, 0, 0)$ 开始, $k = 0, 1$ 两次迭代均成功. $k = 2$ 时, $\mathbf{x}_2 = (1/2, 0, 0)$ 邻域无法找到改善点, 转入其离散邻集 $N(\mathbf{x}_2) = \{(1/2, 0, 1)\}$. 在 $(1/2, 0, 1)$ 邻域筛选得到下降序列 $\{\mathbf{y}_2^j\}_{j=0}^J, \mathbf{z}_2 = \mathbf{y}_2^J = (1/2, 3/4, 1)$, 因此在 \mathbf{z}_2 的邻域扩展筛选. 扩展筛选的结果仍未找到改善点, 算法返回点 \mathbf{x}_2 , 递减步长, 并在 \mathbf{x}_2 再次迭代. 可以看出, 步 2、步 3 重复筛选和扩展筛选的结果, 最终将使得 $\Delta_k \rightarrow 0$, 并且函数在点 $(0, 0, 0)$ 收敛, $f_{\min} = 0$.

$k = 2$ 时, 在强化条件下也可选取 $N^c(\mathbf{y}_2^0)$ 中的最小值 $\mathbf{y}_2^1 = (-5/4, 0, 1) (f(\mathbf{y}_2^1) = -5/4)$, 以保证 \mathbf{y}_2^0 梯度的最速下降, 因此, 不需要再次计算下降序列.

表 1 从初始点 (1,0,0) 开始的主要迭代步骤

k	Δ_k	\mathbf{x}_k	$\mathbf{x}_k + \Delta_k(0, 1, 0)$	$\mathbf{x}_k + \Delta_k(0, -1, 0)$	$\mathbf{x}_k + \Delta_k(5, 0, 0)$	$\mathbf{x}_k + \Delta_k(-7, 0, 0)$
0	1/4	(1, 0, 0)	(1, 1/4, 0)	(1, -1/4, 0)	(9/4, 0, 0)	(-3/4, 0, 0)
1		1	17/16	17/16	81/16	<u>9/16</u>
		(-3/4, 0, 0)	(-3/4, 1/4, 0)	(-3/4, -1/4, 0)	(1/2, 0, 0)	(-10/4, 0, 0)
2		9/16	10/16	10/16	<u>1/4</u>	100/16
		<u>(1/2, 0, 0)</u>	(1/2, 1/4, 0)	(1/2, -1/4, 0)	(7/4, 0, 0)	(-5/4, 0, 0)
		<u>1/4</u>	5/16	5/16	49/16	25/16
扩展筛选			$\mathbf{y}_k = \mathbf{y}_k^0$	\mathbf{y}_k^1	$\mathbf{y}_k^2 \cdots$	$\mathbf{z}_k = \mathbf{y}_k^J$
		$N(\mathbf{x}_2) \rightarrow$	<u>(1/2, 0, 1)</u>	(1/2, 1/4, 1)	(1/2, 1/2, 1)	(1/2, 3/4, 1)
		$f(\mathbf{x}_2) + \xi = (1/4 + 1) >$	<u>1/2</u>	7/16	6/16	<u>5/16</u>
			$\mathbf{x}_k + \Delta_k(0, -1, 0)$	$\mathbf{x}_k + \Delta_k(5, 1, 0)$	$\mathbf{x}_k + \Delta_k(-7, 1, 0)$	
		$N^c(\mathbf{z}_k)$	(1/2, 1/2, 1)	(7/4, 1, 1)	(-5/4, 1, 1)	
			6/16	49/16	25/16	
3	1/8	(1/2, 0, 0)	...			
		<u>1/4</u>	...			

3.6 在紧致集上的搜索

鉴于 GPS 类算法存在的问题, 网格自适应直接搜索 MADS^[10] 通过引入筛选尺寸参数 Δ_k^p , 使得可以在紧致(dense)方向集中进行局部搜索.

定义 25. 定义具备如下属性的参数 Δ_k^p 为筛选尺寸参数: (1) $\Delta_k^p \geq \Delta_k^m$; (2) $\lim \Delta_k^m = 0$ 当且仅当 $\lim \Delta_k^p = 0$.

