arXiv:2302.11898v1 [math.OC] 23 Feb 2023

一种用于约束优化的类梯度下降法：算法与应用

Long Chen[a]、Kai-Uwe Bletzinger[b]、Nicolas R. Gauger[a] 和 Yinyu Ye[c]

[a]德国凯泽斯劳滕-兰道大学（RPTU）科学计算教席；[b]德国慕尼黑工业大学结构分析教席；[c]美国斯坦福大学管理科学与工程系及 ICME

文章历史
编译于2023年2月24日

摘要
我们提出了一种用于求解约束优化问题的一阶方法。该方法源自我们先前的工作 [28], ，即一种受奇异值分解启发的修正搜索方向法。在本研究中，我们简化其计算框架，提出一种“类梯度下降”方法（GDAM），即利用目标函数和约束梯度的负向归一化梯度的线性组合来计算搜索方向。我们给出了该方法全局收敛的基本理论保证。本研究重点探讨 GDAM 的算法及应用。我们提出了适用于该方法的计算算法，并借鉴了梯度下降法中的常见策略。我们通过形状优化和传感器网络定位这两项工程应用，展示了该方法的潜力。在实际应用中，GDAM 稳健且极具竞争力，能够求解所考虑的大规模、具有挑战性的优化问题。

关键词
负向归一化梯度；不等式约束优化；梯度下降；内点法；形状优化；传感器网络定位

美国数学学会分类
65K05；90C22；90C30；90C51；90C90

1. 引言

本文提出了一种用于求解不等式约束最优化问题的一阶方法。我们关注的问题为

$$\begin{aligned} &\text{minimize} \quad f(x), \\ &\text{subject to} \quad g_i(x) \leq 0,\ i = 1, ..., m, \end{aligned} \tag{COP}$$

L. Chen。电子邮箱：long.chen@scicomp.uni-kl.de
K.-U.Bletzinger。电子邮箱：kub@tum.de N. R. Gauger。电子邮箱：nicolas.gauger@scicomp.uni-kl.de Y. Ye。电子邮箱：yyye@stanford.edu

其中 f，g_1，……，g_m：$\mathbb{R}^n \to \mathbb{R}$ 均为二阶可微。我们针对带有单个不等式约束 COP，提出了一种类梯度下降法（GDAM） $g(x)$,

$$\begin{aligned} &x_{k+1} = x_k + \alpha_k \mathbf{s}_\zeta(x_k), \\ &\mathbf{s}_\zeta(x_k) = -\frac{\nabla f(x_k)}{|\nabla f(x_k)|} - \zeta \frac{\nabla g(x_k)}{|\nabla g(x_k)|}, \quad \zeta \in [0,1), \end{aligned} \tag{GDAM}$$

其中，α_k 表示步长，“$|\cdot|$”表示欧几里得范数，ζ 是一个参数。$\nabla f(x)$ 和 $\nabla g(x)$ 分别是目标函数和约束函数的梯度列向量。

给定众所周知的对数障碍函数

$$\Phi(x) = -\sum_{i=1}^{m} \log(-g_i(x)), \ i = 1, ..., m, \tag{1}$$

其中 log(·) 表示自然对数，我们将 (GDAM) 推广到带有 $\zeta \in [0, 1)$ 的多约束问题，得到

$$\mathbf{s}_\zeta(x_k) = \begin{cases} -\dfrac{\nabla f(x_k)}{|\nabla f(x_k)|} - \zeta \dfrac{\nabla \Phi(x_k)}{|\nabla \Phi(x_k)|}, & \text{if } \nabla \Phi(x_k) \neq 0; \\ -\nabla f(x_k), & \text{if } \nabla \Phi(x_k) = 0. \end{cases} \tag{2}$$

1.1. 主要贡献

理论、计算算法和应用是优化方法的三大支柱。正如标题所示，本文主要聚焦于 GDAM 的算法和应用。为使论述完整，我们还给出了该方法全局收敛性的基本理论保证。

我们总结了本工作的主要贡献：

1. 我们提出了（GDAM），该方法源自我们先前的工作 [28], ，其中两种方法的等价性并不显而易见。2. 我们采用连续时间动力系统方法，为该方法提供基本理论保证。3. 我们提出了两种基于（GDAM）的计算算法：一种基础实现和一种加速实现。4. 我们将所开发的算法应用于两个工程应用领域：基于节点的形状优化和传感器网络定位。这些应用展示了 GDAM 在解决大规模且困难的优化问题时的实用价值。

1.2. 论文结构

在第2节中，我们首先回顾相关研究，重点关注算法和应用方面。在第3节中，我们推导出适用于单不等式约束问题的方法。第4节通过示例直观地介绍该方法。我们在第5

节给出该方法的基本理论保证，并在第6节展示其向多重约束的推广。我们在第7 节介绍 GDAM 的计算算法，并展示初步数值实验。第8 节和9节介绍两个工程应用：形状优化和传感器网络定位。最后，我们在第10节讨论并总结全文。

2. 相关工作与应用

2.1. 梯度下降法

梯度下降法最初由柯西提出，是一种用于无约束优化的一阶方法，通过目标函数的负梯度进行变量更新 [64]。其典型形式为

$$x_{k+1} = x_k - \alpha_k \nabla f(x_k), \tag{3}$$

梯度下降的方向 是欧几里得范数下的最速下降方向。为说明这一点，我们在目标函数当前迭代点 x_k处进行一阶泰勒展开，

$$f(x_k + d) \approx f(x_k) + \langle \nabla f(x_k), d \rangle.$$

其中，“$\langle , \rangle$”表示欧几里得空间中的内积。最速下降方 d 向由以下优化问题求得

$$\begin{aligned} &\text{minimize} \quad \langle \nabla f(x_k), d \rangle \\ &\text{subject to} \quad |d| = 1. \end{aligned}$$

根据柯西-施瓦茨不等式，可得

$$d(x_k) = -\frac{\nabla f(x_k)}{|\nabla f(x_k)|}, \tag{4}$$

即负向归一化的目标函数梯度。比较（4）和（GDAM），

$$\mathbf{s}_\zeta(x_k) = -\frac{\nabla f(x_k)}{|\nabla f(x_k)|} - \zeta \frac{\nabla g(x_k)}{|\nabla g(x_k)|}, \quad \zeta \in [0, 1),$$

将目标函数梯度和约束梯度的负方向归一化后进行线性组合；本文考虑一种类梯度下降法，用于求解不等式约束优化问题。本文方法可以直观地解释如下：对于一个不等式约束问题，我们利用目标函数梯度下降方向来最小化 $f(x)$，并利用约束梯度下降方向来最小化 $g(x)$（从而通过远离约束边界来保持可行性）。由于梯度方向均已归一化为 1，参数 $0 \le \zeta < 1$ 可确保计算得到的方向是目标函数的下降方向。

2.2. 内点法

内点法（IPM）大多基于牛顿法，是约束优化问题中最具竞争力的方法之一。内点法的特点是存在连续参数化的近似解族，这些近似解渐近收敛于精确解 [40]。内点法广泛应用于各领域的凸优化和非凸优化。大量优秀研究致力于内点法。全面综述这类方法显然超出了本文的范围，不过我们建议感兴趣的读者参阅 [40][87][112],较近期的研究，见 [43]，以及许多其他优秀的优化著作。

本文的方法所得到的轨迹渐近收敛于某种特定内点法——对数障碍法的中心路径 [38]。接下来，我们将介绍它与本文方法之间的联系和区别。考虑如下形式的凸优化问题（COP），我们从使用对数障碍函数得到的近似无约束子问题开始 $\Phi(x)$,

$$\text{minimize} \quad f(x) + \eta\Phi(x), \tag{5}$$

其中，障碍参数 η 为正参数。当 $\eta \to 0$时，近似问题的解收敛于原问题的解。中心路径由满足充要条件 [24]的点集刻画：

$$0 = \nabla f(x^\star) + \eta\nabla\Phi(x^\star), \quad x^\star \in \Omega_-, \tag{6}$$

其中 $\Omega_- = \{x : \quad g_i(x) < 0, \ i = 1, \ldots\ldots, \ m\}$。文献中将条件（6）解释为修正后的 KKT 系统 [24][27][40]。障碍法通过以下方式为原问题求得一个近似解：1）逐步减小障碍参数 η；2）在每次迭代中，使用牛顿法求解由（6）定义的子问题（修正后的 KKT 系统）。因此，障碍参数 η可视为中心路径参数。

在本方法中，我们不对中心路径进行参数化，而是将目标函数的梯度和约束的梯度归一化。对于只有一个不等式约束的优化问题，我们提出归一化中心路径条件 如下：

$$\frac{\nabla f(x)}{|\nabla f(x)|} + \frac{\nabla g(x)}{|\nabla g(x)|} = 0. \tag{7}$$

条件（7）用于第5 节的分析。借助对数障碍函数（见第6节），将归一化中心路径条件 从单个不等式约束推广到多个不等式约束，得到：

$$\frac{\nabla f(x)}{|\nabla f(x)|} + \frac{\nabla \Phi(x)}{|\nabla \Phi(x)|} = 0. \tag{8}$$

与障碍法相比，路径参数 η 已消失。条件（8）刻画了中心路径，这与（6）不同；后者刻画的是中心路径上的一个点。

为了了解 GDAM 与内点法之间的联系，我们首先写出

梯度下降方向 $d_{\Phi,\ \eta}$ 对于障碍 η-子问题（5），

$$d_{\Phi,\eta} = -\nabla f(x) - \eta\Phi(x). \tag{9}$$

设计一阶方法的一种直接思路，是遵循二阶内点法的双循环结构：1）在外循环中迭代减小障碍参数 η，2）在内循环中通过梯度下降（5）（9）求解子问题。不幸的是，这种方法早已被认为在计算上不切实际，参见例如 [70, 第 469] 页的讨论。粗略地说，当 η 非常小时，$d_{\Phi,\ \eta}$ 往往会导致优化轨迹沿着约束的边界前进，并且受到严重的病态条件影响。

接下来，我们用 $|\nabla f(x_k)|$ 缩放 GDAM 搜索方向 s_ζ，

$$|\nabla f(x_k)|s_\zeta = -\nabla f(x_k) - \zeta\frac{|\nabla f(x_k)|}{|\nabla\Phi(x_k)|}\nabla\Phi(x_k). \tag{10}$$

比较 (10) 与 (9) 可知，（缩放后的）GDAM 提出了一种动态计算障碍参数的方式，且在每一步 k,

$$\eta(x_k) = \zeta\frac{|\nabla f(x_k)|}{|\nabla\Phi(x_k)|}. \tag{11}$$

GDAM 产生的优化轨迹沿着中心路径前进（处于相对于参数 ζ 的某个邻域内）——这是路径跟踪方法的典型行为。因此，我们将 GDAM 视为一种一阶内点法。作为本文的主要贡献之一，我们将证明，与依次对障碍子问题应用梯度下降不同，GDAM 是一种在计算上切实可行的优化方法。

2.3. 动力系统方法

许多研究使用动力系统方法来研究优化方法，因此我们的文献综述无法面面俱到。关于内点流与线性规划方法之间联系的大量研究，见 [49, 第 4] 章及其中的参考文献。从无约束优化的角度研究非线性耗散动力系统的工作见 [11]。 [102] 使用动力系统作为分析工具，研究了著名的Nesterov加速梯度法。近年来，动力系统已被用于研究解决机器学习应用中产生的问题的优化算法，例如见 [10][30][56][110]。在一些文献中，使用动力系统的优化方法被称为轨迹方法。这些方法以这样的方式构造优化路径，使得优化问题的一个或全部解先验地已知位于这些路径上[35]。通常，这些优化路径是一阶或二阶常微分方程的解轨迹。轨迹方法主要用于无约束优化，以寻找局部解 [13][22], 和全局解 [45][96]。然而，关于约束优化的研究非常有限，见 [3][21][108] 及其中的参考文献。在本研究中，我们从动力系统视角给出了当前方法的基本理论保证——证明该方法全局收敛到一阶驻点。

2.4. 可行方向法

可行方向法（MFD）可追溯至Zoutendijk于20世纪60年代开展的工作 [115] ，并在数十年间取得了丰硕的发展。由于最终得到满足一组不等式约束所表达的硬性规范的设计至关重要，MFD在工程界尤其受到欢迎[29]。MFD背后的基本思想是，迭代地从一个可行设计转向一个改进后的可行设计，从而找到局部解 [9]。在当前方法中，搜索方向不必是可行方向。该方法设计背后的思路是，在保持目标函数下降方向的同时，寻找一个趋近中心路径的搜索方向。由于这一重大差异，我们不将该方法归类为可行方向法。

在本文的方法中，我们计算使用归一化梯度的搜索方向。在 [99], 中，作者也提出了一种应用归一化梯度的可行方向法。他们的工作随后在 [33]和 [100]中得到延续和进一步发展。在这些研究中，采用了有效集策略。其共同思路是根据给定的输入准则，在选定的（有效）约束工作集上建立线性系统，并通过求解该线性系统获得可行下降搜索方向。在本文的方法中，我们采用基于障碍函数的公式来处理多个不等式约束。搜索方向由负的归一化目标函数梯度和障碍梯度的线性组合计算得到。

2.5. 形状优化

作为设计优化的一个子集，形状优化的特点是设计变量数量极大，甚至无限多；这些变量描述优化过程中不断变化的边界。 [48][98]介绍了形状优化。形状优化不同于设计优化中的另一个著名问题——拓扑优化 [14] （有时也称为均匀化方法[4]）。两者的主要区别在于，拓扑优化方法取消了形状优化中的光滑性和拓扑约束，因此产生了不同的优化公式。许多拓扑优化问题可以表述为（等价的）凸优化问题，而形状优化通常是非线性的，且往往是非凸的 [53]。这一区别在一定程度上解释了为何已有针对大规模[1] 拓扑优化问题的成功内点法实现 [55][58][61][73], ，但仅有少数研究提出了将内点法用作优化器的形状优化方法 [8][50]。在后者中，形状优化问题的规模仅属中等，因此未能充分发挥内点法的优势。非线性拓扑优化最成功的方法之一是 Svanberg 于 1987 [105]提出的移动渐近线法（MMA）。在每次迭代中，MMA都会生成并求解一个与原问题相关的凸近似问题。然而，对于形状优化，目前尚无文献讨论使用MMA求解大规模问题。

形状优化是多学科设计优化（MDO）中经常涉及的课题 [47]。为设计复杂的工程系统，多学科设计优化会考虑多个学科及其相互作用。对于形状优化，目前的研究致力于计算耦合学科的形状梯度，例如稳态耦合问题 [59][79] 和瞬态耦合问题 [63]。

[1]在本文中，我们将具有高维变量空间的问题称为大规模优化问题。

尽管耦合梯度的开发颇具挑战，但其优势在于能够准确计算导数并大幅降低计算成本，这对于大规模优化至关重要 [76]。在许多实际情况下，形状几何会用有限个参数进行重参数化。如果存在可复现性限制 [67]，或者初始几何变量不可微 [54]，通常就需要对形状进行重参数化。参数化良好的形状几何往往能够采用标准优化方法 [41][51][84] 或牛顿-克雷洛夫类方法[34]。另一方面，可实现的形状受到限制，并取决于所选的参数化方式。直接基于有限元网格 [68][114] 或水平集方法 [5][95],的高保真形状优化能够针对现实问题利用尽可能大的形状空间，但也会带来颇具挑战的优化问题 [37][71]。

手动推导和实现形状导数可能是一项繁琐且容易出错的工作。解决这一难题的一个重要方法是算法微分（AD） [44], ，它能够计算以计算机程序形式给出的函数的导数。在形状优化领域，算法微分已成功应用于 OpenFOAM[107], 、SU2[2], 和 FEniCS[36], 等开源求解器，使得形状设计优化能够应用于日益复杂的问题。另一项正在进行的研究是计算形状海森矩阵，即使对于中等规模的问题，这也是复杂的对象。近期有多项研究计算近似形状海森矩阵，并将基于牛顿法的方法用于设计优化 [92][93]。总体而言，迄今为止，大规模形状优化主要采用梯度下降类方法 [94]。在工程实践中，可能需要考虑大量约束。相关文献的缺乏促使我们开发了文献 [28]中的 MSDM。在本研究中，我们将其计算框架简化为一种类梯度下降法。一个重要成果是，实施工作量和计算成本均显著降低。这使形状优化有望应用于更广泛的领域。

2.6. 基于半定规划松弛的传感器网络定位

无线传感器网络（WSN）由成本低、能耗低且具备多种功能的传感器组成，这些传感器能够进行短距离通信。当大量传感器以无线方式互连并部署时，无线传感器网络为监测和控制物理环境提供了许多机会 [1]。无线传感器网络在环境、工业、城市、健康等领域有着广泛的应用，相关综述可参见 [57][88]。在许多应用中，了解传感器位置对于准确解读所收集的感知数据至关重要，参见例如 [26][65][90][104]。因此，基于成对距离估计传感器位置的传感器网络定位（SNL），被认为是支持无线传感器网络应用的关键使能技术之一。

传感器网络定位（SNL）是一类具有挑战性的优化问题，因为它是非凸问题，通常难以求得全局解。在各种求解方法中，基于半定规划（最初由 Biswas-Ye 提出 [18]）的凸松弛被视为全局定位最突出的途径之一 [31]。理论上，如果某个传感器网络定位问题是唯一可定位的 [97]，半定规划（SDP）松弛就能给出该问题的精确解。阻碍 SDP 松弛在大规模 SNL 问题中实际应用的挑战

主要在于计算方面[2]。基于二阶内点法（[7][15][86][106][111]）的成熟 SDP 求解器，能够以任意高的精度求解包含数百个传感器的 SNL 问题，但无法高效求解规模更大的问题。这促使人们开发先进的建模方法来缓解计算难题，例如参见 [19][60][109]。与此同时，过去二十年间，研究人员成功开发了若干可扩展的 SDP 求解器。例如，PENNON 软件包 [62] 和 SDPNAL+ [103] 基于增广拉格朗日法，而 SCS [82] 基于交替方向乘子法（ADMM）。文献 [74]全面综述了 SDP 求解器可扩展性的最新进展。本文采用 GDAM 求解 SNL 问题的 Biswas-Ye 半定规划松弛。我们对半定锥约束采用经典的对数障碍函数方法，因此，我们的方法具有求解半定规划问题的通用性。计算实验表明，在实际实现中，GDAM 在求解中等规模 SNL 问题时具有很强的竞争力，能够找到精度中等的解；同时还能够求解规模极大的问题（传感器数量超过 5000），而这些问题对于作为比较对象的求解器而言难以求解。

3. 推导单不等式约束优化的搜索方向

本节将针对单个不等式约束，展示我们先前工作中GDAM的一致推导[28] 。首先，我们回顾修正搜索方向法的基本思想，然后展示其推导。在本文余下部分，当上下文含义明确时，为简化记号，我们使用 ∇f 和 ∇g ，而不使用 $\nabla f(x)$ 和 $\nabla g(x)$。

在 MSDM 中，对每个迭代点 x，我们构造一个灵敏度矩阵，其中包含归一化的目标函数梯度和约束梯度，

$$\mathbf{m}(x) = \begin{bmatrix} \frac{\nabla f^T}{|\nabla f|} \\ \frac{\nabla g^T}{|\nabla g|} \end{bmatrix}. \tag{12}$$