图 12 和图 13 分别给出 GPS 和 MADS 在 \Re^2 上

的网格 M_k 及部分搜索方向集 D_k 的示意图. 帧模式^[28-29] $P_k = \{\mathbf{x}_k + \Delta_k^m \mathbf{d} : \mathbf{d} \in D_k\} = \{\mathbf{p}^1, \mathbf{p}^2, \mathbf{p}^3\} \subset M_k$.

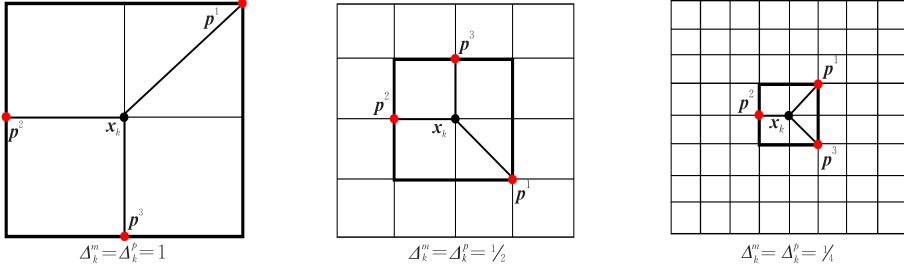


图 12 GPS 在 \mathbb{R}^2 上的搜索方向集示意图(最多允许构造 14 个搜索方向集)^[10]

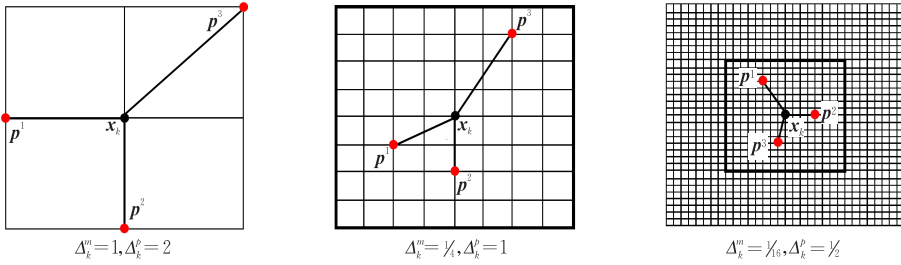


图 13 MADS 在 \mathbb{R}^2 上的搜索方向集示意图(右端网格允许多达 104 个方向集)^[10]

参数 Δ_k^p 的引入,使得可以采用不同的方法(确定或随机),以产生搜索方向 D_k 的紧致集. 基于下三角矩阵的随机构造法 LTMADS^[10],能以概率 1 产生 \mathbb{R}^n 的紧致方向集;特别是针对超正切锥(Hyper-tangent Cone),在陷入边界局部极小时,将有更多的网格点方向可供选择,因此更增加了迭代跳出局部极小点的可能性. 以下为 LTMADS 算法:

1. 参数设置: 令生成矩阵 $\mathbf{G} = \mathbf{I}$, 基矩阵 $\mathbf{D} = \mathbf{Z} = [\mathbf{I} \quad -\mathbf{I}]$. $\tau=4$; $\omega_0 = -1$, $\omega_1 = 1$. 初始化网格和筛选尺寸参数 $\Delta_0^m = \Delta_0^p = 1$;

2. 更新规则: (1) $\Delta_{k+1}^m = \Delta_k^m/4$, 若 \mathbf{x}_k 是局部极小值; (2) $\Delta_{k+1}^m = 4\Delta_k^m$, 若找到改善点, 且 $\Delta_k^m \leq 1/4$; (3) $\Delta_{k+1}^m = \Delta_k^m$, 其它情况;

3. 正基构造: (1) 等概率地从 $\pm 1/\sqrt{\Delta_k^m}$ 选取主对角元素, 从开区间 $(-1/\sqrt{\Delta_k^m}, +1/\sqrt{\Delta_k^m})$ 选取下三角元素, 构造下三角矩阵 \mathbf{B} ; (2) 矩阵 \mathbf{B} 的行列随机置换, 设 $\mathbf{d}_p^q = \mathbf{B}(i_p, j_q)$; (3) 由 $\mathbf{B} = [\mathbf{d}^1 \cdots \mathbf{d}^n]$ 构造最小正基: $D_k = \{\mathbf{d}^1, \dots, \mathbf{d}^n, \mathbf{d}^{n+1}\}$, 其中 $\mathbf{d}^{n+1} = -\sum_{i=1}^n \mathbf{d}^i$ (命题 4), 设 $\Delta_k^p = n \sqrt{\Delta_k^m} \geq \Delta_k^m$; 构造最大正基 $D_k = \{\mathbf{d}^1, \mathbf{d}^2, \dots, \mathbf{d}^{2n}\}$, 其中 $\mathbf{d}^{n+i} = -\mathbf{d}^i$ (命题 2), 设 $\Delta_k^p = \sqrt{\Delta_k^m} \geq \Delta_k^m$.

定理 6. 设 $\hat{\mathbf{x}} \in \Omega$ 是由 LTMADS 算法产生的局部极小点, 则筛选方向集以概率 1 在 $\hat{\mathbf{x}}$ 的超正切锥 $T_\Omega^H(\hat{\mathbf{x}})$ 中紧致(文献[10]定理 4.2).

图 13 给出在不同的 Δ_k^m (以及相应的 Δ_k^p) 下, \mathbb{R}^2 上 3 种可能的最小正基:

按正基定义, 图 12 的 8 个网格邻域点, 仅允许 GPS 构造 14 个搜索方向集, 并且在每次迭代中不会改变.

$$\begin{aligned} D_1 &= \{(-1, 0)', (0, -1)', (1, 1)'\}, \\ D_2 &= \{(-2, -1)', (0, 2)', (2, 3)'\}, \\ D_3 &= \{(4, 0)', (-1, -4)', (-3, 4)'\}. \end{aligned}$$

注意到 $\Delta_k^m \rightarrow 0$ 当且仅当 $\Delta_k^p \rightarrow 0$. 故当 $\Delta_k^m \rightarrow 0$ 时, MADS 允许在 Δ_k^p 确定的网格邻域选择任意方向; 并且将逐渐增加, 直至紧致(图 13 的右端网格允许多达 104 个方向集).

4 讨论和推广研究

模式搜索法并非如我们期望的那样, 能解决所有优化问题, 因此以下讨论模式搜索对实际工程问题的适用性并进行推广研究. 首先指出模式搜索应用到非光滑随机优化问题时可能达到的期望结果; 其次讨论模式搜索如何与传统优化技术结合, 以便加速搜索和迭代收敛过程.

4.1 对非光滑问题的适用性

需要强调, 采用模式搜索法的初衷是因为其简单性和对随机优化问题的广泛适用性, 只需要对目标函数逐个搜索点进行估值, 最终总能找到最优解, 因此特别适合于工程应用中的很多坏函数(badly behaved)优化. 在这些问题中, 目标函数可能不可微, 并且受到因物理测量而导致的不确定随机噪声的影响, 没有任何导数信息可供利用, 即使有限差分逼近也是不准确甚至完全错误的. 然而, 在采用模式

搜索法解决这些问题时,在期望的结果和实际计算结果之间存在一定差异.我们首先解释模式搜索法如何跨过大多数基于梯度的优化技术无法跨越的鸿沟,然后说明模式搜索法真正能做什么以及它们在求解这些问题时能达到什么样的期望.

当应用基于导数的方法求解因随机噪声而导致可能存在测量误差的目标函数时,这些函数会受到低幅高频噪声的影响,并且噪声会因差分进一步放大.通常的做法是,有目的地对导数采用较粗的有限差分逼近,来消除噪声对目标函数的影响.在模式搜索法中,若采用较大的步长控制参数进行采样,能够隐藏噪声对目标函数的影响.当采样足够粗,即步长超过噪声的尺度范围时,噪声对目标函数的影响就被覆盖掉了;在较细尺度下的高频特征因而被忽略了,转而代之的是噪声目标函数的较平滑版本(虽然仍是呈锯齿状).而当步长逼近 0 时,目标函数的高频特征再次呈现,并通过较小步长的搜索得以解析.