目标函数和约束函数在 x 处的全微分为

$$\begin{aligned} df &= \nabla f^T dx, \\ dg &= \nabla g^T dx. \end{aligned} \tag{13}$$

从矩阵$\mathbf{m}(x)$ 所建立的输入输出系统视角可得

$$\begin{pmatrix} \frac{df}{|\nabla f|} \\ \frac{dg}{|\nabla g|} \end{pmatrix} = \mathbf{m}(x)dx. \tag{14}$$

对矩阵 $\mathbf{m}(x)$进行奇异值分解，可得

$$\mathbf{m}(x) = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T = \sum_{i=1}^{min(2,n)} \sigma_i \mathbf{u}_i \mathbf{v}_i^T. \tag{15}$$

[2]事实上，可扩展性是将半定规划成功应用于许多大型实际问题所面临的一项主要挑战。

因此，得到一组正交归一基$\mathbf{v}_i$，$i = 1,2,\mathbf{v}_i \in \mathbb{R}^n$。每个$\mathbf{v}_i$都可用作变量更新的基搜索方向，而$\mathbf{v}_1$和$\mathbf{v}_2$定义如下：

- $\mathbf{v}_1$：取 $\delta\mathbf{v}_1$ 作为变量更新，我们会使目标函数和约束函数发生变化 $[\frac{df}{|\nabla f|}, \frac{dg}{|\nabla g|}]^T = \sigma_1\delta\mathbf{u}_1$, 其中目标函数**下降**，而约束函数**上升**。- $\mathbf{v}_2$：取 $\delta\mathbf{v}_2$ 作为变量更新，我们会使目标函数和约束函数发生变化 $[\frac{df}{|\nabla f|}, \frac{dg}{|\nabla g|}]^T = \sigma_2\delta\mathbf{u}_2$, 其中目标函数**下降**，且约束函数**下降**。

值得一提的是，更新步长 $\delta\mathbf{v}_1$ 产生的结果与 [39], 中介绍的过滤器方法类似，该方法试图最小化所谓的双目标优化问题，其两个目标分别是最小化目标函数 f 和约束违反 $|g|$。

利用 $\mathbf{v}_1$ 和 $\mathbf{v}_2$, ，我们可以将归一化最速下降方向 $-\frac{\nabla f}{|\nabla f|}$ 改写为

$$-\frac{\nabla f}{|\nabla f|} = \cos\alpha_1\mathbf{v}_1 + \cos\alpha_2\mathbf{v}_2, \tag{16}$$

其中， α_1 是$\mathbf{v}_1$ 与$-\frac{\nabla f}{|\nabla f|}$之间的夹角， α_2 是$\mathbf{v}_2$ 与$-\frac{\nabla f}{|\nabla f|}$之间的夹角。修正搜索方向为

$$\mathbf{s}_c = \cos\alpha_1\mathbf{v}_1 + c\cdot\cos\alpha_2\mathbf{v}_2, \tag{17}$$

其中， $c \geq 1$ 的引入是为了增大$\mathbf{v}_2$ 对变量更新的贡献。下面，我们展示从式（GDAM）推导式（17）的过程。根据式（15），可得

$$\mathbf{m}\mathbf{m}^T = \mathbf{U}\mathbf{\Sigma}\mathbf{\Sigma}^T\mathbf{U}^T, \tag{18}$$

这是对称矩阵$\mathbf{m}\mathbf{m}^T$的对角化。我们有

$$\mathbf{m}\mathbf{m}^T = \begin{pmatrix} \frac{\nabla f^T}{|\nabla f|} \\ \frac{\nabla g^T}{|\nabla g|} \end{pmatrix} \begin{pmatrix} \frac{\nabla f}{|\nabla f|} & \frac{\nabla g}{|\nabla g|} \end{pmatrix} = \begin{pmatrix} 1 & \cos\theta \\ \cos\theta & 1 \end{pmatrix}, \tag{19}$$

其中， θ 是 ∇f 与 ∇g之间的夹角。$\mathbf{m}\mathbf{m}^T$ 的特征值为

$$\begin{aligned} \lambda_1 &= 1 - \cos\theta, \\ \lambda_2 &= 1 + \cos\theta. \end{aligned} \tag{20}$$

根据 MSDM 中$\mathbf{v}_1$ 和$\mathbf{v}_2$ 的定义，$\mathbf{m}\mathbf{m}^T$ 的特征向量可以分解为，

$$\begin{aligned} \mathbf{u}_1 &= \left(-\frac{\sqrt{2}}{2}, \quad \frac{\sqrt{2}}{2}\right)^T, \\ \mathbf{u}_2 &= \left(-\frac{\sqrt{2}}{2}, \quad -\frac{\sqrt{2}}{2}\right)^T. \end{aligned} \tag{21}$$

相应的奇异值为

$$\begin{aligned}\sigma_1 &= \sqrt{1-\cos\theta},\\ \sigma_2 &= \sqrt{1+\cos\theta}.\end{aligned} \tag{22}$$

根据奇异值分解的定义，

$$\begin{pmatrix}\frac{\nabla f^T}{|\nabla f|}\\ \frac{\nabla g^T}{|\nabla g|}\end{pmatrix} = \begin{pmatrix}-\frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2}\\ \frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2}\end{pmatrix}\begin{pmatrix}\sqrt{1-\cos\theta} & 0\\ 0 & \sqrt{1+\cos\theta}\end{pmatrix}\begin{pmatrix}\mathbf{v}_1^T\\ \mathbf{v}_2^T\end{pmatrix}. \tag{23}$$

因此，我们得到

$$\begin{aligned}\mathbf{v}_1 &= \frac{1}{\sqrt{2-2\cos\theta}}\left(-\frac{\nabla f}{|\nabla f|}+\frac{\nabla g}{|\nabla g|}\right),\\ \mathbf{v}_2 &= \frac{1}{\sqrt{2+2\cos\theta}}\left(-\frac{\nabla f}{|\nabla f|}-\frac{\nabla g}{|\nabla g|}\right),\end{aligned} \tag{24}$$

其中， θ 是目标函数梯度 ∇f 与约束函数梯度 ∇g 之间的夹角（详细内容见附录 A）。令 θ ，我们还有

$$\begin{aligned}\cos\alpha_1 &= -\langle\frac{\nabla f}{|\nabla f|},\mathbf{v}_1\rangle = \frac{\sqrt{1-\cos\theta}}{\sqrt{2}},\\ \cos\alpha_2 &= -\langle\frac{\nabla f}{|\nabla f|},\mathbf{v}_2\rangle = \frac{\sqrt{1+\cos\theta}}{\sqrt{2}}.\end{aligned} \tag{25}$$

将（24）、（25）代入（17），可得

$$\mathbf{s}_c = -\frac{\nabla f}{|\nabla f|}-\frac{(c-1)}{2}\left(\frac{\nabla f}{|\nabla f|}+\frac{\nabla g}{|\nabla g|}\right). \tag{26}$$

由于我们主要关注向量场$\mathbf{s}_c$的方向，因此可以将其改写为

$$\mathbf{s}_\zeta = -\frac{\nabla f}{|\nabla f|}-\zeta\frac{\nabla g}{|\nabla g|}, \tag{27}$$

令 $\zeta = \frac{c-1}{c+1}$。由于 $c\in[1,+\infty)$，我们有 $\zeta\in[0,1)$，从而得到当前的搜索方向（GDAM）。

4. 示例与直觉

本节通过两个示例展示 GDAM，并对其优化行为给出直观解释。

4.1. 固定 ζ 时该方法的行为

我们考虑一个二次约束非凸优化问题,

$$\begin{aligned} &\text{minimize} \quad f(x_1, x_2) = 1.5x_1^2 + x_2^2 - 2x_1x_2 + 2x_1^3 + 0.5x_1^4, \\ &\text{subject to} \quad g(x_1, x_2) = x_1^2 - x_2 - 2.2 \leq 0. \end{aligned} \tag{28}$$

由于非凸性，该问题存在多个局部解。图1显示，在所示的变量空间中，所考虑的问题有两个局部解:

1. $f(x)$ 的一个临界点（以星号表示）位于可行集内部。2. 一个 KKT 解（以圆点表示）位于约束的边界上。

可以观察到，所有优化轨迹都收敛到临界点（以星号表示）。此外，左侧两条轨迹与约束边界的交点是近似 KKT 点。前一种情况是梯度流方法的常见行为，而后一种情况则是本方法所独有的。

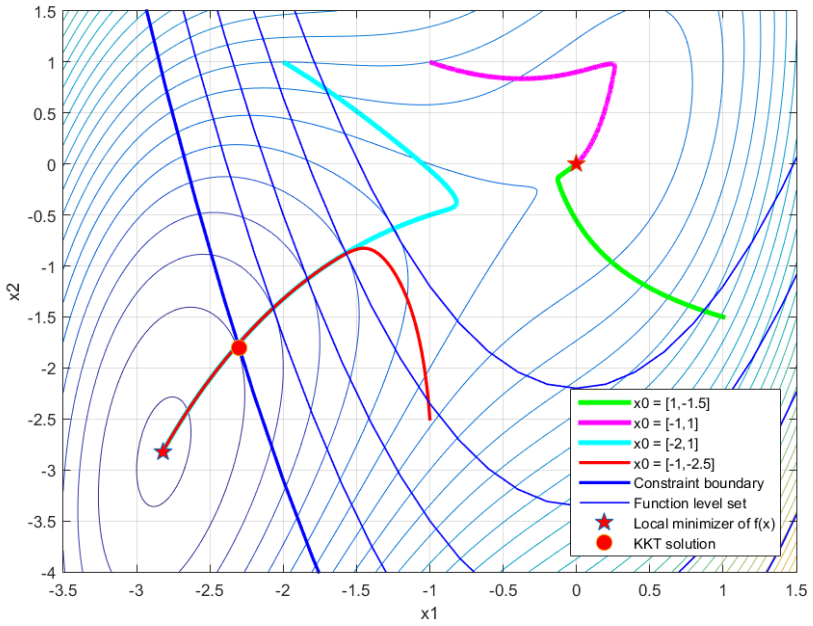


图 1：问题（28）的优化轨迹，其中 $\zeta = 0.95$ 并采用不同的初始化。根据初始化的不同，会找到该优化问题的不同解。

4.2. 方法随 ζ 的行为

我们通过解析求解一个二维优化问题，展示该方法在不同 ζ 下的行为。该优化问题如下：

$$\begin{aligned} &\text{minimize} \quad f(x_1, x_2) = \frac{1}{2}(x_1^2 + x_2^2), \\ &\text{subject to} \quad g(x_1, x_2) = -x_2 + 10 \leq 0. \end{aligned} \tag{29}$$

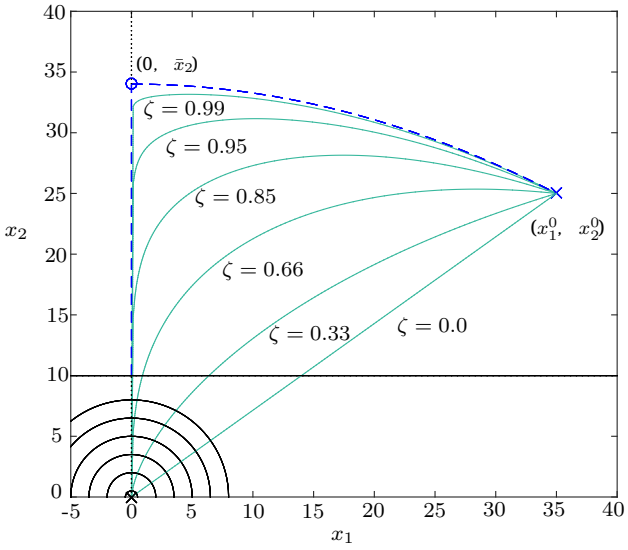


图 2：不同 29 下，二维线性约束优化问题（ζ）的优化轨迹。黑色圆圈表示目标函数等高线。黑色直线表示约束函数。虚线表示中心路径。当 $\zeta \to 1^-$ 时，基于当前搜索方向场（Γ^ζ 27）的一段轨迹收敛到中心路径。

当前搜索方向场 $\mathbf{s}_\zeta$ 表示为

$$\mathbf{s}_\zeta = \frac{-1}{\sqrt{x_1^2 + x_2^2}} \left\{ x_1, x_2 - \zeta\sqrt{x_1^2 + x_2^2} \right\}^T. \tag{30}$$

我们将初始化定义为 $(x_1^0,\ x_2^0)$。设 $\bar{x}_2 = \frac{1}{2}\left(x_2^0 + \sqrt{(x_1^0)^2 + (x_2^0)^2}\right)$，则当前搜索方向场 $\mathbf{s}_\zeta$ 的轨迹 Γ^ζ 为

$$x_2 + \sqrt{x_1^2 + x_2^2} = 2\bar{x}_2 \left| \frac{x_1}{x_1^0} \right|^{1-\zeta}. \tag{Γ^ζ}$$

令 $(x_{1\,\zeta},\ x_{2,\ \zeta})$ 是轨迹 Γ^ζ 上具有最大 x_2 分量的一个点，那么

我们有

$$(x_{2,\zeta})^{\zeta} = \frac{2\zeta}{1+\zeta}\frac{\bar{x}_2}{|x_1^0|^{1-\zeta}}\left(\frac{\sqrt{1-\zeta^2}}{\zeta}\right)^{1-\zeta}, \tag{31}$$

and

$$|x_{1,\zeta}| = \frac{\bar{x}_2}{\zeta}\sqrt{1-\zeta^2}. \tag{32}$$

令 $\zeta \to 1^-$，那么，$x_{1,\zeta} \to 0$, $x_{2,\zeta} \to \bar{x}_2$，轨迹 Γ^{ζ} 将收敛到曲线 Γ，该曲线由抛物线

$$x_1^2 = 4\bar{x}_2^2 - 4x_2\bar{x}_2, \quad x_1 \in (0, x_1^0)(\quad \text{or}(x_1^0, 0),) \tag{33}$$

以及区间 $(0,\ \bar{x}_2)$（位于 x_2 轴上），如图 2 所示（完整推导详见附录 A）。

现在我们给出轨迹 (Γ^{ζ}) 与约束边界的交点 $x_{\zeta}^{\sharp}$ 的估计，

$$\begin{cases} x_{\zeta,2}^{\sharp} + \sqrt{(x_{\zeta,1}^{\sharp})^2 + (x_{\zeta,2}^{\sharp})^2} = 2\bar{x}_2 \left|\dfrac{x_{\zeta,1}^{\sharp}}{x_1^0}\right|^{1-\zeta}, \\ x_{\zeta,2}^{\sharp} = 10; \end{cases} \tag{34}$$

为便于说明，我们首先假设 $x_1^0 > 0$。根据

$$\frac{dx_1}{dt} = \frac{-x_1}{\sqrt{x_1^2 + x_2^2}},$$

当 $x_1 > 0$ 时，优化轨迹单调递减，因此 $x_{\zeta,\ 1}^{\sharp} < x_{1,\ \zeta}$。此外，$\frac{dx_1}{dt}|_{x_1=0} = 0$ 保证对任意 t，$x_1(t) \geq 0$ 成立。综合这两个论据，我们得到交点 $x_{\zeta}^{\sharp}$ 的估计，

$$\begin{cases} 0 \leq x_{\zeta,1}^{\sharp} \leq \dfrac{\bar{x}_2}{\zeta}\sqrt{1-\zeta^2}, \\ x_{\zeta,2}^{\sharp} = 10. \end{cases} \tag{35}$$

显然，$x_1^0 < 0$ 也允许进行类似的研究。因此我们得到

$$|x_{\zeta}^{\sharp} - x^{\star}| \leq \frac{\bar{x}_2}{\zeta}\sqrt{1-\zeta^2}, \tag{36}$$

其中 $x^{\star} = (0,\ 10)$ 是 KKT 解。需要指出的是，误差界（36）是针对问题（29）专门推导的，并且是基于一个特殊点（$x_{1,\ \zeta}$，$x_{2,\ \zeta}$）得出的非常粗略的估计。在下一节中，我们将给出关于此类误差界的一般结果，以及其他基本理论保证。

5. 全局收敛

本节将给出该方法在全局收敛方面的基本理论保证。我们将首先分析单约束优化问题（SCOP），然后可以通过对数障碍函数（见下一节）轻松地将其结果推广到多重约束情形。

现在，我们考虑

$$\begin{aligned} &\text{minimize} \quad f(x), \\ &\text{subject to} \quad g(x) \leq 0. \end{aligned} \tag{SCOP}$$

回顾（GDAM）的迭代更新公式，

$$\begin{aligned} &x_{k+1} = x_k + \alpha_k \mathbf{s}_\zeta(x_k), \\ &\mathbf{s}_\zeta(x_k) = -\frac{\nabla f(x_k)}{|\nabla f(x_k)|} - \zeta \frac{\nabla g(x_k)}{|\nabla g(x_k)|}, \quad \zeta \in [0, 1). \end{aligned} \tag{GDAM}$$

我们利用以下动力系统研究该方法的收敛性，

$$\frac{dx}{dt} = -\frac{\nabla f(x)}{|\nabla f(x)|} - \zeta \frac{\nabla g(x)}{|\nabla g(x)|}, \quad \zeta \in [0, 1). \tag{DS}$$

除了理论上的收敛保证外，我们还报告了该方法连续时间轨迹的全局行为结果，这些结果有助于实际算法的设计。

5.1. 假设与基本结果

对于系统（DS）的分析，我们作出以下假设：

（A1）目标函数 $f(x)$ 满足强制性条件，

$$\lim_{x \to \infty} f(x) = +\infty;$$

（A2） $\nabla f(x) \neq 0$ 在可行集 $\Omega = \{x$ 中： $g(x) \leq 0\}$；（A3） $\nabla g(x) \neq 0$ 在可行集 Ω 中；（A4） f 和 g 是二阶连续可微函数。

如果函数 f（x）是强制性的且连续的，那么它存在一个全局极小点。引入假设（A2）和（A3）是由于我们的连续时间分析需要系统（DS）定义良好。我们将在小节5.2中讨论不包含（A2）的情形。在实际应用中，（A3）不一定总是满足。为了避开约束函数的临界点，我们建议使用梯度下降搜索方向（另见第6节）。

令 $x(t; \zeta, x_0)$ 为该系统的解：

$$\begin{cases} \dfrac{dx}{dt} = \mathbf{s}_\zeta(x), \\ x|_{t=0} = x_0. \end{cases} \tag{37}$$

其中， x_0 为可行集中的初始设计，$T_{\zeta,\ x_0}$ 为 $x(t;\ \zeta,\ x_0)$ 在集合 $E \subset \mathbb{R}^n$ 中的最大存在区间，

$$E := \{x : |\nabla f(x)| \neq 0, |\nabla g(x)| \neq 0, x \in \mathbb{R}^n\}.$$

有关存在最大区间的形式化定义，请参见 [85] 。下面，我们展示轨迹的一些基本性质 $x(t;\ \zeta,\ x_0)$。

5.1.1. $f(x(t,\zeta,x_0))$ 和 $g(x(t,\zeta,x_0))$的时间导数

首先，我们考察目标函数 $f(x)$ 和约束函数 $g(x)$ 沿解轨迹 $x(t,\ \zeta,\ x_0)$ 的变化。这通过计算$f(x(t,\ \zeta,\ x_0))$ 和 $g(x(t,\ \zeta,\ x_0))$ 的时间导数来实现，

$$\begin{aligned} &\frac{d}{dt}\left(f(x(t;\zeta,x_0))\right) = \frac{df(x)}{dx}\cdot\frac{dx(t;\zeta,x_0)}{dt} \\ &= \nabla f(x)^T\left(-\frac{\nabla f(x)}{|\nabla f(x)|} - \zeta\frac{\nabla g(x)}{|\nabla g(x)|}\right) \\ &= -|\nabla f(x)|\left(\frac{\nabla f(x)^T\nabla f(x)}{|\nabla f(x)||\nabla f(x)|} + \zeta\frac{\nabla f(x)^T\nabla g(x)}{|\nabla f(x)||\nabla g(x)|}\right) \\ &= -|\nabla f(x)|\left(1+\zeta\cos\theta(x)\right), \end{aligned} \tag{38}$$

类似地，

$$\frac{d}{dt}\left(g(x(t;\zeta,x_0))\right) = -|\nabla g(x)|(\cos\theta(x)+\zeta), \tag{39}$$

其中

$$\cos\theta(x) = \frac{\nabla f(x)^T\nabla g(x)}{|\nabla f(x)||\nabla g(x)|}. \tag{40}$$

显然，在假设 (38)、(39) 和 (40) 下，$(A1)-(A4)$ 且 $x \in \in$ 可行集。在下面，我们基于上述三个方程推导基本理论结果。

5.1.2. 解 $x(t;\ \zeta, x_0)$的有界性

首先，由（38）容易看出， $f(x)$沿着 $x(t;\ \zeta, x_0)$单调递减，其中$\zeta \in [0, 1$固定。此外，由 $f(x)$的 $coercive$ 假设（1），可知目标函数存在一个全局下界。我们可以证明，解轨迹$x(t;\ \zeta, x_0)$始终处于一个有界集中，该有界集取决于初始化 x_0。结果形式化表述如下。

首先，我们定义与可行初始值 x_0 相关的有界集 。

定义 5.1 （有界集 $B_{f(x_0)}$）。 给定一个可行初始值 x_0,

$$x_0 \in \Omega = \{x : g(x) \leq 0\}.$$

有界集 $B_{f(x_0)}$ 定义为

$$B_{f(x_0)} := \{x : f(x) \leq f(x_0)\}. \tag{41}$$

引理 5.2 (有界性)。 假设假设 *(A1) - (A4)* 成立，并令$\zeta \in [0, 1)$， $x_0 \in \Omega$。则轨迹 $x(t;\ \zeta,\ x_0)$ 始终保持在与可行初始值 x_0 相关的有界集$B_{f(x_0)}$ 内。

证明。 基于目标函数 $f(x)$ 沿轨迹$x(t;\ \zeta,\ x_0)$ 的变形（如式 38 所示），并且有 $\zeta \in [0, 1)$, $\cos\ \theta(x)\ \in [-1, 1]$, 可得

$$1 + \zeta \cos\theta(x) > 0. \tag{42}$$

因此,

$$\frac{df(x(t;\zeta, x_0))}{dt} \leq 0. \tag{43}$$

由此，证明完毕。 □

注 1。 引理 5.2 是一个基本结果，它表明在我们的假设下，式（38）的积分始终有界，即

$$\int_0^{T_{\zeta,x_0}} |\nabla f(x(t;\zeta, x_0))|(1 + \zeta \cos\theta(x))dt = f(x_0) - f(x(T_{\zeta,x_0};\zeta, x_0)) < \infty. \tag{44}$$

5.1.3. 向量场$\mathbf{s}_\zeta$的 *Lipschitz* 连续性

可行集 Ω 与有界集 $B_{f(x_0)}$ 的交集定义了有界可行集。

定义 5.3 （有界可行集 $\Omega_{f(x_0)}$）. 给定一个可行初始值 x_0,

$$x_0 \in \Omega = \{x : g(x) \leq 0\}.$$

有界可行集 $\Omega_{f(x_0)}$ 定义为

$$\Omega_{f(x_0)} := \{x : f(x) \leq f(x_0), x \in \Omega\}. \tag{45}$$

在有界可行集 $\Omega_{f(x_0)}$中，我们证明了当前向量场$\mathbf{s}_\zeta$的 Lipschitz 连续性。

引理 5.4. 假设条件 *(A1)-(A4)* 成立，且有界可行集 $\Omega_{f(x_0)}$ 非空。那么，向量场 $\mathbf{s}_\zeta(x)$ $\zeta \in [0, 1]$ 在 $x \in \Omega_{f(x_0)}$上是 *Lipschitz* 连续的，即:

$$|\mathbf{s}_\zeta(x) - \mathbf{s}_\zeta(y)| \leq L|x - y|, \tag{46}$$

其中 L 是正常数。

证明。 根据假设（A4）， f 和 g 的二阶导数在可行集 $\Omega_{f(x_0)}$上有界。因此， ∇f 和 ∇g 在可行集 Ω_{x_0}上是 Lipschitz 连续的。根据假设（A2）和（A3），我们有

$$|\nabla f(x)| \geq a, \quad |\nabla g(x)| \geq b, \quad \forall x \in \Omega_{x_0},$$

对于某些正数 a, b。因此， $\frac{\nabla f(x)}{|\nabla f(x)|}$ 和 $\frac{\nabla g(x)}{|\nabla g(x)|}$ 是 Lipschitz 连续的。因此， $s\zeta(x) = -\frac{\nabla f(x)}{|\nabla f(x)|} - \zeta \frac{\nabla g}{|\nabla g(x)|}(x)|$ 在可行集 $\Omega_{f(x_0}$上是 Lipschitz 连续的。 □

5.1.4. 中心路径邻域的余弦度量

在不等式约束优化的研究中，中心路径被视为“一个基本的数学对象” [12]。例如，在内点法的背景下，其总曲率被用于研究线性规划是否存在强多项式算法 [6]。方程（40）启发我们为中心路径及其邻域提出一种余弦度量。

回顾归一化中心路径条件（7），

$$\frac{\nabla f(x)}{|\nabla f(x)|} + \frac{\nabla g(x)}{|\nabla g(x)|} = 0,$$

该条件可表述为 $\nabla f(x)$ 与 $\nabla g(x)$ 之间的角度关系

$$\cos\theta(x) = \frac{\langle \nabla f(x), \nabla g(x) \rangle}{|\nabla f(x)||\nabla g(x)|} = -1. \tag{47}$$

在假设 (A1)-(A4) 下，$\cos\theta(x)$ 是 x 在 $\Omega_{f(x_0)}$ 上的连续可微函数。此外，我们注意到，当 x 位于中心路径上时，$\cos\theta(x)$ 达到其自身的最小值。结合这两个论点，我们可以方便地利用 $\cos\theta(x)$ 的等值集来定义中心路径的一个邻域。

定义 5.5 （μ–邻域）。 中心路径的 μ–邻域定义为

$$\Theta_\mu = \{x : \cos\theta(x) < -\mu, x \in \Omega_{f(x_0)}\}, \quad \mu \in [0, 1].$$

直观地说， Θ_μ 是中心路径周围的一个锥形邻域。显然，随着 $\mu \to 1^-.$， Θ_μ 会收缩至中心路径。根据 μ–邻域的定义，我们可以证明优化轨迹相对于约束函数 $g(x)$ 呈现以下行为。

引理 5.6. 假设 *(A1)-(A4)* 成立， $\zeta \in [0, 1)$ 为固定值，且 $x_0 \in$ 可行集，则约束函数 $g(x)$ 沿轨迹 $x(t; \zeta, x_0)$ 在邻域 Θ_ζ之外递减，并在 Θ_ζ中递增，其中 Θ_ζ定义为

$$\Theta_\zeta = \{x : \cos\theta(x) < -\zeta, x \in \Omega_{f(x_0)}\}. \tag{48}$$

证明. 根据式（ g 沿轨迹$x(t;\zeta,x_0)$ 所示的约束函数变形39),

$$\frac{d}{dt}g(x(t;\zeta,x_0)) = -|\nabla g|(\zeta+\cos\theta(x)),$$

我们可以直接得到一个证明。 □

注 2. 引理5.6 表明，当迭代点$x(t_k) \in \Theta_\zeta$ 时，优化进展得更快，反之则不然。这一结果有助于设计实用算法，通过开发相应的机制/启发式方法来调整步长和/或参数ζ ，从而实现更快的收敛。

5.2. f的x全局收敛至临界点

我们证明，只要 $\zeta \in [0,1)$，当前优化轨迹总是收敛到目标函数的一个临界点。

定理 5.7. 假设条件 *(A1)*、*(A3)*、*(A4)* 成立，且在整个 $\mathbb{R}^n$ 中 $|\nabla g| \neq 0$ 。根据 $\zeta \in [0,1)$，轨迹 $x(t;\ \zeta、\ x_0)$ 收敛到目标函数的临界点的一个连通子集。特别地，如果目标函数的临界点是孤立的，那么

$$\lim_{t\to T^-_{\zeta,x_0}} x(t;\zeta,x_0) = x_c, \qquad \forall\zeta\in[0,1), \tag{49}$$

其中 x_c 是 f 的临界点。

证明。 首先请注意，当前流可能是有限时间收敛的[3]。第 4.2 节中的解析示例表明，当前系统可以有限时间收敛：1）解析轨迹具有有限长度；2）根据当前系统的定义，流的速度具有最小范数 $1-\zeta$ 。为克服这一困难，我们引入一个经过时间重参数化的Y– 系统:

$$\begin{cases} \dfrac{dy}{d\tau} = |\nabla f(y)|\mathbf{s}_\zeta(y), \\ y|_{\tau=0} = x_0. \end{cases} \tag{50}$$

系统（50）在子集 $E\subset\mathbb{R}^n$ 中与（37）具有相同的轨道。在我们的假设下，Y 系统是一致 Lipschitz 连续的，并且关于 τ 连续。 根据皮卡存在定理，它有唯一解 y（τ; ζ, x_0），其存在区间无限长。

对于 $\zeta\in[0,1)$，考虑沿 $y(\tau;\ \zeta,\ x_0)$ 的积分，

$$f(y(T;\zeta,x_0)) = f(x_0) - \int_0^T |\nabla f|^2(1+\zeta\cos\theta)(y(\tau;\zeta,x_0))d\tau.$$

[3]如果 $\zeta=0$，当前系统就化为无约束最小化的归一化梯度流；通过非平凡的非光滑稳定性分析，已证明该流有限时间收敛 [32,定理 8]。

引理5.2 保证

$$\int_0^T |\nabla f|^2(1+\zeta\cos\theta)(y(\tau;\zeta,x_0))d\tau = f(x_0) - f(y(T;\zeta,x_0)) \le M \tag{51}$$

其中 M 是一个与 T 无关的正数。因此

$$\int_0^\infty |\nabla f|^2(y(\tau;\zeta,x_0))d\tau \le \frac{M}{1-\zeta} < +\infty. \tag{52}$$

注意到

$$\begin{aligned}
\frac{d}{d\tau}|\nabla f(y(\tau;\zeta,x_0))| &= \frac{d}{d\tau}\left(\sum_{j=1}^n \left(\frac{\partial f}{\partial y_j}\right)^2\right)^{\frac{1}{2}} \\
&= \frac{1}{2}\left(\sum_{j=1}^n \left(\frac{\partial f}{\partial y_j}\right)^2\right)^{-\frac{1}{2}} \sum_{j=1}^n \frac{d}{d\tau}\left(\frac{\partial f}{\partial y_j}\right)^2 \\
&= \frac{1}{|\nabla f|}\sum_{j=1}^n \frac{\partial f}{\partial y_j}\sum_{k=1}^n \frac{\partial}{\partial y_k}\left(\frac{\partial f}{\partial y_j}\right)\frac{dy_k}{d\tau} \\
&= \frac{1}{|\nabla f|}\sum_{j=1,k=1}^n \frac{\partial f}{\partial y_j}\frac{\partial^2 f}{\partial y_j y_k}\frac{dy_k}{d\tau} \\
&= -\frac{1}{|\nabla f|}\sum_{j=1,k=1}^n \frac{\partial f}{\partial y_j}\frac{\partial^2 f}{\partial y_j \partial y_k}|\nabla f|\left(\frac{1}{|\nabla f|}\frac{\partial f}{\partial y_k} + \frac{\zeta}{|\nabla g|}\frac{\partial g}{\partial y_k}\right).
\end{aligned}$$

注意到 $\zeta \in [0,1)$，我们有

$$\left|\frac{d}{d\tau}|\nabla f(y(\tau;\zeta,x_0))|\right| \le \sum_{k=1,j=1}^n \left|\frac{\partial f}{\partial y_j}\right|\left|\frac{\partial^2 f}{\partial y_k \partial y_j}\right|\left(\frac{1}{|\nabla f|}\left|\frac{\partial f}{\partial y_k}\right| + \frac{1}{|\nabla g|}\left|\frac{\partial g}{\partial y_k}\right|\right).$$

由柯西-施瓦茨不等式可得

$$\left|\frac{d}{d\tau}|\nabla f(y(\tau;\zeta,x_0))|\right| \le |\nabla f|\sqrt{\sum_{k=1,j=1}^n \left|\frac{\partial^2 f}{\partial y_k \partial y_j}\right|^2}\sqrt{\sum_{k=1}^n \left(\frac{1}{|\nabla f|}\left|\frac{\partial f}{\partial y_k}\right| + \frac{1}{|\nabla g|}\left|\frac{\partial g}{\partial y_k}\right|\right)^2}.$$

此外，我们有

$$\begin{aligned}
\sum_{k=1}^n \left(\frac{1}{|\nabla f|}\left|\frac{\partial f}{\partial y_k}\right| + \frac{1}{|\nabla g|}\left|\frac{\partial g}{\partial y_k}\right|\right)^2 &= \sum_{k=1}^n \left(\frac{1}{|\nabla f|^2}\left|\frac{\partial f}{\partial y_k}\right|^2 + 2\frac{1}{|\nabla f||\nabla g|}\left|\frac{\partial f}{\partial y_k}\frac{\partial g}{\partial y_k}\right| + \frac{1}{|\nabla g|^2}\left|\frac{\partial g}{\partial y_k}\right|^2\right) \\
&\le \sum_{k=1}^n 2\left(\frac{1}{|\nabla f|^2}\left|\frac{\partial f}{\partial y_k}\right|^2 + \frac{1}{|\nabla g|^2}\left|\frac{\partial g}{\partial y_k}\right|^2\right) = 4.
\end{aligned}$$

因此，

$$\left|\frac{d}{d\tau}|\nabla f(y(\tau;\zeta,x_0))|\right| \le 2|\nabla f|\sqrt{\sum_{k=1,j=1}^{n}\left|\frac{\partial^2 f}{\partial y_k \partial y_j}\right|^2}. \tag{53}$$

根据假设（A4）和引理5.2，上述方程右端有界。因此存在常数 l 使得

$$\left|\frac{d}{d\tau}|\nabla f(y(\tau;\zeta,x_0))|\right| \le l,\ \forall\tau.$$

因此，$|\nabla f(y(\tau;\zeta,x_0))|$满足 Lipschitz 连续性：

$$\big||\nabla f(y(\tau';\zeta,x_0))| - |\nabla f(y(\tau'';\zeta,x_0))|\big| \le l|\tau'-\tau''|, \qquad \forall\tau',\tau''. \tag{54}$$

现在，我们断言

$$\lim_{\tau\to+\infty}|\nabla f(y(\tau;\zeta,x_0))| = 0. \tag{55}$$

否则，存在一个序列 $\tau_j \to +\infty$ 和一个正常数 b，使得

$$|\nabla f(y(\tau_j;\zeta,x_0))| \ge b > 0.$$

取 $\delta = \frac{b}{2l}$，则 –

$$\begin{aligned}|\nabla f(y(\tau;\zeta,x_0))| &\ge |\nabla f(y(\tau_j;\zeta,x_0))|\\ &\quad - ||\nabla f(y(\tau_j;\zeta,x_0))| - |\nabla f(y(\tau;\zeta,x_0))||\\ &\ge |\nabla f(y(\tau_j;\zeta,x_0))| - l|\tau_j-\tau| \ge b - \delta l = \frac{b}{2}\end{aligned}$$

对于任意 $|\tau_j-\tau| \le \delta.$，因此，

$$\begin{aligned}\int_0^\infty |\nabla f(y(\tau;\zeta,x_0))|^2 d\tau &\ge \sum_{j=1}^{\infty}\int_{\tau_j-\delta}^{\tau_j+\delta}|\nabla f(y(\tau;\zeta,x_0))|^2 d\tau\\ &\ge \sum_{j=1}^{\infty}\int_{\tau_j-\delta}^{\tau_j+\delta}\frac{b^2}{4}d\tau = \sum_{j=1}^{\infty}\frac{b^3}{4l} = +\infty.\end{aligned}$$

这与（52）矛盾。因此

$$\lim_{\tau\to+\infty}|\nabla f(y(\tau;\zeta,x_0))| = 0. \tag{56}$$

这意味着 $y(\tau,\ \zeta,\ x_0)$趋近于临界点的连通子集。孤立条件确保

$$\lim_{\tau\to+\infty} y(\tau;\zeta,x_0) = x_c, \tag{57}$$

对于目标函数的某个临界点 x_c。请回忆，系统 (50) 和 (37) 在 E中轨道等价，因此

$$\lim_{t \to T^-_{\zeta,x_0}} x(t;\zeta,x_0) = x_c, \qquad \forall \zeta \in [0,1). \tag{58}$$

□

注释 3. 系统（37）的全局行为类似于梯度流。显然，当 $\zeta = 0$ 时，系统退化为目标函数的归一化梯度流。具体而言，当收敛到同一个临界点时，根据微分方程的解对参数的连续依赖性，当前轨迹与 $\zeta \in (0,\ 1)$ 对应的梯度下降轨迹同伦。

注释 4. 解析轨迹可能在没有条件（A3）的情况下找到约束 g 的临界点。在实际实现中，我们建议使用梯度下降$-\nabla f(x)$ 来离开这些临界点。

注释 5. 我们注意到，定理 5.7 中不要求假设（A2）。如果可行集 Ω 中存在 x_c 的某个临界点 $f(x)$，那么优化轨迹可能收敛到该点（另请比较引理 5.9，其中分析中包含假设（A2））。在这种情况下，显然可以找到所考虑约束优化问题的一阶解。图 1 展示了一个二维问题的示例。

5.3. 在约束边界处全局收敛至 ***KKT*** 解

在上一小节中，我们已经证明，连续优化轨迹收敛至目标函数 x_c 的一个临界点 $f(x)$，无论 x_c 是否位于可行集 Ω 中。对于 $x_c \in \Omega$ 的情况，这意味着我们能够利用当前优化轨迹求得一个（一阶）最优解。剩下的问题是，当前方法能否在约束边界处找到一个（近似最优）解，也就是第 4.1 节中介绍的第二种情况。下文将对此给出肯定的回答。为了严谨地阐述，我们首先定义“找到”一词的含义，并据此建立解到 KKT 点的误差界。