因此这就能够解释步长控制参数的更新规则.可以看出,模式搜索法仅在有限步成功迭代,然而有无限次失败迭代,以便可以到达局部极小值所在的吸引域(basin of attractions).在该吸引域,步长足够小以至于可以停止迭代搜索过程.当然,我们也期望对较大的步长,能够找到局部极小点所在的较大吸引域,以便避免停留在因随机噪声而导致的众多不感兴趣的局部极小吸引域,并且跳出这些吸引域继续进行迭代搜索,最终找到全局最优点.

然而,即使模式搜索法可以应用,到目前为止,仍然没有任何理论依据可以保证应用到非光滑问题时它能取得成功.也就是说,对非光滑随机优化问题,模式搜索法可能不会收敛到一个稳定点——特别是噪声非连续和无特定结构的情况.而且,在非光滑情况下也没有任何机制保证一定可以寻找到下降方向.例如典型的 Dennis-Woods 函数及其变种,这些情况下无法应用模式搜索法.

事实上,对非光滑随机优化问题,人们并不期望一定要找到最终的全局最优解(除非时间和成本允许).对大多数工程应用来说,一定程度上对解的提升就足够了.因此,这就解释了对非光滑性和随机噪声不敏感的模式搜索法为什么在实际应用中,比大多数基于梯度的方法,甚至现有很多新算法,如遗传和进化、神经网络、蚁群算法等,更适合非光滑随机优化问题.

4.2 加速搜索的其它方法

其它的启发式方法也可以集成到模式搜索法中

以加速搜索过程,例如前文的 Rosenbrock 香蕉函数,通过对 $2n$ 个正交矢量进行旋转,可以使得至少有一个方向与函数的局部拓扑更一致.另外,基于仿真的优化对响应表面(Response Surface Methodology)的拟合采用最小二乘法^[41-42],而导数无关优化 DFO(Derivative Free Optimization)则通过插值构造模型^[43-44].

当前很多研究专注于基于采样的导数无关方法来构造模型,利用采样点的一阶或二阶信息来捕获目标函数的局部拓扑.模型首先由最简单的线性模型来拟合;当有更多的采样点获得时,就可以构造更复杂的高阶模型.类似 L_k 的角色,这些启发式算法或基于模型的搜索都试图通过简单模型来对原始的复杂模型进行逼近,以便可以加速搜索,同时也避免对原有复杂目标函数的估值.如果目标函数受到噪声干扰,某些情况下要估计目标函数值可能是非常困难甚至根本不准确的.

其它一些策略也可以引入以加速收敛,例如在每次搜索中从 D 中选择一个新的方向集 $d_k^{(i)}$;并且每个方向的搜索步长 $\Delta_k^{(i)}$ 可以独立进行修改.这也使得不同方向的搜索尺度可以独立进行扩展,从而与目标函数的局部拓扑更好地吻合.在非成功迭代步,收缩步长参数 θ 也不必限定为严格低于 1,可以等于甚至大于 1,以使得在当前非成功迭代点可以进行局部扩展搜索,这通常会导致发现一个更好的候选点,进而跳出局部极小.这种思想在 GMVP 和 MADS 中均得到应用.当然,上述对模式搜索法的修改不应该影响其基本的收敛性准则,即 $\lim_{k \rightarrow +\infty} \Delta_k = 0$,对至少一个非成功迭代子序列 $\kappa \subseteq U(k \in \kappa)$.

5 结 论

对直接搜索法、特别是模式搜索法的历史回顾及算法的理论分析和研究,使得可以按统一方式对工程应用中的很多优化问题进行求解.模式搜索法通常作为上述类型的非线性规划问题求解的首选,但不是唯一的选择.对形态较好(如连续、可微)的目标函数和约束条件,仍然会考虑用传统的规划方法求解.然而在传统方法不可行时,模式搜索法却成为唯一可用的方法.对模式搜索法在实际工程问题中的应用及未来发展方向需做如下说明:

(1) 模式搜索法不是保证寻求到全局最优点(尽管前文的收敛性分析表明在足够多的迭代次数下这也是可行的),而是试图在有限的计算资源下,

获得满足当前应用所需的较优解. 因此这有别于从数学角度对问题模型的精确求解.