5.3.1. 在优化轨迹与约束边界的交点处找到解

如第二个预备示例的图 2 所示，对于不同的 $x(t;\ \zeta,\ x_0)$，其 ζ取值不同的优化轨迹会收敛至目标函数的临界点，同时在某个时刻 t与约束边界相交。选择的 ζ越大，交点就越接近 KKT 解。我们将这些交点视为当前方法针对单个约束问题所找到的解。形式化定义如下。

定义 5.8。 (解点 $x^\sharp_\zeta$) 令 $\zeta \in [0,1)$ 和 $x_0 \in \Omega$，并假设优化轨迹 $x(t;\ \zeta, x_0)$ 在点 $x^\sharp_\zeta = x(t^\sharp;\ \zeta, x_0)$ 与约束边界相交，其中 $t^\sharp_\zeta$ 是轨迹到达约束所需的最短时间

边界。我们称交点 $x_\zeta^\sharp$ 为通过系统（*DS*）为问题（SCOP）得到的解点。

$$\begin{cases} g(x(t^\sharp;\zeta,x_0))=0, \\ g(x(t;\zeta,x_0))<0, \quad t<t^\sharp. \end{cases} \tag{59}$$

接下来，我们论证，在假设 $(A1)-(A4)$ 成立时，优化轨迹$x(t;\ \zeta,\ x_0)$，其中 $\zeta\in[0,1)$ 且 $x_0\in\Omega$，总会与约束边界相交。尽管根据定理 5.7很容易得出这一结论，为清晰起见，我们还是给出一个独立而简明的证明。

引理 5.9。 假设 *(A1) - (A4)* 成立，且令 $\zeta\in[0,1)$，$x_0\in\Omega$。那么，轨迹 $x(t;\ \zeta,x_0)$ 必定会离开可行集。因此，必定存在一个解点$x_\zeta^\sharp$ 。

证明。 我们用反证法证明该引理。假设 $x(t;\ \zeta,\ x_0)$ 对任意 $t<T_{\zeta,x_0}$始终处于可行集 Ω 中，那么向量场$\mathbf{s}_\zeta$ 在轨迹 C^1 的邻域内保持连续，该轨迹为 $x(t;\ \zeta,\ x_0)|_{[0,T_{\zeta,x_0})}$，因为 Ω 中不存在 $f(x)$ 和 $g(x)$ 的临界点。引理 5.2 保证

$$|\nabla f(x(t;\zeta,x_0))|\geq A, |f(x(t;\zeta,x_0)|\leq B, \quad \forall t<T_{\zeta,x_0}, \tag{60}$$

其中 A 和 B为正数。另一方面，$\mathbf{s}_\zeta(x(t))$ 在 $x(t)$ 上一致 Lipschitz 连续，这是由引理5.4得出的。皮卡存在定理意味着$T_{\zeta,\ x_0}=+\infty$. (38) 的积分表明（另见注 1）

$$\int_0^{T_{\zeta,x_0}}|\nabla f(x(t;\zeta,x_0))|(1+\zeta\cos\theta)dt=f(x_0)-f(x(T_{\zeta,x_0};\zeta,x_0))<\infty.$$

对于任意 $\zeta\in[0,1)$，$1+\zeta\cos\ \theta>0$ 成立，因此

$$\int_0^\infty|\nabla f(x(t;\zeta,x_0))|dt<\infty. \tag{61}$$

(61) 与 (60) 矛盾。证明完毕。 □

5.3.2. 相对于 ζ 的解时间 $t^\sharp$ 上界

我们估计解时间 $t^\sharp$，即轨迹 $x(t;\ \zeta,\ x_0)$ 到达约束边界所需的时间。 $t^\sharp$ 的上界如下所示。

引理 5.10。 假设假设 *(A1)- (A4)* 成立，且令 $\zeta\in[0,1)$、$x_0\in$可行集，则优化轨迹 $x(t;\ \zeta,x_0)$ 到达可行集边界的时间至多为 $\frac{C}{1-\zeta}$，其中 C 与 ζ 无关。

证明。 由引理 5.9，轨迹 $x(t;\ \zeta,x_0)$ 必须离开可行集。设$T_{\zeta\ ,x_0}^\sharp$ 为该轨迹首次到达可行集边界的时间，

则（60）对 $0 < t < T^{\sharp}_{\zeta,x_0}$ 成立。由（38），

$$\int_0^{T^{\sharp}_{\zeta,x_0}} |\nabla f|(1+\zeta\cos\theta)dt = f(x_0) - f(x(T^{\sharp}_{\zeta,x_0};\zeta,x_0)) \le 2B. \tag{62}$$

因此

$$T^{\sharp}_{\zeta,x_0} A(1-\zeta) \le 2B, \tag{63}$$

这意味着

$$T^{\sharp}_{\zeta,x_0} \le \frac{2B}{A(1-\zeta)}, \tag{64}$$

从而有 $T^{\sharp}_{\zeta,x_0} \le \frac{C}{1-\zeta}$ 成立。这就完成了证明。 □

引理 5.10 意味着系统（DS）在 $x^{\sharp}_{\zeta}$ 中$t^{\sharp} \le O(\frac{1}{1-\zeta})$找到一个解。

5.3.3. 相对于 ζ 的解误差上界

最后，我们将给出一个解（交点） $x^{\sharp}_{\zeta}$ 相对于 KKT 解 $x^{\star}$的误差度量。问题（SCOP）的 KKT 条件写为

$$\nabla f(x^{\star}) + \lambda^{\star}\nabla g(x^{\star}) = 0, \tag{65a}$$

$$\lambda^{\star} = \frac{|\nabla f(x^{\star})|}{|\nabla g(x^{\star})|} \ge 0, \tag{65b}$$

$$g(x^{\star}) = 0. \tag{65c}$$

显然，条件（65a）和（65b）等价于归一化中心性条件（7），为方便起见，现重述如下：

$$\frac{\nabla f(x)}{|\nabla f(x)|} + \frac{\nabla g(x)}{|\nabla g(x)|} = 0,$$

此外，（65c）由（59）满足。根据归一化中心路径条件（7），可以使用 L2 范数为交点 $x^{\sharp}_{\zeta}$ 定义适当的残差：

$$r_{L2}(x^{\sharp}_{\zeta}) = \left| \frac{\nabla f(x^{\sharp}_{\zeta})}{|\nabla f(x^{\sharp}_{\zeta})|} + \frac{\nabla g(x^{\sharp}_{\zeta})}{|\nabla g(x^{\sharp}_{\zeta})|} \right|. \tag{66}$$

基于（66），我们给出误差度量 ϵ的正式定义。

定义 5.11 (解误差)。一种误差度量 ϵ 解点 $x^{\sharp}_{\zeta}$ 定义为残差的 L2 范数（66），

$$\epsilon(x^{\sharp}) = r_{L2}(x^{\sharp}). \tag{67}$$

下文将展示，该方法的误差 ϵ 相对于参数 ζ是有上界的。

定理 5.12。 假设 ***(A1)-(A4)*** 成立，并令 $\zeta \in [0,1)$， $x_0 \in$ 可行集。则， $x_\zeta^\sharp$ 的解误差相对于参数 ζ的上界为，

$$\epsilon(x_\zeta^\sharp) \leq \sqrt{2(1-\zeta)}. \tag{68}$$

证明。 根据引理 5.9，当从可行集 可行集初始化时，优化轨迹 $x(t;\ \zeta,\ x_0)$ 必须与约束边界相交。由 (59)，在交点 $x(t^\sharp;\ \zeta,\ x_0)$ 处有，

$$\frac{d}{dt} g(x_\zeta^\sharp) \geq 0. \tag{69}$$

由（39），

$$|\nabla g(x_\zeta^\sharp)|(\zeta + \cos\theta(x_\zeta^\sharp)) \leq 0. \tag{70}$$

因此，

$$\cos\theta(x_\zeta^\sharp) \leq -\zeta. \tag{71}$$

根据定义 5.11，有

$$\epsilon^2(x_\zeta^\sharp) = 2(1 + \cos\theta(x_\zeta^\sharp)). \tag{72}$$

因此，

$$\epsilon(x_\zeta^\sharp) \leq \sqrt{2(1-\zeta)}.$$

我们的证明至此完成。 □

注 6. 根据引理 5.10和定理 5.12，我们可以得出，优化轨迹达到近似解所需的时间复杂度为

$$|x_\zeta^\sharp - x^\star| \leq O(\frac{1}{\sqrt{t}}). \tag{73}$$

该结果达到了适用于一般优化问题的一阶方法的遍历收敛率。

6. 多重约束方法：基于对数障碍函数的表述

回顾问题（COP）的对数障碍函数，

$$\Phi(x) = -\sum_{i=1}^{m} \log(-g_i(x)),\ i = 1, ..., m.$$

我们首先注意到，当 $g_i(x) \to 0^-$ 时，障碍函数会无限增长，并且在可行集的边界处不可微。为克服这一困难，我们考虑可行集的一个子集，

$$\Omega_M = \{x : \Phi(x) \leq M\}, \tag{74}$$

其中 M 是一个足够大的正数，使得 Ω_M 逼近原可行集 Ω。障碍函数在 Ω_M 上二次连续可微，因此满足假设（A4）。

Set

$$G_M(x) = \Phi(x) - M,\ x \in \Omega_M.$$

原问题（COP）可近似地重新表述为

$$\begin{aligned} &\text{minimize} \quad f(x), \\ &\text{subject to} \quad G_M(x) \leq 0, \end{aligned} \tag{75}$$

因此可以应用当前的系统（DS）。

第 5 节所示的全局行为基于以下假设：在可行集 Ω 中 $\nabla g \neq 0$ 。一般而言，对于函数 G_M（x） $\forall x \in \Omega_M$，这一点未必总是成立。为了避开 $\nabla\Phi$（x） $= 0$ 所对应的点，我们建议使用梯度下降 $-\nabla f$（x）。这样便得到一个用于求解近似问题（75）的动力系统，

$$\frac{dx}{dt} = \begin{cases} -\dfrac{\nabla f(x)}{|\nabla f(x)|} - \zeta \dfrac{\nabla \Phi(x)}{|\nabla \Phi(x)|}, & \text{if } \nabla\Phi(x) \neq 0; \\ -\nabla f(x) & \text{if } \nabla\Phi(x) = 0. \end{cases} \tag{DSM}$$

注意，在计算实践中， $\nabla\Phi = 0$ 很少出现。上述系统中的第二个常微分方程为当前方法提供了一道保障。

推论 6.1. 假设满足假设*(A1)*、*(A2)*和*(A4)*，且 $\nabla g_i(x) \neq 0, i = 1, ..., m$ 属于可行子集可行集$_M$。设 $x_\zeta^\sharp$ 为近似问题 (DSM)的系统解点 (75)，则

$$x_\zeta^\sharp \in \{x : \Phi(x) = M, \cos\theta(x) \leq -\zeta\}. \tag{76}$$

而问题(75) 的 *KKT* 解的解误差为

$$\epsilon \leq O(\sqrt{1-\zeta}).$$

证明。 与定理5.12 中采用的类似方法给出了证明。 □

总之，采用问题（COP）的障碍函数形式，所得轨迹会达到一个近似局部解，该解位于子集 Ω 的边界上$_M$。还要注意，系统（DSM）不依赖于 M 的选择，并且 Ω_M 穷尽 Ω，即 $\Omega = \cup_{M>0}\Omega_M$。这意味着，所得轨迹会持续逼近原始可行集 Ω 的边界，并穿过任意依赖于 $_M$ 的子集 Ω 的边界 M。因此，该方法更易于实际实施，因为无需为停止准则定义额外参数 M，只需检查每个约束违反情况即可。

7. 算法与初步计算实验

我们介绍 GDAM 的两种计算实现，并展示其在凸和非凸约束优化中常见基准问题上的初步计算测试结果。

7.1. 计算算法

针对问题（COP）实现本文方法时，标准的一阶优化步骤如下：

$$x_{k+1} = x_k + \alpha_k \mathbf{s}_\zeta(x_k), \tag{77}$$

with

$$\mathbf{s}_\zeta(x_k) = \begin{cases} -\dfrac{\nabla f(x_k)}{|\nabla f(x_k)|} - \zeta \dfrac{\nabla \Phi(x_k)}{|\nabla \Phi(x_k)|}, & \text{if } \nabla\Phi(x_k) \neq 0; \\ -\nabla f(x_k) & \text{if } \nabla\Phi(x_k) = 0. \end{cases}$$

本文为极小化点非平凡地位于约束边界上的问题设计计算算法；我们的目标应用——形状优化和传感器网络定位——正是这种情况。

7.1.1. 固定长度步长

显然， α_k 的选择对于该方法的计算性能至关重要。当然，实现收敛这一目标的方法有很多种。在本文中，我们为该方法提出了一种固定长度步长规则，即

$$\alpha_k = \frac{\beta}{|s_\zeta(x_k)|}, \tag{78}$$

其中， β 是某个正参数。因此，

$$|x_{k-1} - x_k| = \beta. \tag{79}$$

这个步长规则可以说是较为激进的，因为在中心点 x_c处，当 $\zeta \to 1^-$，$|s_\zeta(x_c)| \to 0$。固定长度步长（78）利用以下方式对步长进行激进的重新缩放：

将 $|s_\zeta(x)|$的倒数用于缩放，从而产生有限的变量更新。次梯度方法也采用固定长度步长，通过动态调整 β 使其趋近于零，以处理解处的非光滑性，参见例如 [16, 命题 3.2。7]。对于约束问题，当解处有多个不等式约束处于活跃状态时，解顶点本身可能是非光滑的，并会导致其邻近区域内对数障碍函数的病态性。固定长度步长限制了更新的范围，因此可视为一种类似信赖域的方法，能够稳定迭代过程。最后，还可以加入各种线搜索方法，以动态调整参数 β。

7.1.2. 原始实现

基于定长步长方法的原始实现总结于算法1的伪代码中。对于常数参数 β，每当违反约束时，我们就终止优化。

算法1 原始 GDAM

初始化： x_0，ζ，β，β_{min}，$k = 0$
当未满足停止准则**时执行**
  如果 $\nabla\Phi(x_k) \neq 0$，**则**
    $\mathbf{s}_\zeta(x_k) \leftarrow \frac{\nabla f(x_k)}{|\nabla f(x_k)|} - \zeta \frac{\nabla\Phi(x_k)}{|\nabla\Phi(x_k)|}$
  else
    $\mathbf{s}_\zeta(x_k) \leftarrow -\nabla f(x_k)$
  结束如果
  更新： $\alpha_k \leftarrow \frac{}{|s_\zeta^{\alpha}(y_k)|}$，$x_{k+1} \leftarrow x_k + \alpha_k s_\zeta(x_k)$　$k \leftarrow k+1$
  （线搜索以更新 β**）**
结束 while 循环

在采用额外线搜索后，停止准则设定为同时满足以下两个条件：

1. 任一约束被违反；2. $\beta < \beta_{min}$。

例如，可以为该算法配备简单的回溯线搜索以提升性能，即每当违反约束时，就将步长 β 缩小一个缩放因子 τ，直到满足 $\beta < \beta_{min}$。

7.1.3. 加速实现

一种提高梯度下降法收敛速率的突出方法是 Nesterov 加速梯度法（NAG）[81]。我们通过将梯度下降更新步骤替换为 GDAM 更新步骤，把 NAG 应用于当前方法。我们在算法 2 中总结了其伪代码。请注意，与原始实现相比，新增了动量参数 m 和一系列辅助变量 $\{y_k\}$。

以下展示了所提出的原始版和加速 GDAM 算法在常见基准测试上的初步计算实验。计算结果表明，当前实现能够稳健地找到近似最优解，这在实际工程应用中可能具有实用价值。

算法 2 加速 GDAM

初始化： y_0, β, β_{min}, m, $k=0$
计算： $\alpha_0 = \frac{\alpha}{|s_\zeta(y_0)|}$, $x_0 \leftarrow y_0 + \alpha_0 s_\zeta(y_0)$, $y_1 \leftarrow x_0$, $k=1$
当 停止准则未满足**时**
  如果 $\nabla\Phi(x_k) \neq 0$ **则**
    $s_\zeta(y_k) \leftarrow \frac{\nabla f(y_k)}{|\nabla f(y_k)|} - \zeta \frac{\nabla\Phi(y_k)|\nabla\Phi}{(y_k)|}$
  else
    $s_\zeta(y_k) \leftarrow -\nabla f(y_k)$
  结束 if
  更新： $\alpha_k \leftarrow \frac{\beta}{|s_\zeta(y_k)|}$, $x_k \leftarrow y_k + \alpha_k s_\zeta(y_k)$ $y_{k+1} \leftarrow x_k + m(x_k - x_{k-1})$ $k \leftarrow k+1$
  （用于更新 β **的线搜索）**
结束 while 循环

7.2. 使用原始实现进行 *IEEE CEC 2006* 测试的实验

我们首先展示在 2006 年 IEEE进化计算大会进化算法竞赛中提出的不等式约束问题的数值实验 [66]。这些基准测试在进化算法领域被广泛使用。我们选择它们作为测试示例，原因有三。第一，这些问题定义明确，且具有不同特征 [89]。第二，使用一阶方法求解这些问题并非易事。第三，优化问题的形式相对简单，使我们能够更深入地了解当前方法的数值行为。我们选择这些不等式约束优化问题进行实验。问题 $G12$ 由于其可行集由 9^3 个互不相连的球体组成，因此不予考虑。对于问题 $G24$，其可行集由两个彼此不连通的子区域组成；我们选择位于包含已报告最优解的子区域内的初始设计。

我们使用 $\zeta = 0.98$ 和经过调优的定长步长参数 β，对原始实现（算法 1）进行数值实验。所有边界约束均视为不等式约束。在可行集中选取远离所报告最优解的随机初始化。我们通过以下方式评估目标函数值的误差：

$$\mathrm{err} = \frac{|f(\mathbf{x}^\star) - f(\mathbf{x}_\zeta)|}{1 + |f(\mathbf{x}^\star)|}, \tag{80}$$

其中 $\mathbf{x}^\star$ 是报告的全局最优解。如表 1所示，除问题 G19 外，标准 GDAM 得到的解与报告的全局最优解相比，目标函数值误差均小于 2e-2。这有些出人意料，因为 GDAM 是一种局部搜索算法。采用更短的步长和更大的参数 ζ可获得更精确的结果。对于问题 G19，使用当前的定长步长规则和合理的常数参数 $\mathbf{x}^\star =$ (1.6699e-17, 3.9637e-16, 3.9459, 1.060e-16, 3.2831, 9.9999, 1.1283e- 17, 1.2026e-17, 2.5071e-15, 2.2462e-15, 0.3708, 0.2785, 0.5238, 0.3886, 0.2982)无法得到报告的最优解。对于问题 G01、G04、G06、G07、G08 和 G24，我们找到了接近报告的最优设计的次优解。对于问题 G09、G10、G18 和 G19，我们找到了接近局部极小值的次优解。

表 1：算法 1 的参数 $\zeta = 0.98$ 和调优后的固定步长

概率	步长	迭代次数	f(x*)	f(xζ)	目标误差
G01	0.002	2362	-15	-14.7215	1.74e-2
G04	0.2	136	-3.0665e+4	-3.0657e+4	2.61e-4
G06	0.002	4826	-6.9618e+3	-6.8371e+3	1.79e-2
G07	0.0027	3009	24.3062	24.7876	1.90e-2
G08	0.01	66	-9.5825e-2	-9.5063e-2	6.95e-4
G09	0.05	120	6.8063e+2	6.9238e+2	1.72e-2
G10	0.35	5319	7.0492e+3	7.1898e+3	1.99e-2
G18	0.01	257	-0.8660	-0.8546	6.10e-3
G19	0.05	294	32.6556	2.7120e+2	7.09
G24	0.02	268	-5.5080	-5.4147	1.43e-2

7.3. 使用加速实现对 ***Maros*** 和 ***Meszaros*** 二次规划问题进行实验

Maros 和 Meszaros 测试集 [75] 包含一系列来自不同来源的困难凸二次规划问题。所包含的问题具有以下形式

$$\begin{aligned} \min_{x} \quad & \frac{1}{2}x^T Hx + c^T x + c_0, \\ \text{s.t.} \quad & Ax = b, \\ & l \le x \le u, \end{aligned} \tag{81}$$

其中，$x \in \mathbb{R}^n$, $H \in \mathbb{R}^{n\times n}$ 是一个对称正定矩阵，$A \in \mathbb{R}^{m\times n}$ 是一个稀疏矩阵，c, $c_0 \in \mathbb{R}^n$, $b \in \mathbb{R}^m$和 l，$u \in \mathbb{R}^n$。注意，l和 u的分量可能为无穷大，并且 $|c_0| < +\infty$。

算法 1 和 2 可以通过梯度投影法轻松扩展，以处理线性等式约束。假设 A 是非退化的，按照 Rosen 的方法 [91], 线性等式张成子空间的投影矩阵$Ax = b$ 写作

$$P_A = I_n - A^T(AA^T)^{-1}A. \tag{82}$$

任意向量 $v \in \mathbb{R}^n$ 的投影向量 v_p 为

$$v_p = P_A v. \tag{83}$$

于是，投影后的 GDAM 搜索方向为：

$$\mathbf{s}_{\zeta,P}(x_k) = -\frac{P_A\nabla f(x_k)}{|P_A\nabla f(x_k)|} - \zeta\frac{P_A\nabla\Phi(x_k)}{|P_A\nabla\Phi(x_k)|}. \tag{84}$$

实际上，我们可以预先计算并存储投影矩阵 P_A。对于大规模稀疏问题（n 和 m 较大，且 A 为稀疏矩阵），我们可以使用 Cholesky

分解来处理 AA^T，以提高计算效率。处理线性等式的更高级方法，例如 [25], 也可应用于当前方法。

我们在实验中使用了 Maros 和 Meszaros 实例的一个子集。这些实例经过精心挑选，涵盖多种问题模型，以便对专用二次规划问题求解器 [78]进行性能比较。我们通过以下方式评估目标函数值的误差：

$$\text{err} = \frac{|\text{obj}_{\text{best}} - \text{obj}|}{1 + |\text{obj}_{\text{best}}|}, \tag{85}$$

其中，obj_{best} 是各求解器得到的最小目标函数值。在表 2中，我们展示了本文方法的结果，并将其与成熟的求解器 Matlab quadprog、Gurobi [46], 和 OSQP [101]进行了比较。Matlab quadprog 和 Gurobi 均实现了用于凸二次规划的先进二阶求解器，而 OSQP 则是基于交替方向乘子法的一阶凸二次规划求解器 [23]。对于 Matlab quadprog 和 Gurobi，我们采用默认设置。对于 OSQP，我们将绝对容差和相对容差设为 $1e-4$，以获得更精确的解（默认设置均为 $1e-3$）。对于 GDAM，选择 $\zeta = 0.999$ 以及缩减参数为 $\tau = 0.3$ 的回溯线搜索。我们将 OSQP 和 GDAM 的最大迭代次数均设为 10000。所有实验均在一台 Linux 工作站上运行，该工作站配备 2.70 GHz、6 核的 AMD Ryzen 5 5600U 处理器和 16 GB 内存。

表2：大型 Maros 和 Meszaros 实例（感谢 [77] 提供 MATLAB 格式）

问题	运行时间 [秒]				迭代次数				目标误差			
	MATLAB	Gurobi	OSQP	GDAM	MATLAB	Gurobi	OSQP	GDAM	MATLAB	Gurobi	OSQP	GDAM
AUG2D	0.37	0.02	0.03	0.27	1	2	25	300	7.06e-16	opt[a]	5.84e-6	2.33e-7
AUG2DC	0.11	0.02	0.03	0.33	1	2	25	350	opt	2.62e-16	5.81e-6	1.29e-7
AUG2DCQP	0.95	0.06	0.59	1.46	10	16	1075	1050	4.89e-12	opt	7.30e-5	3.36e-2
AUG2DQP	0.90	0.06	0.55	1.53	10	16	975	1100	7.17e-12	opt	1.98e-4	3.34e-2
AUG3D	0.09	0.004	0.008	0.05	1	2	25	400	5.32e-15	opt	2.02e-10	1.03e-4
AUG3DC	0.01	0.006	0.009	0.05	1	2	50	350	opt	1.39e-15	6.69e-11	4.24e-5
AUG3DCQP	0.11	0.02	0.009	0.16	8	15	50	640	1.21e-8	opt	3.09e-6	8.61e-5
AUG3DQP	0.12	0.02	0.01	0.13	7	15	75	500	2.93e-7	opt	1.32e-9	4.67e-4
BOYD1	24.56	0.36	9.76	28.70	26	26	2750	10000	4.31e-12	opt	5.26	1.97e-2
CONT-050	0.29	0.02	1.01	0.32	11	15	5150	900	2.83e-6	opt	1.24e-4	2.18e-4
CONT-100	0.50	0.11	12.45	2.49	4	15	10000	1450	2.48e-2	opt	2.80e-4	1.47e-3
CONT-101	0.31	0.09	14.67	0.83	3	11	10000	452	0.69	opt	1.44e-2	5.97e-7
CONT-200	3.76	0.75	119.59	9.89	4	20	10000	750	0.30	opt	0.46	3.91e-3
CONT-201	1.93	0.52	112.91	5.21	3	14	10000	397	0.67	opt	0.69	1.06e-6
CONT-300	5.13	1.32	295.89	22.02	3	14	10000	715	0.67	opt	0.52	7.56e-6
CVXQP1 L	11.48	12.66	61.25	0.56	13	59	9075	800	opt	4.63e-8	2.23e-5	3.24e-5
CVXQP2 L	4.81	1.23	5.84	0.27	13	13	425	850	7.66e-12	opt	3.63e-9	3.12e-5
CVXQP3 L	17.06	8.99	29.56	0.83	15	51	3225	800	opt	8.52e-6	4.62e-5	2.76e-5
DTOC3	0.03	0.02	0.03	0.19	1	2	50	400	opt	1.37e-13	0.18	8.73e-5
HUES-MOD	0.25	0.01	0.03	0.17	11	14	175	650	1.46e-11	opt	∞	2.03e-5
HUESTIS	0.26	0.01	0.01	0.16	11	15	50	600	opt	5.78e-15	∞	1.76e-5
STCQP1	4.06	0.09	0.04	4.17	9	11	150	650	4.42e-10	opt	2.78e-8	9.99e-5
STCQP2	0.59	0.16	0.06	0.11	9	11	125	500	4.55e-10	opt	9.99e-7	1.72e-4
UBH1	0.16	0.01	0.03	1.78	3	7	50	2300	6.72e-11	opt	1.21	8.68

[a]选择目标函数值最低的可行解作为参考解，并将其视为“最优”。

如表 2所示，加速 GDAM 除 UBH1 外，在所有实例上都能找到近似最优解，这体现了其稳健性。总体而言，Gurobi 是速度最快、稳健性最强的求解器。GDAM 在求解 CVXQP* 实例时效率更高。通过设置更小的容差和更大的最大迭代次数，OSQP 能为 BOYD1、CONT* 和 DTOC3 实例找到更精确的解。我们注意到

与其他成熟的二次规划求解器不同，GDAM 尚未实现预求解和预条件处理策略，因为本文旨在展示该算法的稳健性。另一方面，这也为提升计算效率留下了改进空间。

8. 形状优化的应用

我们展示了原始 GDAM（算法 1）在 Computational Mechanics 中各种形状优化问题上的应用。我们首先简要介绍本研究采用的形状优化框架，然后展示使用 GDAM 得到的优化结果。

8.1. Vertex Morphing

我们旨在求解的形状优化问题几乎总是适定性不佳，这是因为基于有限元网格的离散设计空间非常庞大。为解决这一问题，我们采用 [52] 中介绍的顶点形变方法（VM）。

Vertex Morphing 的核心思想是通过设计控制量 $\mathbf{p} = [p_1, p_2, ..., p_n]^T$ 来控制离散表面坐标 $\mathbf{x} = [x_1, x_2, ..., x_n]^T$，并通过滤波函数对其进行滤波。VM 中采用的显式滤波是将坐标场 x 与核函数进行卷积。在离散系统中，这相当于一次矩阵向量乘法，即对经核函数加权的节点贡献求和。与标准显式滤波不同，VM 会将滤波过程应用两次。

首先，使用线性滤波矩阵$\mathbf{R}$ 的所谓前向映射步骤定义如下：

$$x_i = R_{ij} p_j. \tag{86}$$

同样，控制变量 $\delta\mathbf{p}$ 的变化会映射为设计构型 $\delta\mathbf{x}$的变化

$$\delta x_i = R_{ij} \delta p_j. \tag{87}$$

设计控制量是基于梯度的优化中的控制变量。变量的数量等于表面坐标的数量。根据微分的链式法则，响应函数 Ψ 相对于离散几何$\mathbf{x}$ 的灵敏度，会通过伴随映射矩阵或反向映射矩阵$\mathbf{R}^*$反向映射至设计控制量；对于规则网格，有$\mathbf{R}^* = \mathbf{R}^T$，

$$\frac{d\Psi}{dp_i} = \frac{d\Psi}{dx_j}\frac{dx_j}{dp_i} = R_{ji}\frac{d\Psi}{dx_j}. \tag{88}$$

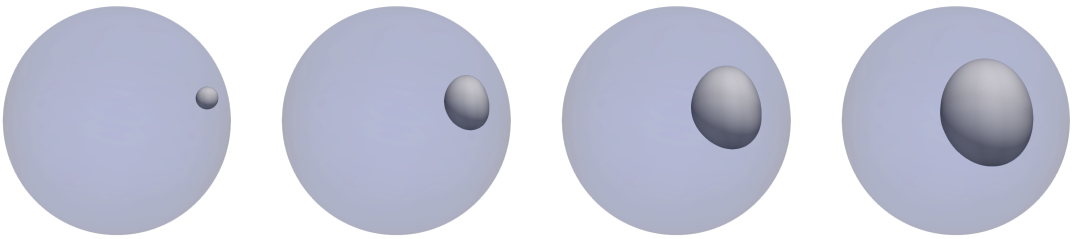
方程（87）和（88）可用于梯度下降框架，确保每次迭代中的形状平滑更新。

8.2. 学术示例

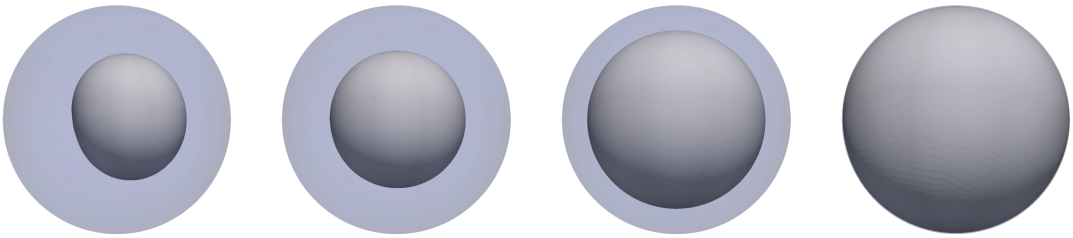
首先，我们展示一个学术性的凸优化问题。目标函数是最大化一个小球的体积。这个小球位于一个更大的球体内，该球体充当几何约束。两个球体的形状均由有限元网格表示 [114]。该优化问题可写为

$$\begin{aligned} &\text{minimize} \quad -V(x), \\ &\text{subject to} \quad g_i(x) \leq 0,\ i = 1, ..., m, \end{aligned} \tag{89}$$

其中 $V(x)$ 是体积函数， $g_i(x)$ 是针对第 i个设计节点逐点定义的几何约束， m 是设计网格的节点数， $x \in \mathbb{R}^{3m}$是设计球网格的节点坐标场。小球（设计球）的节点数为 19897。因此，设计变量总数为 59691，约束总数为 19897。对于多重约束，我们使用对数障碍函数，并选择参数 $\zeta = 0.95$。图 3展示了形状变化过程及相应的迭代次数。初始时，设计球靠近约束球的边界。在形状变化过程中，它逐渐移向约束球的中心，同时在每次迭代中调整自身形状。可以很容易看出，该优化问题正逐渐逼近并沿着中心路径前进，直至约束变为活跃约束并找到解。

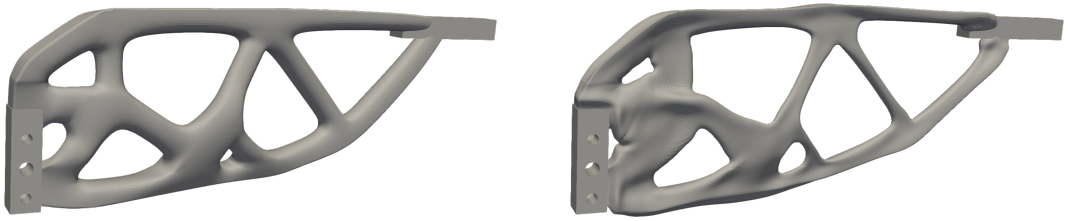


(a) 迭代 1 (b) 迭代 15 (c) 迭代 30 (d) 迭代 45



(e) 迭代 60 (f) 迭代 70 (g) 迭代 80 (h) 迭代 101

图3.：设计更新。



（a）初始框架设计　　（b）优化后的框架设计

图4：实际框架结构的设计优化。

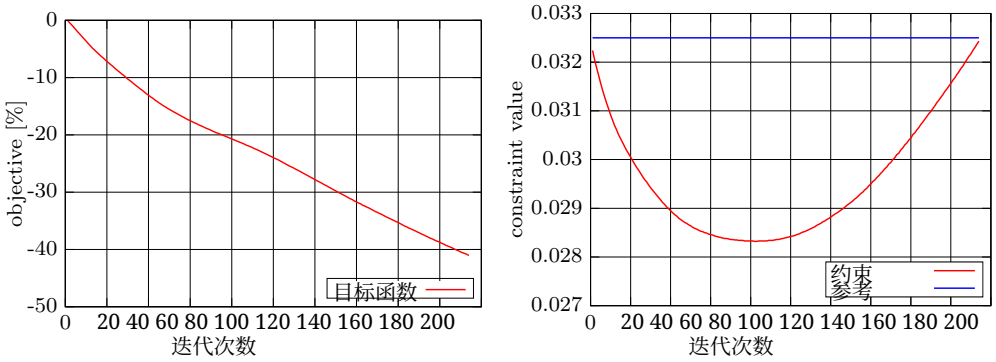
8.3. 实际示例

我们考虑一个实际应用中的形状优化问题。本文方法在 ShapeModule 中实现。ShapeModule 是一个灵活且与求解器无关的优化平台，提供优化算法以及形状控制方法，例如 Vertex Morphing [52]。该优化问题旨在满足载荷-位移约束（即每个表面节点的位移均受到限制）的条件下，最小化框架结构的质量。优化问题可写为

$$\begin{aligned} &\text{minimize} \quad M(x), \\ &\text{subject to} \quad g_i(x,u) \leq 0,\ i = 1, ..., m, \end{aligned} \tag{90}$$

其中，$M(x)$ 是质量函数，$g_i(x,u)$ 是针对 i 节点的逐点形式化位移约束，m 是设计表面网格的节点数，$x \in \mathbb{R}^{3m}$ 是设计表面网格的节点坐标场，$u \in \mathbb{R}^{3m}$ 是节点位移场。设计变量的数量为 144423，约束的数量为 48141。请注意，对于多重约束，我们可以像前面的测试示例一样，采用对数障碍函数形式。利用伴随灵敏度分析，可以高效地计算每个单独的位移约束梯度。在本应用中，我们使用软件 OptiStruct 提供的载荷-位移灵敏度，以符合标准的工业设计流程。我们选择参数 $\zeta = 0.95$。需要指出的是，我们的理论分析与该方法的大规模实现之间存在相当大的差距，因为我们的结果是渐近的，而预言机是黑盒。尽管如此，我们仍获得了与理论定性一致的结果。

在图 4 中，我们展示了初始框架设计以及经过 194 次迭代后的优化形状设计。如图5a所示，结构质量减少了 41%。在图 5b中，我们绘制了每次迭代的最大约束值 $g = max\{g_i\}$ 。在图 6中，我们展示了优化过程能够逼近并沿着 ζ邻域内的中心路径前进。



(a) 帧目标函数　　(b) 帧约束

图5：帧目标函数和约束的曲线图。

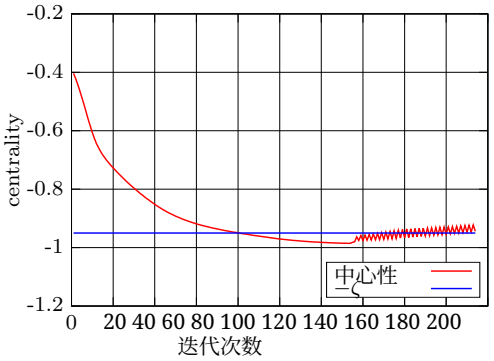


图 6：中心性度量图

备注 7. 沿着中心路径，中间设计不仅改善了目标函数，也改善了约束函数。以第 80 次迭代的设计为例：质量减少了 20.5%，位移减少了 12.1%。如果将原问题重新表述为双目标优化问题，并将质量和最大位移都设为目标函数，这些设计可以丰富设计选项。由此得到的中间设计与中心路径上的设计均为帕累托解的近似值。

备注 8. 本节重点介绍 GDAM 的原始实现，并展示其在求解基于节点的形状优化问题时的稳健性，因为稳健性被认为是实际形状设计应用取得成功的关键[20]。加速 GDAM 可用于进一步减少迭代次数。不过，加速 GDAM 和 Vertex Morphing 都会在优化过程中引入各自的辅助变量：$\{y_k\}$ 对于 GDAM 的算法2 而言是控制变量 $\mathbf{p}$，对于 Vertex Morphing 而言则是控制变量。此外，VM 中控制空间与变量空间之间的正向和反向映射，使得应用加速 GDAM 更为复杂。如何以最高效、稳健的方式结合这两种方法，值得进一步研究。这一问题独立于本文的研究范围，将留待未来工作。