(2) 模式搜索法是导数无关法, 但这并不意味着不需要使用方向导数信息, 类似有限差分或基于模型的插值逼近可以加速迭代过程在函数局部区域的搜索速度. 然而这种方式能否保证收敛到全局稳定点, 仍然缺乏足够的理论依据. 在非连续、不可微, 甚至黑盒函数情况下, 对下降梯度的有限差分或插值逼近反而可能导致迭代过程快速收敛到局部极小点.

(3) 全局收敛性与局部收敛性的关系. 全局收敛意味着从任意起始点开始迭代搜索, 一阶收敛到某个稳定点 x_* , 即 $\nabla f(x_*) = 0$ (假设 f 可微). 相反, 局部收敛则指一个点向局部极小值的逼近, 即点向其吸引域的收敛速度. 通常, 我们认为全局收敛是指收敛到一个全局极小点 (拟牛顿方法例外). 然而目前几乎所有对模式搜索的研究都关注于其全局收敛性, 特别是一阶优化的必要条件. 当然, 一阶收敛并不意味着极小点满足二阶收敛条件 (Hessian 矩阵为正定或半正定); 对模式搜索的迭代过程在函数局部区域的微观行为及局部收敛性分析以及局部收敛性对全局收敛性影响的研究尚不多见.

(4) 模式搜索法因为采用迭代搜索, 通常会导致较高的计算复杂度和存储空间需求, 并且在向大规模和高维空间问题推广时尚存在较多问题. 目前提出的并行模式搜索或许可以通过并行计算^[15]在一定程度上降低计算复杂度, 然而这需要问题空间进行划分以适合并行计算.

(5) 同样思路, 在模式搜索法的搜索阶段, 允许引入更多搜索策略甚至并行搜索, 以便可以从解区域的多个初始点开始寻求全局最优. 例如, 对工程应用中的复杂函数估计, 可以通过代理函数 (Surrogate) 寻找次优解; 传统方法如模拟退火、遗传和进化等也可以在搜索阶段得到应用. 在筛选阶段, 也允许每次采用不同的筛选方向集或仅进行部分筛选以加速收敛、减少计算.

(6) 统一模式搜索及其扩展算法, 并能应用于求解复杂工程问题的软件^{①②}及硬件实现. 这对实际应用及该领域的进一步发展将产生重大影响.

致 谢 作者对审稿人的宝贵意见及编辑的辛勤劳动表示衷心感谢!

参 考 文 献

[1] Lewis R M, Torczon V, Trosset M W. Direct search meth-

- ods: then and now. *Journal of Computational and Applied Mathematics*, 2000, 124(1-2): 191-207
- [2] Armijo L. Minimization of functions having Lipschitz continuous first partial derivatives. *Pacific Journal of Mathematics*, 1966, 16(1): 1-3
- [3] Goldstein A A. *Constructive Real Analysis*. New York, NY: Harper & Row, 1967
- [4] Wolfe P. Convergence conditions for ascent methods. *SIAM Review*, 1969, 11(2): 226-235
- [5] Spendley W, Hext G R, Himsworth F R. Sequential application of simplex designs in optimization and evolutionary operation. *Technometrics*, 1962, 4(4): 441-461
- [6] Torczon V. Multi-directional search: A direct search algorithm for parallel machines [Ph. D. dissertation]. Department of Mathematical Sciences, Rice University, Houston, TX, 1989
- [7] Torczon V. On the convergence of pattern search algorithms. *SIAM Journal on Optimization*, 1997, 7(1): 1-25
- [8] Audet C, Dennis J E. Analysis of generalized pattern searches. *SIAM Journal on Optimization*, 2003, 13(3): 889-903
- [9] Audet C, Dennis J E. Pattern search algorithms for mixed variable programming. *SIAM Journal on Optimization*, 2000, 11(3): 573-594
- [10] Audet C, Dennis J E. Mesh adaptive direct search algorithms for constrained optimization. *SIAM Journal on Optimization*, 2006, 17(2): 188-217
- [11] Box G E P. Evolutionary operation: A method for increasing industrial productivity. *Applied Statistics*, 1957, 6(2): 81-101
- [12] Nelder J A, Mead R. A simplex method for function minimization. *The Computer Journal*, 1965, 7(4): 308-313
- [13] McKinnon K I M. Convergence of the Nelder-Mead simplex method to a non-stationary point. *SIAM Journal on Optimization*, 1998, 9(1): 148-158
- [14] Lagarias J C, Reeds J A, Wright M H. Convergence properties of the Nelder-Mead simplex method in low dimensions. *SIAM Journal on Optimization*, 1998, 9(1): 112-147
- [15] Dennis J E, Torczon V. Direct search methods on parallel machines. *SIAM Journal on Optimization*, 1991, 1(4): 448-474
- [16] Torczon V. On the convergence of the multi-directional search algorithm. *SIAM Journal on Optimization*, 1991, 1(1): 123-145
- [17] Dividon W C. Variable metric method for minimization. *SIAM Journal Optimization*, 1991, 1(1): 1-17
- [18] Hooke R, Jeeves T A. Direct search solution of numerical and statistical problems. *Journal of the Association for Computing Machinery (ACM)*, 1961, 8(2): 212-229