9. 通过半定规划求解传感器网络定位

本节将应用本文的方法，通过半定规划（SDP）公式求解传感器网络定位（SNL）问题 [18]。首先，我们简要介绍基于半定规划的传感器网络定位的基础知识，随后展示计算实验，并将结果与最先进的一阶和二阶半定规划求解器进行比较。

9.1. 通过半定松弛进行传感器网络定位

传感器网络定位（SNL）旨在根据传感器之间（部分已知的）距离信息恢复未知的传感器位置，因此可视为一个逆问题。考虑用于传感器网络定位问题的非线性最小二乘（NLS）公式，其中包含 n 个传感器和 m 个二维锚点：

$$\min_{x_1,\dots,x_n\in\mathbb{R}^2} \sum_{(i,j)\in\mathcal{M}} \left(|x_i-x_j|^2-d_{ij}^2\right)^2 + \sum_{(k,j)\in\bar{\mathcal{M}}} \left(|a_k-x_j|^2-d_{kj}^2\right)^2, \tag{NLS}$$

其中，需要确定每个传感器 $x_i\in\mathbb{R}^d$、$i=1$、..... n 的位置，而每个锚点 $a_k\in\mathbb{R}^2$、$k=1$、..... m 的位置已知。距离 $\{d_{ij}:(i,\ j)\in\mathcal{M}\}$和 $\{d_{kj}:(k,\ j)\in\mathcal{M}\}$ 分别是传感器对之间以及传感器与锚点对之间的已知观测值。通常，在给定半径 r内检测这些距离。

我们可以将传感器网络定位表述为一个约束优化问题，

$$\begin{aligned}\min_{x_1,\dots,x_n\in\mathbb{R}^2}\quad & 0,\\ \text{s.t.}\quad & |x_i-x_j|^2=d_{ij}^2,\quad (i,j)\in\mathcal{M}\\ & |a_k-x_j|^2=d_{kj}^2,\quad (k,j)\in\bar{\mathcal{M}}\end{aligned} \tag{91}$$

传感器网络定位是一个典型的非凸二次优化问题 [72][80] ，对于该问题，我们不满足于找到一个局部解，而是希望得到全局最优解，从而恢复真实的传感器位置。沿用相关研究 [18]，我们使用半正定规划对这个 NP难非凸问题（91）进行松弛。首先，我们重写传感器对之间的距离，

$$|x_i-x_j|^2=(e_i-e_j)^TX^TX(e_i-e_j)=\langle E_{ij},X^TX\rangle, \tag{92}$$

其中，$e_i\in\mathbb{R}^n$ 是第 i个单位向量，X是一个 $2\times n$ 矩阵，其第 i列为 x_i，且$E_{ij}=($ $e_i-e_j)(e_i-e_j)^T\in\mathbb{S}^n$。类似地，我们有

$$|a_k-x_j|^2=\begin{bmatrix}a_k^T,-e_j^T\end{bmatrix}\begin{bmatrix}I_2 & X\\ X^T & X^TX\end{bmatrix}\begin{bmatrix}a_k\\ -e_j\end{bmatrix}=\langle\bar{M}_{kj},Z\rangle, \tag{93}$$

其中

$$\bar{M}_{kj}=\begin{bmatrix}a_k\\ -e_j\end{bmatrix}\begin{bmatrix}a_k^T,-e_j^T\end{bmatrix},$$

and

$$Z = \begin{bmatrix} I_2 & X \\ X^T & X^T X \end{bmatrix}. \tag{94}$$

此外，设

$$M_{ij} = \begin{bmatrix} 0 & 0 \\ 0 & E_{ij} \end{bmatrix}. \tag{95}$$

通过提升变量，我们得到关于变量 $Z \in \mathbb{S}^{n+2}$的等价表述（91），

$$\begin{aligned} \min \quad & 0, \\ \text{s.t.} \quad & \langle M_{ij}, Z \rangle = d_{ij}^2, \quad (i,j) \in \mathcal{M} \\ & \langle \bar{M}_{kj}, Z \rangle = d_{kj}^2, \quad (k,j) \in \bar{\mathcal{M}} \\ & Z_{1:2,1:2} = I_2, \ Z \succeq 0, \\ & \operatorname{rank}(Z) = 2. \end{aligned} \tag{96}$$

问题（96）仍然是一个极具挑战性的非凸优化问题。然而，通过提升变量，非凸性现在仅体现在秩约束上

$$\operatorname{rank}(Z) = 2.$$

半定松弛就是去掉这个秩约束，从而得到一个凸优化问题，

$$\begin{aligned} \min \quad & 0, \\ \text{s.t.} \quad & \langle M_{ij}, Z \rangle = d_{ij}^2, \quad (i,j) \in \mathcal{M} \\ & \langle \bar{M}_{kj}, Z \rangle = d_{kj}^2, \quad (k,j) \in \bar{\mathcal{M}} \\ & Z_{1:2,1:2} = I_2, \ Z \succeq 0. \end{aligned} \tag{SNLP}$$

（SNLP）的对偶问题写作：

$$\begin{aligned} \max_{V,y} \quad & \langle I_2, V \rangle + \sum_{(i,j)\in\mathcal{M}} y_{ij} d_{ij}^2 + \sum_{(k,j)\in\bar{\mathcal{M}}} y_{kj} d_{kj}^2 \\ \text{s.t.} \quad & \begin{bmatrix} V & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} + \sum_{(i,j)\in\mathcal{M}} y_{ij} M_{ij} + \sum_{(k,j)\in\bar{\mathcal{M}}} y_{kj} \bar{M}_{kj} + S = 0, \\ & S \succeq 0. \end{aligned} \tag{SNLD}$$

半定松弛在数学上十分优雅，许多应用表明，事实上，半定松弛能够对原始NP难问题给出紧的近似，例如最大割问题的开创性工作 [42]。对于传感器网络定位，文献 [97] 表明，如果传感器凭借所提供的距离信息是唯一可定位的，那么半定松弛能够为原始问题提供精确解。另一方面

，半定松弛的代价在于问题维数大幅增加，并引入了非平凡的半正定锥约束，这对大规模应用而言在计算上颇具挑战性。

请注意，传感器网络定位问题的半正定松弛公式（SNLP）是一个可行性规划——任意可行解都能解出该优化问题。由于优化迭代点位于可行集内，因此当前方法不能直接应用。另一方面，对偶公式（SNLD）是不等式约束问题，因此非常适合作为当前方法的优化公式。我们在 MATLAB R2022a 中实现加速 GDAM（算法2），以求解对偶问题（SNLD）。有关实现的更多细节见附录B。

9.2. 传感器网络定位示例

我们随机生成一组示例，其中包含 n 个节点，表示真实的传感器位置 $[-0.5, 0.5]^2$，n 的取值范围为 100 到 10000。此外，还生成了另外四个锚节点，其位置先验已知。在无线电探测范围 r 内测得了总计 m 对距离信息。为了衡量估计精度，我们使用均方根距离（RMSD）。该指标已广泛用于传感器网络定位的性能测试，其定义如下:

$$\mathrm{RMSD} = \left(\frac{1}{n} \sum_{i=1}^{n} |x_i - x_i^{\mathrm{true}}|^2 \right)^{1/2}, \tag{97}$$

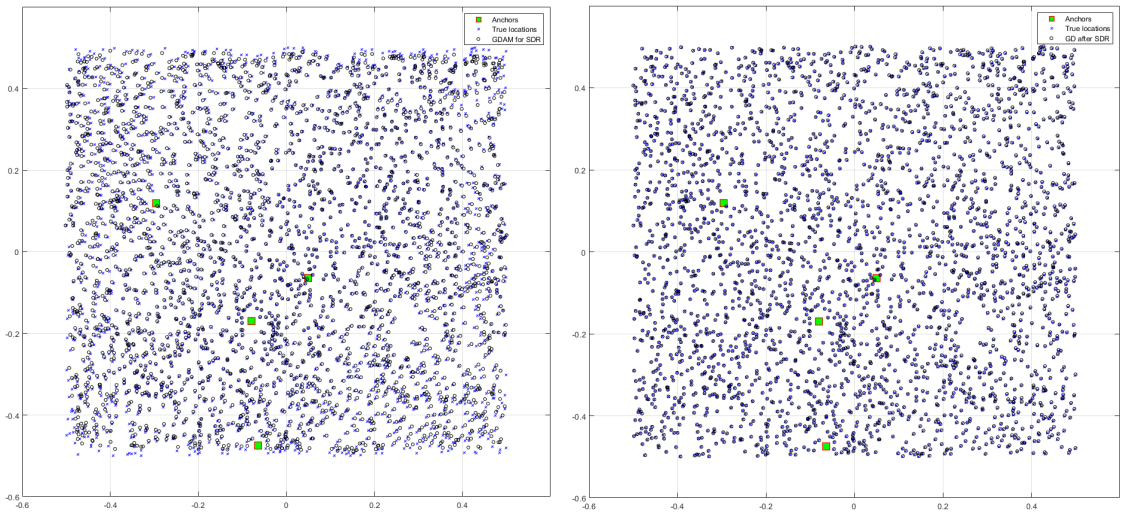
其中，x_i 是估计位置，x_i^{true}是传感器 i的真实位置。所有实验均在一台 Linux 工作站上运行，该工作站配备 2.70 GHz 6 核 AMD Ryzen 55600U 处理器和 16 GB 内存。

9.2.1. GDAM 结果

我们在表中报告了本文方法的运行时间和均方根偏差3。从均方根偏差来看，所得解的精度适中。解矩阵的特征值 Z 表明这些矩阵接近秩 2 矩阵。因此，本文方法不仅近似求解了半定松弛问题（SNLP），还以适中的精度求解了秩约束半正定规划问题（96）。由于（96）等价于原始可行性问题（91），我们可以期待本文方法为原始 NP-hard 传感器网络定位问题提供近似最优解。事实上，正如我们在大量数值实验中观察到的那样，将所得半定松弛解用作初始化，并对非线性最小二乘（NLS）应用梯度下降等局部搜索方法，可以准确恢复所有真实的传感器位置。图7 和8展示了一些图形结果。左图显示了对半定松弛（SNLD）应用加速 GDAM 得到的结果；右图则显示了对原始问题和低维问题（NLS）应用梯度下降，对 GDAM 解进行细化后的结果。

表3：SNL 问题的 GDAM 结果。

n	m	r	ζ	迭代次数	RMSD	运行时间（时:分:秒）
100	1058	0.3	0.9999	256	3.09e-3	$< 1s$
500	14699	0.21	0.9999	948	2.29e-3	00:00:09
1500	98007	0.18	0.9999	1312	3.27e-3	00:02:41
3000	246177	0.14	0.9999	1164	5.38e-3	00:12:37
5000	510116	0.12	0.9999	1231	5.63e-3	00:49:58
10000	1446144	0.10	0.9999	1427	3.97e-3	05:51:51



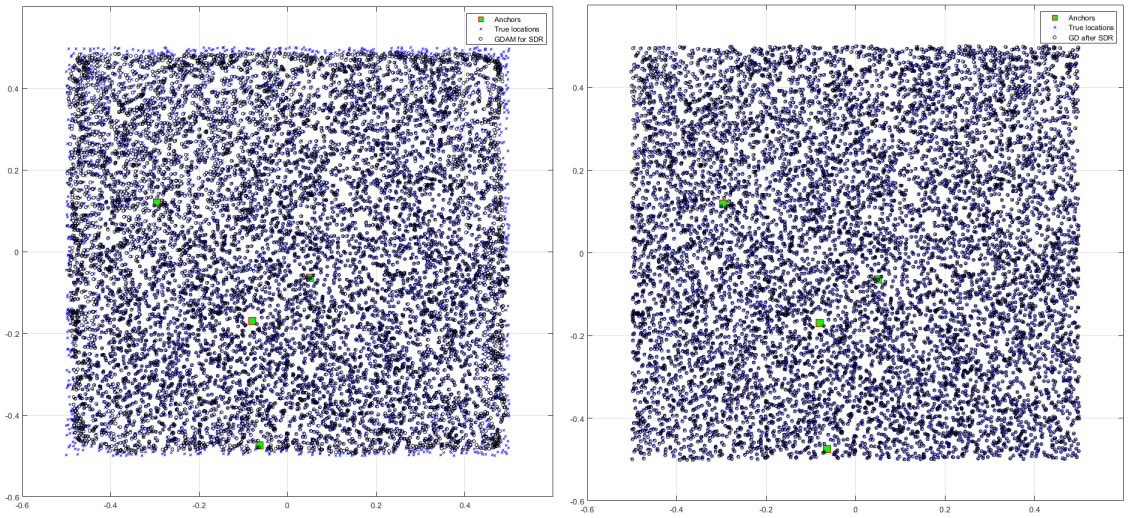
（a）用于半定松弛的 GDAM　　（b）半定松弛后的梯度下降

图 7：包含 3000 个传感器的传感器网络定位问题中 GDAM 的图形结果　　s.

9.2.2. 比较

我们从运行时间和均方根偏差两方面，将结果与三个成熟的求解器进行比较：DSDP v5。8 [15]，这是一种基于二阶对偶内点法的通用半正定规划求解器；SCS（分裂锥规划求解器）3.1 版。0 [82]，这是一种基于交替方向乘子法的通用大规模锥规划求解器；以及 SDPNAL+ 1 版。0 [103]，这是一种基于半光滑牛顿共轭梯度增广拉格朗日法的大规模半正定规划求解器。我们运行了表3中列出的相同示例。

DSDP：我们采用 DSDP 的默认设置，其中最优性间隙的容差设为 $1e-6$。DSDP 是一种高效的二阶求解器，适用于 SNL 问题；它利用问题的特性减少存储需求并提高效率。表 4 显示，DSDP 能以极高的精度求解问题。对于规模为 $n \geq 1500$ 的问题，工作站会耗尽内存。虽然 DSDP 能高精度地求解 SNL 问题，但 GDAM 找到近似解所需的时间要短得多。为了恢复真实的传感器位置，我们对低维非线性最小二乘问题（NLS）运行梯度下降，这是一种用于 SNL 问题的成熟方法 [17]，属于求解 SDP 松弛问题的一般两阶段策略



(a) 用于半定松弛的 GDAM　　(b) 半定松弛后的梯度下降

图 8：GDAM 用于包含 10000 个传感器的传感器网络定位问题的图形结果　　s.

其中，半正定规划解被用作非线性规划方法的初始化，以局部求解原始 NP难问题 [72]。

表 4：传感器网络定位问题的 DSDP 结果

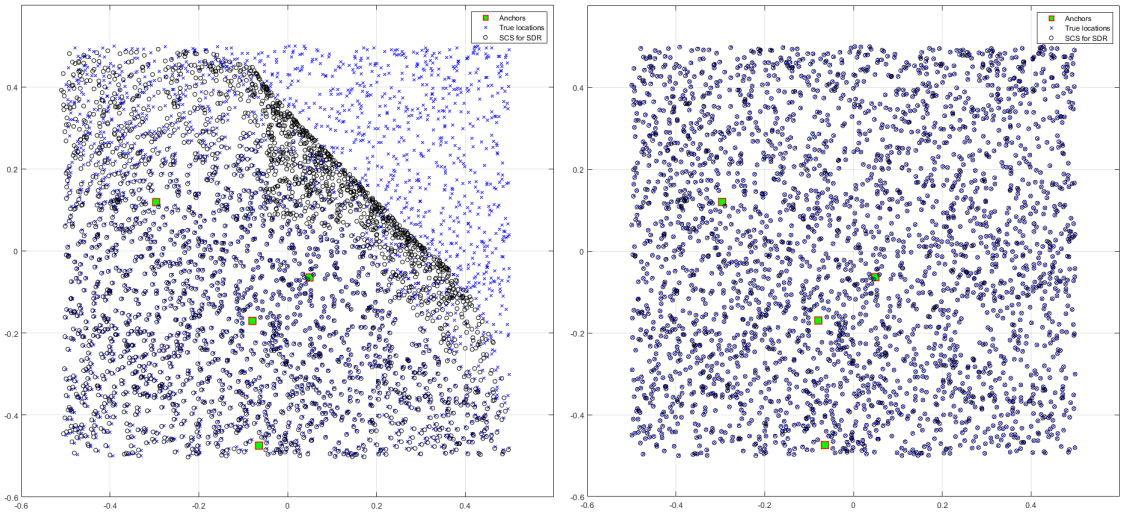
n	$tol.$	迭代次数	RMSD	运行时间（hh:mm:ss）
100	1e-6	36	5.06e-10	$< 1s$
500	1e-6	32	6.37e-10	00:06:47

SCS：在线性求解设置中，我们使用“稀疏间接法”，并将收敛容差设置在 $tol. = 1e-2$和 $tol. = 1e-4$之间，同时将最大运行时间设置为24小时。我们观察到，当目标函数为空时，SCS会遇到数值困难，这正是我们的可行性表述（SNLP）的情况。因此，我们向模型中添加目标函数 $\pm$迹(Z)，并发现运行时间对符号的选择相当敏感。因此，我们调整了符号 $\pm$ ，并报告性能更好的结果。我们使用YALMIP（版本2021年3月31日） [69] 进行问题建模以及与SCS的接口连接。

我们在表中报告 SCS 的结果5。对于 $n \leq 1500$的情形，采用$1e-3$容差可获得与 GDAM 相当准确的结果。对于 $n = 3000$的情形，需要采用更小的容差 $1e-4$，才能得到适合作为局部细化初始化的良好估计，如图9所示。对于所有实例，GDAM 在运行时间方面更具竞争力。例如，对于 $n = 3000$的情形，GDAM 用约 12 分钟便找到了一个精度尚可的解，其均方根偏差为 $= 5.38e-3$；而 SCS 在五个半小时后得到的解则较差。

表 5：SCS 在传感器网络定位问题上的结果。

n	$tol.$	迭代次数	RMSD	运行时间（hh:mm:ss）
100	1e-2	400	7.64e-2	00:00:02
	1e-3	650	2.91e-3	00:00:03
500	1e-2	400	1.57e-01	00:00:48
	1e-3	950	1.03e-02	00:01:23
1500	1e-2	400	2.29e-1	00:13:21
	1e-3	1225	1.82e-03	00:40:50
3000	1e-3	1950	1.21e-1	05:28:20
	1e-4	7225	6.14e-4	18:35:00



(a) 用于半定松弛的 SCS（ $tol. = 1e-3$ ）　(b) 用于半定松弛的 SCS（ $tol. = 1e-4$ ）

图9：SCS 在包含 3000 个传感器的传感器网络定位问题上的图形结果。

SDPNAL+：SDPNAL+ v1.0 是一种两阶段求解器：第一阶段采用交替方向乘子法算法，为第二阶段算法提供初始点；第二阶段算法为半光滑牛顿共轭梯度增广拉格朗日法。我们使用 SDPNAL+ 的默认设置，并将容差设为 $1e-2$ 和 $1e-3$，最大运行时间设为 24 小时。我们求解第9.2.1 节所示的测试示例，并在表6中报告结果。比较表 6 和表5可见，SDPNAL+与 SCS 的性能相当。在运行时间大致相同的情况下，对于 + 的情形，SDPNAL $n \leq 1500$得到的解更准确；而对于 $n = 3000$ 的情形，SCS 得到的解更准确（见 SDPNAL+ 在图10中的结果）。比较表6和表3 可见，对于所考虑的传感器网络定位问题，GDAM 在寻找均方根偏差精度尚可的解方面比 SDPNAL+ 更有效。