① Abramson M A. NOMADm optimization software (v4.6). <http://www.ait.edu/en/faculty/MAbramson/>, 2007

② The MathWorks. GADS — Genetic algorithm and direct search toolbox — direct Search and simulated annealing Tools. <http://www.mathworks.com/products/gads/>, 2006—2008

- [19] Rosenbrock H H. An automatic method for finding the greatest or least value of a function. *The Computer Journal*, 1960, 3(3): 175-184
- [20] Swann W H. Report on the development of a new direct search method of optimization. I. C. I Central Instrument Laboratory; Report Note 64/3, 1964
- [21] Swann W H. Direct search method//Murray eds. *Numerical methods for unconstrained optimization*. London and New York: Academic Press, 1972: 13-28
- [22] Powell M J D. An efficient method for finding the minimum of a function of several variables without calculating derivatives. *The Computer Journal*, 1964, 7(2): 155-162
- [23] Polak E. *Computational Methods in Optimization: A Unified Approach*. New York: Academic Press, 1971
- [24] C  a J. *Optimisation: Th  orie et algorithmes*. Dunod, Paris, 1971
- [25] Berman G. Lattice approximations to the minima of functions of several variables. *Journal of the Association for Computing Machinery (ACM)*, 1969, 16(2): 286-294
- [26] Lewis R M, Torczon V. Pattern search algorithms for bound constrained minimization. *SIAM Journal on Optimization*, 1999, 9(4): 1082-1099
- [27] Lewis R M, Torczon V. Pattern search algorithms for linear constrained minimization. *SIAM Journal on Optimization*, 2000, 10(3): 917-941
- [28] Coope I D, Price C J. Frame-based methods for unconstrained optimization. *Journal of Optimization Theory and Application*, 2000, 107(2): 261-274
- [29] Coope I D, Price C J. On the convergence of grid-based methods for unconstrained optimization. *SIAM Journal on Optimization*, 2001, 11(4): 859-869
- [30] Davis C. Theory of positive linear dependence. *American Journal of Mathematics*, 1954, 76(4): 733-746
- [31] Wen-Ci Yu. Positive basis and a class of direct search techniques. *Scientia Sinica, Special Issue of Mathematics*, 1979, 1(26): 53-68
- [32] Lewis R M, Torczon V. Rank ordering and positive bases in pattern search algorithms. NASA Langley Research Center, Hampton, Virginia; ICASE Technical Report 96-71, 1996
- [33] Bertsekas D P. *Convex analysis and optimization*. Athena Scientific, 2003
- [34] Kolda T G, Lewis R M, Torczon V. Optimization by direct search: New perspectives on some classical and modern methods. *SIAM Review*, 2003, 45(3): 385-482
- [35] Klingman W R, Himmelblau D M. Nonlinear programming with the aid of a multiple-gradient summation technique. *Journal of the Association for Computing Machinery (ACM)*, 1964, 11: 400-415
- [36] Lucidi S, Sciandrone M, Tseng P. Objective-derivative-free methods for constrained optimization. *Mathematics Programming*, 2002, 92: 37-59
- [37] Conn A R, Gould N I M, Toint Ph L. A globally convergent augmented Lagrangian algorithm for optimization with general constraints and simple bounds. *SIAM Journal on Numerical Analysis*, 1991, 28(2): 545-572
- [38] Lewis R M, Torczon V. A globally convergent augmented Lagrangian pattern search algorithm for optimization with general constraints and simple bounds. *SIAM Journal on Optimization*, 2002, 12: 1075-1089
- [39] Fletcher R, Leyffer S. Nonlinear programming without a penalty function. *Mathematics Programming*, 2002, 91(A): 239-269
- [40] Audet C, Dennis J E. A pattern search filter method for nonlinear programming without derivatives. *SIAM Journal on Optimization*, 2004, 14(4): 980-1010
- [41] Box G E P. The exploration and exploitation of response surfaces: Some general considerations and examples. *Biometrics*, 1954, 10: 16-60
- [42] Box G E P, Draper N R. *Empirical Model-Building and Response Surfaces*. New York: Wiley, 1987
- [43] Conn A R, Toint P L. An algorithm using quadratic interpolation for unconstrained derivative free optimization//*Nonlinear Optimization and Applications*. New York: Kluwer Academic/Plenum Publishers, 1996: 27-47
- [44] Conn A R, Scheinberg K, Toint P L. A derivative free optimization algorithm in practice//*Proceedings of the 7th AIAA/USAF/NASA/ISSMO Symposium on Multi-Disciplinary Analysis and Optimization*. St. Louis, MO, 1998