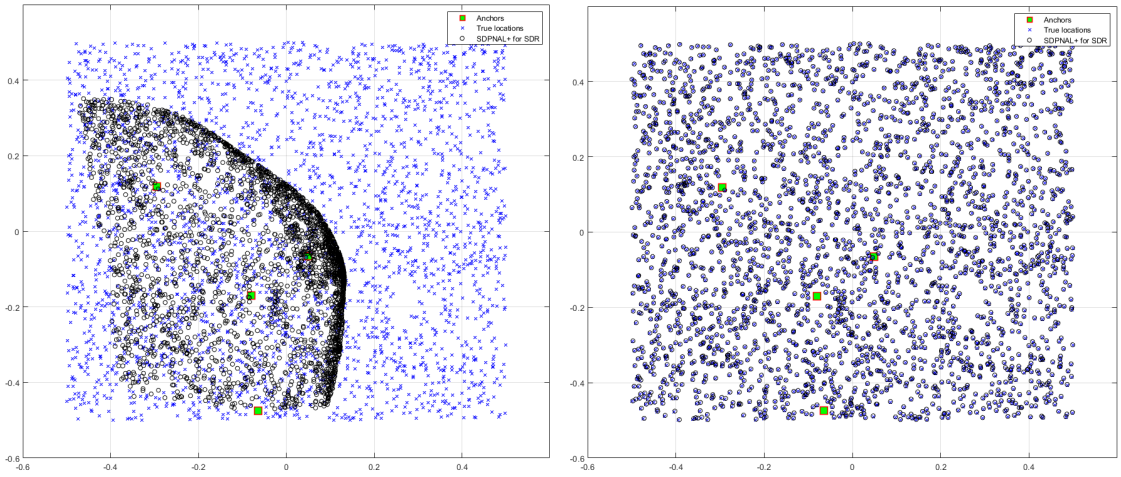
注9。 由 SNL 问题松弛得到的大规模半定规划是颇具挑战性的凸优化问题。即使是 SCS 和 SDPNAL+ 等先进的大规模半定规划求解器，也需要在计算时间与解的精度之间做出权衡。一个实际难点在于如何

预先选择这样的容差 先验。以 $n = 3000$为例，SCS 需要$tol. = 1e-4$ 才能得到精度适中的定位解，这比具有 $n \leq 1500$的问题所需的容差小一个数量级。对于同一问题 $n = 3000$，SDPNAL 的实际容差+ 可能介于 $1e-2$ 和$1e-3$之间，而这两个容差对应的运行时间相差约 18 小时。另一方面，如果不知道真实的传感器位置，设计实用的停止准则可能并非易事。在这方面，GDAM 只需使用一个固定参数 $\zeta = 0.9999$（对不同传感器数量、半径和锚点位置均适用），展现了其实用优势。理论上，ζ 与解的精度有关（见定理5.12），但其对大规模半定规划问题的实际意义可能更为深远，值得进一步研究。

备注 10。 与 SCS 和 SDPNAL 相比+，随着问题维数增加，GDAM 所需的迭代次数仅缓慢增长。事实上，据我们所知，GDAM 是唯一能够在笔记本工作站上将半定规划模型（SNLP）或（SNLD）求解至包含 10000 个传感器的方法。这清楚地表明，GDAM 有望成为解决某些大规模且困难的约束优化问题的实用优化方法。

表 6：SDPNAL+ 用于 SNL 问题的结果。

n	$tol.$	迭代次数。交替方向乘子法+/ SSN	RMSD	运行时间（hh:mm:ss）
100	1e-2	114/9	8.013e-4	00:00:01
500	1e-2	370/21	8.38e-3	00:00:42
	1e-3	500/22	1.43e-3	00:00:47
1500	1e-2	300/31	4.90e-2	00:20:43
	1e-3	3600/43	1.85e-5	00:36:53
3000	1e-2	414/38	2.15e-1	01:55:06
	1e-3	21603/165	1.40e-3	19:49:07



(a) SDPNAL+ 使用 $tol. = 1e-2$ 进行半定松弛　(b) SDPNAL+ 使用 $tol. = 1e-3$ 进行半定松弛

Fig图 10.：SDPNAL+ 用于含有 3000 个传感器的传感器网络定位问题的图形结果　s.

10. 讨论

本文提出了一种类梯度下降法，用于求解约束优化问题。GDAM 可视为一种约束优化梯度下降法，也可视为一种一阶内点法。我们给出了该方法全局收敛至一阶解的重要理论保证。我们提出了基于 GDAM 的计算算法，并展示了它们在两个工程问题中的应用：形状优化和传感器网络定位。计算实验表明，GDAM 稳健且极具竞争力，能够求得精度适中的解，并且可良好扩展至超大规模问题。鉴于这两种应用的特征存在相当大的差异，我们认为，GDAM 也适用于其他大规模约束优化问题，能够提供虽不精确但有用的解。

致谢

作者 LC 感谢德国国家高性能计算联盟提供的经费支持。我们感谢宝马集团的 ShapeModule 团队提供真实世界模型及其形状优化框架。

参考资料

[1] I.F. Akyildiz, W.Su, Y.Sankarasubramaniam, 和 E. Cayirci，无线传感器网络：综述，《计算机网络》38 (2002)，第 393–422 页。[2] T.A. Albring, M. Sagebaum, 和 N.R. Gauger，在 *SU2* 中使用离散伴随法进行高效气动设计，载于第 *17* 届 *AIAA/ISSMO* 多学科分析与优化会议。2016 年，第 3518 页。[3] M.M. Ali 和 T.L. Oliphant，一种用于约束非线性优化问题的基于轨迹的方法，《优化理论与应用杂志》177 (2018)，第 479–497 页。[4] G. Allaire，均匀化方法下的形状优化，第 146 卷，施普林格科学与商业媒体，2012。[5] G. Allaire、F. Jouve 和 A.M. Toader，基于灵敏度分析和水平集方法的结构优化，《计算物理学杂志》194 (2004)，第 363–393 页。[6] X. Allamigeon、P. Benchimol、S. Gaubert 和 M. Joswig，对数障碍内点法并非强多项式，《应用代数与几何学会杂志》2 (2018)，第 140–178 页。[7] E.D. Andersen 和 K.D. Andersen，用于线性规划的 *MOSEK* 内点优化器：齐次算法的实现，载于《高性能优化》，施普林格，2000 年，第 197–232 页。[8] H. Antil、R.H. Hoppe 和 C. Linsenmann，用于定常流问题形状优化的路径跟踪原始-对偶内点法，《数值数学杂志》15 (2007)，第 81–100 页。[9] J.S. Arora，最优设计导论，第 3 版，Elsevier，2011。[10] S. Arora、N. Cohen 和 E. Hazan，深度网络的优化：通过过参数化实现隐式加速，载于 第 *35* 届国际机器学习大会（*ICML 2018*）。国际机器学习学会（IMLS），2018 年，第 372–389 页。[11] H. Attouch、X. Goudou 和 P. Redont，带摩擦重球法，*I.* 连续动力系统：通过耗散动力系统的渐近分析，对实值函数的局部极小值进行全局探索

函数，通过耗散动力系统的渐近分析，*2000* 年，《当代数学通讯》*2*，第 *1–34* 页。[12] *D.A. Bayer* 和 *J.C. Lagarias*，线性规划的非线性几何。*I.* 仿射与射影尺度轨迹，《美国数学会汇刊》*314 (1989)*，第 *499–526* 页。[13] *W. Behrman*，一种高效的无约束优化梯度流方法，斯坦福大学博士论文，*1998* 年。[14] *M.P. Bendsoe* 和 *O. Sigmund*，拓扑优化：理论、方法与应用，施普林格科学与商业媒体，*2013* 年。[15] *S.J. Benson* 和 *Y. Ye*，算法 *875*：*DSDP5*——半定规划软件，《*ACM*数学软件汇刊》*34 (2008)*，第 *1–20* 页。[16] *D. Bertsekas*，凸优化算法，*Athena Scientific*，*2015* 年。[17] *P. Biswas*、*T.C. Lian*、*T.C. Wang* 和 *Y. Ye*，基于半定规划的传感器网络定位算法，《*ACM*传感器网络汇刊》*2 (2006)*，第 *188–220* 页。[18] *P. Biswas* 和 *Y. Ye*，用于自组无线传感器网络定位的半定规划，载于第 *3* 届国际传感器网络信息处理研讨会论文集。*2004* 年，第 *46–54* 页。[19] *P. Biswas* 和 *Y. Ye*，一种用于求解自组无线传感器网络定位所产生半定规划的分布式方法，载于《多尺度优化方法及应用》，施普林格，*2006* 年，第 *69–84* 页。[20] *K.U. Bletzinger*，结构的自由形状最优设计，载于《计算设计建模》，施普林格，*2011* 年，第 *25–37* 页。[21] *R.I. Bot*、*E.R. Csetnek* 和 *S.C. L´aszl´o*，一种用于结构化凸最小化问题的原始-对偶动力学方法，《微分方程杂志》*269 (2020)*，第 *10717–10757* 页。[22] *C.A. Botsaris*，微分梯度方法，《数学分析与应用杂志》*63 (1978)*，第 *177–198* 页。[23] *S. Boyd*、*N. Parikh*、*E. Chu*、*B. Peleato*、*J. Eckstein* 等，基于交替方向乘子法的分布式优化与统计学习，《机器学习基础与趋势》*3 (2011)*，第 *1–122* 页。[24] *S. Boyd* 和 *L. Vandenberghe*，凸优化，剑桥大学出版社，*2004* 年。[25] *J.J. Brust*、*R.F. Marcia*、*C.G. Petra* 和 *M.A. Saunders*，使用约简紧凑表示求解带线性等式约束的大规模优化问题，《*SIAM*科学计算杂志》*44 (2022)*，第 *A103–A127* 页。[26] *R.M. Buehrer*、*H. Wymeersch* 和 *R.M. Vaghefi*，协作式传感器网络定位：算法与实际问题，《*IEEE*学报》*106 (2018)*，第 *1089–1114* 页。[27] *R.H. Byrd*、*J.C. Gilbert* 和 *J. Nocedal*，一种基于内点技术的非线性规划信赖域方法，*Mathematical Programming 89 (2000)*，第 *149–185* 页。[28] *L. Chen*、*K. Bletzinger*、*A. Geiser* 和 *R. W¨uchner*，一种利用归一化响应梯度的奇异值分解求解不等式约束优化问题的修正搜索方向法，*Structural and Multidisciplinary Optimization 60 (2019)*，第 *2305–2323* 页。[29] *X. Chen* 和 *M.M. Kostreva*，可行方向法：综述，载于《优化进展》，施普林格，*2000* 年，第 *205–219* 页。[30]*L. Chizat* 和 *F. Bach*，利用最优传输研究过参数化模型中梯度下降的全局收敛，载于《神经信息处理系统进展》。*2018* 年，第 *3036–3046* 页。[31] *T.J. Chowdhury*、*C. Elkin*、*V. Devabhaktuni*、*D.B. Rawat* 和 *J. Oluoch*，无线传感器网络定位技术进展：综述，《计算机网络》*110 (2016)*，第 *284–305* 页。[32] *J. Cortes*，具有网络一致性应用的有限时间收敛梯度流，《自动化》*42 (2006)*，第 *1993–2000* 页。

[33] *E. De Klerk*、*J. Snyman*、*S.O.R. Group* 等，《一种用于线性约束最小化问题的可行下降锥方法》，*Computers & Mathematics with Applications 28 (1994)*，第*33–44*页。[34] *A. Dener*、*G.K. Kenway*、*J.E. Hicken* 和 *J. Martins*，《用于气动形状优化的非精确与拟牛顿算法比较》，载于第*53*届*AIAA*航空航天科学会议，*2015*年，第*1945*页。[35] *I. Diener*，《全局优化中的轨迹方法》，载于《全局优化手册》，施普林格，*1995*年，第*649–668*页。[36] *J.S. Dokken*、*S.K. Mitusch* 和 *S.W. Funke*，《*FEniCS* 和 *Firedrake* 中瞬态偏微分方程的自动形状导数》，*arXiv* 预印本 *arXiv:2001.10058*（*2020*）。[37] *F. Feppon*、*G. Allaire* 和 *C. Dapogny*，《用于约束优化的零空间梯度流及其在形状优化中的应用》，*ESAIM: Control, Optimisation and Calculus of Variations*（*2020*）。[38] *A.V. Fiacco* 和 *G.P. McCormick*，《非线性规划：序贯无约束最小化技术》，第*4*卷，*SIAM*，*1990*年。[39] *R. Fletcher* 和 *S. Leyffer*，《无需罚函数的非线性规划》，*Mathematical Programming 91*（*2002*），第*239–269*页。[40] *A. Forsgren*、*P.E. Gill* 和 *M.H. Wright*，《非线性优化的内点法》，*SIAM Review 44*（*2002*），第*525–597*页。[41] *B. Fr¨ohlich*、*J. Gade*、*F. Geiger*、*M. Bischoff* 和 *P. Eberhard*，《基于有限元的形状优化中的几何单元参数化与参数化模型降阶》，*Computational Mechanics 63*（*2019*），第*853–868*页。[42] *M.X. Goemans* 和 *D.P. Williamson*，《利用半定规划改进最大割和可满足性问题的近似算法》，*Journal of the ACM (JACM) 42*（*1995*），第*1115–1145*页。[43] *J. Gondzio*，《*25*年后的内点法》，*European Journal of Operational Research 218*（*2012*），第*587–601*页。[44] *A. Griewank* 和 *A. Walther*，《导数求值：算法微分的原理与技术》，*SIAM*，*2008*年。[45] *A.O. Griewank*，《用于全局优化的广义下降》，*Journal of Optimization Theory and Applications 34*（*1981*），第*11–39*页。[46] *Gurobi*，《*Gurobi* 优化器 *10.0*》（*2023*）。网址：*https://www.gurobi.com/*。[47] *R.T. Haftka*、*J. Sobieszczanski-Sobieski* 和 *S.L. Padula*，《关于多学科分析与设计优化方案的探讨》，*Structural Optimization 4*（*1992*），第*65–74*页。[48] *J. Haslinger* 和 *R.A. M¨akinen*，《形状优化导论：理论、近似与计算》，*SIAM*，*2003*年。[49] *U. Helmke* 和 *J.B. Moore*，《优化与动力系统》，施普林格科学与商业媒体，*1996*年。[50] *J. Herskovits*、*G. Dias*、*G. Santos* 和 *C.M. Soares*，《采用内点非线性规划算法的形状结构优化》，*Structural and Multidisciplinary Optimization 20*（*2000*），第*107–115*页。[51] *J.E. Hicken* 和 *D.W. Zingg*，《集成几何参数化与网格移动的气动优化算法》，*AIAA*期刊 *48*（*2010*），第*400–413*页。[52] *M. Hojjat*、*E. Stavropoulou* 和 *K.U. Bletzinger*，《用于基于节点的形状优化的顶点形变方法》，*Computer Methods in Applied Mechanics and Engineering 268*（*2014*），第*494–513*页。[53] *R.H. Hoppe*、*C. Linsenmann* 和 *H. Antil*，《用于线性和非线性斯托克斯流问题形状优化的自适应路径跟踪原始-对偶内点法》，载于国际大规模科学计算会议，施普林格，*2007*年，第*259–266*页。[54] *J.T. Hwang*，《航空航天系统大规模设计优化的模块化方法》，博士论文（*2015*）。[55] *F. Jarre*、*M. Kocvara* 和 *J. Zowe*，《采用内点法的桁架最优设计》，*SIAM Journal on Optimization 8*（*1998*），第*1084–1107*页。

[56] *M.I. Jordan*，《关于基于梯度优化的动力学、辛几何与随机视角》，载于《国际数学家大会会刊》，第*1*卷，*World Scientific*，*2018*年，第*525–550*页。[57] *D. Kandris*、*C. Nakas*、*D. Vomvas* 和 *G. Koulouras*，《无线传感器网络的应用：最新综述》，*Applied System Innovation 3*（*2020*），第*14*页。[58] *G. Kennedy*，《面向增材制造的大规模多材料拓扑优化》，载于第*56*届*AIAA/ASCE/AHS/ASC*结构、结构动力学与材料会议，*2015*年，第*1799*页。[59] *G.K. Kenway*、*G.J. Kennedy* 和 *J.R. Martins*，《用于高保真稳态气动弹性分析及伴随导数计算的可扩展并行方法》，*AIAA*期刊 *52*（*2014*），第*935–951*页。[60] *S. Kim*、*M. Kojima* 和 *H. Waki*，《利用半定规划松弛中的稀疏性进行传感器网络定位》，*SIAM Journal on Optimization 20*（*2009*），第*192–215*页。[61] *M. Kocvara* 和 *S. Mohammed*，《用于拓扑优化的原始-对偶内点多重网格方法》，*SIAM*科学计算期刊 *38*（*2016*），第*B685–B709*页。[62] *M. Koˇcvara* 和 *M. Stingl*，《*PENNON*：用于凸非线性规划和半定规划的代码》，*Optimization Methods and Software 18*（*2003*），第*317–333*页。[63] *J. Korelc*，《瞬态耦合问题原始分析与灵敏度分析的自动化》，*Computational Mechanics 44*（*2009*），第*631–649*页。[64] *C. Lemar´echal*，《柯西与梯度法》，*Doc Math Extra 251*（*2012*），第*10*页。[65] *N.E. Leonard*、*D.A. Paley*、*F. Lekien*、*R. Sepulchre*、*D.M. Fratantoni* 和 *R.E. Davis*，《集体运动、传感器网络与海洋采样》，*IEEE*会刊 *95*（*2007*），第*48–74*页。[66] *J. Liang*、*T.P. Runarsson*、*E. Mezura-Montes*、*M. Clerc*、*P.N. Suganthan*、*C.C. Coello* 和 *K. Deb*，《*CEC 2006*特别专题：约束实参数优化的问题定义与评估准则》，应用力学期刊 *41*（*2006*），第*8–31*页。[67] *J. Liedmann*、*S. Gerke*、*F.J. Barthold* 和 *M. Br¨unig*，《*X0* 试样的形状优化：理论、数值模拟与实验验证》，*Computational Mechanics*（*2020*），第*1–17*页。[68] *K. Linkwitz* 和 *H.J. Schek*，《关于预应力索网结构计算的若干说明》，*Ingenieur-Archiv 40*（*1971*），第*145–158*页。[69] *J. Lofberg*，《*YALMIP*：用于 *MATLAB* 建模与优化的工具箱》，载于*2004*年*IEEE*机器人与自动化国际会议（*IEEE*目录号 *04CH37508*），*IEEE*，*2004*年，第*284–289*页。[70] *D.G. Luenberger* 和 *Y. Ye*，《线性与非线性规划》，第*5*版，施普林格，*2021*年。[71] *D. Luft*、*V.H. Schulz* 和 *K. Welker*，《利用伴随变量和变分不等式进行形状优化的高效技术》，*SIAM Journal on Optimization 30*（*2020*），第*1922–1953*页。[72] *Z.Q. Luo*、*W.K. Ma*、*A.M.C. So*、*Y. Ye* 和 *S. Zhang*，《二次优化问题的半定松弛》，*IEEE*信号处理杂志 *27*（*2010*），第*20–34*页。[73] *B. Maar* 和 *V. Schulz*，《用于拓扑优化的内点多重网格方法》，*Structural and Multidisciplinary Optimization 19*（*2000*），第*214–224*页。[74] *A. Majumdar*、*G. Hall* 和 *A.A. Ahmadi*，《半定规划可扩展性的最新改进及其在机器学习、控制与机器人学中的应用》，控制、机器人与自主系统年度评论 *3*（*2020*），第*331–360*页。[75] *I. Maros* 和 *C. Meszaros*，《*Maros* 和 *Meszaros* 凸二次规划测试问题集》（*2022*）。网址：*https://www.cuter.rl.ac.uk/Problems/marmes.html*。[76] *J.R. Martins* 和 *A.B. Lambe*，《多学科设计优化：架构综述》，*AIAA*期刊 *51*（*2013*），第*2049–2075*页。[77] *K. Mihi´c*、*M. Zhu* 和 *Y. Ye*，《*RACQP*》（*2019*）。网址：*https://github.com/kmihic/RACQP*。[78] *K. Mihi´c*、*M. Zhu* 和 *Y. Ye*，《二次优化中多块交替方向乘子法的随机化管理》，*Mathematical Programming*

Computation 13（2021），第339–413页。[79] R. Najian Asl、I. Antonau、A. Ghantasala、W.G. Dettmer、R. W¨uchner和K.U. Blet-zinger，涉及非匹配网格的流固耦合伴随形状灵敏度分析的分区方案，Optimization Methods and Software（2020），第1–31页。[80] Y. Nesterov、H. Wolkowicz和Y. Ye，非凸二次优化的半定规划松弛，载于《半定规划手册》，施普林格，2000年，第361–419页。[81] Y.E. Nesterov，求解收敛速率为O（1/kˆ 2）的凸规划问题的一种方法，载于《苏联科学院院刊》，第269卷，1983年，第543–547页。[82] B. O'Donoghue，线性互补问题齐次嵌入的算子分裂，SIAM Journal on Optimization 31（2021），第1999–2023页。[83] B. O'donoghue和E. Candes，加速梯度方案的自适应重启，Foundations of Computational Mathematics 15（2015），第715–732页。[84] R.E. Perez、P.W. Jansen和J.R.R.A. Martins，pyOpt：用于非线性约束优化的基于Python的面向对象框架，Structures and Multidisciplinary Optimization 45（2012），第101–118页。[85] L. Perko，《微分方程与动力系统》，第7卷，施普林格科学与商业媒体，2008年。[86] I. Polik、T. Terlaky和Y. Zinchenko，SeDuMi：一个用于锥优化的软件包，载于IMA优化与控制研讨会，明尼苏达大学，明尼阿波利斯。Citeseer，2007年。[87] F.A. Potra和S.J. Wright，内点法，Journal of Computational and Applied Mathematics 124（2000），第281–302页。[88] D. Puccinelli和M. Haenggi，无线传感器网络：泛在感知的应用与挑战，IEEE电路与系统杂志5（2005），第19–31页。[89] R. Rao，Jaya：一种用于求解约束和无约束优化问题的简单新型优化算法，International Journal of Industrial Engineering Computations 7（2016），第19–34页。[90] T.D. R¨aty，面向公共安全的当代远程监控系统综述，IEEE系统、人类与控制论汇刊C辑（应用与评论）40（2010），第493–515页。[91] J.B. Rosen，非线性规划的梯度投影法。第一部分：线性约束，Journal of the Society for Industrial and Applied Mathematics 8（1960），第181–217页。[92] C. Schillings、S. Schmidt和V. Schulz，确定性与不确定性气动设计的高效形状优化，Computers & Fluids 46（2011），第78–87页。[93] S. Schmidt、C. Ilic、V. Schulz和N.R. Gauger，基于形状微积分的三维大规模气动形状优化，AIAA期刊51（2013），第2615–2627页。[94] V.H. Schulz、M. Siebenborn和K. Welker，面向PDE约束形状优化的拉格朗日–牛顿方法，载于《形状优化新趋势》，施普林格，2015年，第229–249页。[95] J.A. Sethian，《水平集方法与快速行进方法：计算几何、流体力学、计算机视觉和材料科学中的演化界面》，第3卷，剑桥大学出版社，1999年。[96] J. Snyman和L. Fatti，具有动态搜索轨迹的多起点全局最小化算法，《优化理论与应用杂志》54（1987），第121–141页。[97] A.M.C. So和Y. Ye，传感器网络定位的半定规划理论，Mathematical Programming 109（2007），第367–384页。[98] J. Sokolowski和J.P. Zol´esio，《形状优化导论》，载于《形状优化导论》，施普林格，1992年，第5–12页。[99] N. Stander和J. Snyman，结构优化的一种新型一阶内点可行方向法，International Journal for Numerical Methods in Engineering 36（1993），

第4009–4025页。[100] N. Stander、J. Snyman和J. Coster，SAM算法在结构优化中的稳健性与效率，International Journal for Numerical Methods in Engineering 38（1995），第119–135页。[101] B. Stellato、G. Banjac、P. Goulart、A. Bemporad和S. Boyd，OSQP：二次规划的算子分裂求解器，Mathematical Programming Computation 12（2020），第637–672页。[102] W. Su、S. Boyd和E. Candes，用于建立Nesterov加速梯度法模型的微分方程：理论与见解，载于《神经信息处理系统进展》，2014年，第2510–2518页。[103] D. Sun、K.C. Toh、Y. Yuan和X.Y. Zhao，SDPNAL+：带边界约束的半定规划Matlab软件（1.0版），Optimization Methods and Software 35（2020），第87–115页。[104] T. Sun、L.J. Chen、C.C. Han和M. Gerla，用于行星探测的可靠传感器网络，载于《2005 IEEE网络、传感与控制会议论文集》，2005年。IEEE，2005年，第816–821页。[105] K. Svanberg，移动渐近线法——一种新的结构优化方法，International Journal for Numerical Methods in Engineering 24（1987），第359–373页。[106] K.C. Toh、M.J. Todd和R.H. T¨ut¨unc¨u，SDPT3的实现与使用——用于半定-二次-线性规划的Matlab软件包4.0版，载于《半定、锥与多项式优化手册》，施普林格，2012年，第715–754页。[107] M. Towara和U. Naumann，OpenFOAM的离散伴随模型，Procedia Computer Science 18（2013），第429–438页。[108] S. Wang、X. Yang和K.L. Teo，约束非线性优化问题的统一梯度流方法，Computational Optimization and Applications 25（2003），第251–268页。[109] Z. Wang、S. Zheng、Y. Ye和S. Boyd，传感器网络定位半定规划方法的进一步松弛，SIAM Journal on Optimization 19（2008），第655–673页。[110] A.C. Wilson、B. Recht和M.I. Jordan，优化中加速方法的李雅普诺夫分析，Journal of Machine Learning Research 22（2021），第1–34页。[111] M. Yamashita、K. Fujisawa、M. Fukuda、K. Kobayashi、K. Nakata和M. Nakata，SDPA系列求解大规模半定规划的最新进展，载于《半定、锥与多项式优化手册》，施普林格，2012年，第687–713页。[112] Y. Ye，《内点算法：理论与分析》，第44卷，John Wiley & Sons，2011年。[113] Y. Zheng、G. Fantuzzi和A. Papachristodoulou，可扩展半定规划与多项式优化的弦图分解和因子宽度分解，Annual Reviews in Control 52（2021），第243–279页。[114] O.C. Zienkiewicz、R.L. Taylor、P. Nithiarasu和J. Zhu，《有限元法》，第3卷，McGraw-hill London，1977年。[115] G. Zoutendijk，《可行方向法：线性与非线性规划研究》，Elsevier，1960年。

附录 A. 问题 Γ^{ς} 的解析轨迹推导 (29)

对于问题 (29)，搜索方向场$\mathbf{s}_{\varsigma}$ 为

$$\begin{aligned}\mathbf{s}_{\zeta} &= -\frac{\nabla f}{|\nabla f|} - \zeta\frac{\nabla g}{|\nabla g|} \\ &= \frac{-1}{\sqrt{x_1^2+x_2^2}}\left\{x_1, x_2 - \zeta\sqrt{x_1^2+x_2^2}\right\}^T.\end{aligned} \tag{A1}$$

根据常微分方程的唯一性定理，当初始设计满足 $x_1^0 \neq 0.$ 时，向量场 $\mathbf{s}_{\varsigma}$ 的积分曲线具有非零分量 x_1 。因此，我们有，

$$\frac{dx_2}{dx_1} = \frac{x_2}{x_1} - \zeta\frac{\sqrt{x_1^2+x_2^2}}{x_1} \tag{A2}$$

当 $x_1 > 0$ 时，令 $x_2 = x_1 u$，则有 $\frac{dx_2}{dx_1} = u + x_1\frac{du}{dx_1}$。因此，

$$u + x_1\frac{du}{dx_1} = u - \zeta\sqrt{1+u^2}. \tag{A3}$$

我们将 (A3) 改写为

$$\frac{du}{\sqrt{1+u^2}} = \frac{-\zeta}{x_1}dx_1. \tag{A4}$$

求解方程 (A4) 得

$$\log\left(u + \sqrt{1+u^2}\right) = -\zeta\log x_1 + a_0. \tag{A5}$$

Or,

$$x_2 + \sqrt{x_1^2+x_2^2} = a x_1^{1-\zeta}. \tag{A6}$$

类似地，对于 $x_1 < 0$，轨迹的形式为

$$x_2 + \sqrt{x_1^2+x_2^2} = a|x_1|^{1-\zeta}. \tag{A7}$$

假设初始设计为 $\mathbf{x}^0 = (x_1^0,\ x_2^0)$，则可得

$$a_{\varsigma} = \frac{x_2^0 + \sqrt{(x_1^0)^2+(x_2^0)^2}}{|x_1^0|^{1-\zeta}}. \tag{A8}$$

因此，

$$x_2 + \sqrt{x_1^2+x_2^2} = \left|\frac{x_1}{x_1^0}\right|^{1-\zeta}\left(x_2^0 + \sqrt{(x_1^0)^2+(x_2^0)^2}\right). \tag{A9}$$

令 $\bar{x}_2 = \frac{1}{2}\left(x_2^0 + \sqrt{(x_1^0)^2 + (x_2^0)^2}\right)$，则可得到sζ的轨迹 Γ^ζ

$$x_2 + \sqrt{x_1^2 + x_2^2} = 2\bar{x}_2 \left|\frac{x_1}{x_1^0}\right|^{1-\zeta}. \tag{A10}$$

设 $(x_{1,\zeta}, x_{2,\zeta})$ 是轨迹 Γ^ζ 上 x_2 分量最大的点，则

$$\begin{cases} x_{2,\zeta} + \sqrt{x_{1,\zeta}^2 + x_{2,\zeta}^2} = 2\bar{x}_2 \left|\frac{x_{1,\zeta}}{x_1^0}\right|^{1-\zeta}, \\ x_{2,\zeta} - \zeta \cdot \sqrt{x_{1,\zeta}^2 + x_{2,\zeta}^2} = 0. \end{cases} \tag{A11}$$

由此可得

$$(x_{2,\zeta})^\zeta = \frac{2\zeta}{1+\zeta} \frac{\bar{x}_2}{|x_1^0|^{1-\zeta}} \left(\frac{\sqrt{1-\zeta^2}}{\zeta}\right)^{1-\zeta}, \tag{A12}$$

and

$$|x_{1,\zeta}| = \frac{\bar{x}_2}{\zeta}\sqrt{1-\zeta^2}. \tag{A13}$$

设 $\zeta \to 1^-$，则 $x_{1,\zeta} \to 0$ 且 $x_{2,\zeta} \to \bar{x}_2$。轨迹 Γ^ζ 收敛到曲线Γ，该曲线是抛物线

$$x_1^2 = 4\bar{x}_2^2 - 4x_2\bar{x}_2, \quad x_1 \in (0, x_1^0)(\quad \text{or}(x_1^0, 0),) \tag{A14}$$

与 x_2轴上的区间 $(0, \bar{x}_2)$ 的并集。

附录 B. 传感器网络定位问题中 GDAM 的实现细节

除第7.1节求解 SNL 问题的内容外，我们还报告了更多实现细节。

B.1. 半正定锥约束的对数障碍函数

下面简要说明如何使用标准对数障碍函数处理半正定约束。对于原始半定规划形式（SNLP），矩阵变量的半正定锥约束 $Z \succeq 0$ 可表示为 Z 仅具有非负特征值，即

$$\lambda_i \geq 0, i = 1, ..., n, \tag{B1}$$

其中，为便于一般性说明，我们假设 $Z \in \mathbb{S}^n$ 。使用对数障碍函数处理（B1），可得

$$\Phi(Z) = \sum_{i=1}^{n} -\log \lambda_i = -\log\left(\Pi_{i=1}^{n} \lambda_i\right) = -\log\left(\det(Z)\right). \tag{B2}$$

关于矩阵变量 Z， $\Phi(Z)$ 的梯度写作

$$\frac{\partial}{\partial Z_{ij}}\left(-\log\det Z\right) = -\left(Z^{-1}\right)_{ji}. \tag{B3}$$

对于对偶半定规划表述（SNLD），半正定锥约束可以用类似的方式表示。注意，此时变量是对偶变量 $\mathbf{y}$，因此有

$$\Phi(\mathbf{y}) = -\log\left[\det(S(\mathbf{y}))\right]. \tag{B4}$$

$\Phi(\mathbf{y})$ 的梯度写作

$$\frac{\partial}{\partial y_i}\Big(-\log\left[\det(S)\right]\Big) = \langle S^{-1}, \mathbf{A}_i\rangle_{\mathbb{S}^n}, \tag{B5}$$

其中 $A = \{M_{ij}, M_{kl}, (i,j) \in \mathcal{M},\ (k,l) \in \mathcal{M}\}$。由（B3）和（B5）可知，要在对数障碍框架下使用基于梯度的方法，必须对 $\mathbb{S}^n$ 中的矩阵进行求逆；这对原问题和对偶公式均适用。

B.2. 预求解

预求解是优化算法实际实现中的重要环节，可将输入问题转换为更简单的问题。对于 SNL 问题，我们采用一种简单的预求解策略，将网络几何重新缩放至范围 $[-5,5]^2$。从数值角度看，这可视为对所构造的对偶半定规划问题（SNLD）进行预条件处理。与文献中常见的标准传感器范围 $[-0.5,0.5]^2$相比，对偶内点法求解器 DSDP 通过减少迭代次数，验证了该预求解策略的有效性。我们注意到，通用预求解／预条件策略（例如参见 [113]）可能会进一步提升该方法的性能，这留待未来研究。

B.3. 严格可行初始化

求解半定规划时，内点法有多种初始化策略。本文采用先第一阶段后第二阶段方法，即先尝试寻找一个可行点（在我们的情形中，是对偶问题的可行点），然后开始主求解过程。注意，当SNLD满足以下条件时，对偶问题可行： $V = 0$ 且 $y_{ij} = 0$ 对所有 $(i,j) \in \mathcal{M}$ 以及 $y_{kj} = 0$ 对所有 $(i,j) \in \mathcal{M}$。为方便起见，我们将此零初始化记为 y_0。然而，点 y_0 并非可行集中的内点，因此无法以此作为 GDAM 的起点。为此，我们提出执行若干步梯度下降，

从 y_0 开始，求解以下辅助对数障碍问题，

$$\min \quad -\log\det\left(\lambda I_{n+2}-\begin{bmatrix} V & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}-\sum_{(i,j)\in\mathcal{M}} y_{ij}M_{ij}-\sum_{(k,j)\in\bar{\mathcal{M}}} y_{kj}\bar{M}_{kj}\right), \tag{B6}$$

其中 λ 是一个正数，我们将其取为 10。0。在我们的实现中，经过几步梯度下降后，可以很容易地获得对偶问题（SNLD）的可行内点初始化。

B.4. 主求解

找到严格可行初始化后，主求解阶段开始，并应用加速算法 2 求解对偶问题（SNLD）。为进一步提升实际性能，我们针对 SNL 问题开发了自适应步长和重启策略，并对其启发式规则进行了调优。我们为迭代点实现了类回溯步长规则，以使其保持在可行集中；也就是说，每当违反约束时，就通过缩放因子减小步长。我们将初始步长设为$\alpha_0 = 1.618e1$，并将最小步长设为 $\alpha_{min} = 1e-8$。重启策略的设计受文献 [83]启发。具体而言，当目标函数值增加或目标函数下降速度显著变慢时，我们会重启优化算法[4]。为提高效率，我们不会将步长重置为 α_0，而是将其设为重启前步长的某个倍数，即 $\alpha_{r+1} = \kappa\alpha_r$，其中 κ 是一个常数，r 是每次重启的索引。需要注意的是，目前实现的启发式策略仍较为基础，还可以采用更复杂的方法来进一步提升实际性能。尽管如此，我们发现当前实现具有良好的稳健性，并且能够很好地扩展到大规模问题。

B.5. 后处理

求解对偶半定松弛问题（SNLD）后，需要额外执行一个步骤，以恢复原始解 Z，其中包含实际的传感器位置。分别用 $\mathcal{F}_p$ 和 $\mathcal{F}_d$表示（SNLP）和（SNLD）的可行集。记 $\mathcal{F} = \mathcal{F}_p \times \mathcal{F}_d$，并将$o$ o的内部 $\mathcal{F}$ 记为 $\overset{o}{\mathcal{F}}$。假设 $\overset{o}{\mathcal{F}} \neq \emptyset$，半正定规划的中心路径可表示为

$$\mathcal{C} = \{(Z, y, S) \in \overset{o}{\mathcal{F}} : ZS = \eta I, 0 < \eta < \infty\}, \tag{B7}$$

其中 η 为障碍参数（参见（5））， I 为单位矩阵。原始解 Z 可通过对偶问题解 S 得到:

$$Z = \eta S^{-1}, \tag{B8}$$

[4]在我们的实际实现中，目标函数值按固定频率进行监测。

其中 η 为障碍参数，在 GDAM 的计算框架内，可用（11）近似，

$$\eta(x_\zeta^\sharp) = \zeta \frac{|\nabla f(x_\zeta^\sharp)|}{|\nabla \Phi(x_\zeta^\sharp)|},$$

其中， $x_\zeta^\sharp$ 是通过参数 ζ找到的解。为提高精度，我们以主求解的优化器作为初始化，使用 $\zeta = 1.0$ 运行 GDAM。在我们的实现中，我们为后处理设置 $\alpha_{min} = 1e-10$ 。请注意，较小的α_{min} （对于主求解和后处理均如此）会得到更精确的解。然而，步长过小会导致 S出现（近似）奇异矩阵，求逆时会发生数值不稳定性。