HUANG Tian-Yun, born in 1973, Ph. D., associate professor. His current research interests include video flow analysis, optimal video transmission, quality of streaming video, and media adaptation, etc.

Background

The modern constrained optimization problems in engineering are becoming more and more complex, most of them

are simulation-driven stochastic models, the objectives may be non-smooth, thus no information of the derivatives are

available or trustworthy. The constraints may be linear or non-linear, discrete sets, or even black-box functions generated from stochastic simulations or codes. However, many conventional optimization techniques are only applicable to continuous and differentiable convex functions, and they have no use in these cases. The purpose of this work is to reveal another end of the spectrum in the optimization community: Direct searches, especially pattern searches and their evolutions.

The direct searches can be traced back to 1950s, from the Evolutionary Operation (EVOP) of Box to the Simplex of Nelder and Mead, from coordinate search of Davidon to the direct search of Hooke and Jeeves. Direct searches, especially pattern searches are always used as the first resort to many non-linear programming (NLP) problems in engineering. Actually, they work very well in practice, but lack of the theoretical fundamentals till 1990s, only as heuristics.

The direct searches seem to be revived since the new century, some elaborate theories are constructed and provide rigorous guarantees of convergence for a large number of direct search methods, such as generalized pattern searches (GPS), generalized mixed variables programming (GMVP) and mesh adaptive direct search (MADS), etc. Always, the

direct searches remain effective choices, and sometimes the only choice, for several difficult optimization problems.

The authors gives a unified mathematical description and convergence analysis of pattern searches in this work, from bound and linear constrained (GPS), to non-linear constrained problems (GPS Filter), from mixed-variable constrained programming (GMVP), to the new extension on the poll directions to a dense space in mesh adaptive direct search (MADS).

In the past three years, the author and his students had paid more attentions on optimal video transmission, quality of streaming video and media adaptation over IP and wireless. In order to find the optimal scheduling and transmitting strategies for streaming video such as MPEG-4 finer granularity scalability (FGS), MPEG-4 video objects (VOs), they had proposed some elaborate algorithms.

This paper is carried out in order to solve the media adaptation model in the framework of MPEG-21 Digital Items Adaptation (DIA), which was proposed and solved by the authors in related paper. This work is partly supported by the research fund for the State Ethnic Affairs Commission of China (05XN09).