哈爾濱工業大學

组合优化与凸优化

实验报告

题　　目	典型的优化方法及其实现
学　　院	计算学部
专　　业	电子信息
学　　号	24S103184
学　　生	胡冠宇
任 课 教 师	刘绍辉

哈尔滨工业大学计算学部

2025.3

一、实验内容

本实验将进行如下工作：

第一，介绍各种典型的优化方法。将系统地梳理优化算法的基本分类与原理，涵盖从无需梯度信息的无导数优化方法到依赖梯度信息的有导数优化方法。在一阶优化器部分，将重点讲解基于梯度信息的优化策略，包括最基本的梯度下降法、适用于非光滑目标函数的次梯度下降法，以及加速收敛的共轭方向法、共轭梯度法和采用迭代更新 Hessian 矩阵近似的变尺度法（如 BFGS 法）。二阶优化器则引入目标函数的 Hessian 矩阵以提升收敛速度，典型方法包括 Newton 法与其改进形式阻尼 Newton 法，后者通过调整 Hessian 矩阵提升算法稳定性。此外，还将介绍交替方向乘子法（ADMM）这种适用于分布式和约束优化问题的有效算法，以及 Krylov 子空间方法等在高维大规模问题中应用广泛的迭代式优化策略。

第二，实现上述优化方法。利用 Python 编程语言实现所介绍的各类优化算法。为了确保每种方法具有通用性和可扩展性，将使用 PyTorch 框架对各种优化器进行实现，继承 Optimizer 接口，便于后续在不同类型的目标函数上进行测试和比较。

第三，测试上述优化方法。使用一个典型的测试函数对各优化器进行性能评估——Rosenbrock 函数。实验将对每种优化器在在相同初始点下的收敛过程进行可视化分析，并对优化器性能进行比较。

二、优化方法介绍

设优化问题为

$$\min_{\boldsymbol{x}} \ f(\boldsymbol{x}), \tag{1}$$

其中 $f : \mathbb{R}^n \to \mathbb{R}$。下面将对公式 (1) 讨论各种典型的优化方法。

2.1 随机搜索法

随机搜索法（Random Search Method）是一种无导数优化方法，其基本思想是在定义域内随机生成若干点，并选择函数值最小的作为当前最优解。在第 k 次迭代中，随机生成点 $\boldsymbol{x}_k' \in U(\boldsymbol{x}_k)$，其中 $U(\boldsymbol{x}_k)$ 为 $\boldsymbol{x}_k$ 的某一邻域。

则有迭代公式

$$\boldsymbol{x}_{k+1} = \begin{cases} \boldsymbol{x}_k', & f(\boldsymbol{x}_k') < f(\boldsymbol{x}_k), \\ \boldsymbol{x}_k, & f(\boldsymbol{x}_k') \geq f(\boldsymbol{x}_k)。 \end{cases}$$

该方法通过不断试探更新寻找函数值更小的位置，而无需任何梯度信息。

2.2 梯度下降法

梯度下降法（Gradient Descent Method）利用目标函数的梯度信息，以梯度的负方向作为下降方向，即

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \alpha_k \nabla f(\boldsymbol{x}_k),$$

其中 $\alpha_k > 0$ 为学习率，$\nabla f(\boldsymbol{x}_k)$ 为函数 f 在点 $\boldsymbol{x}_k$ 处的梯度。

2.3 次梯度下降法

次梯度下降法（Subgradient Descent Method）用于优化非光滑凸函数。若 f 在点 $\boldsymbol{x}_k$ 处不可导，但存在次梯度 $\boldsymbol{g}_k \in \partial f(\boldsymbol{x}_k)$，则有

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \alpha_k \boldsymbol{g}_k,$$

其中 $\alpha_k > 0$ 为学习率，$\partial f(\boldsymbol{x}_k)$ 为函数 f 在点 $\boldsymbol{x}_k$ 处的次梯度集合，即

$$\partial f(\boldsymbol{x}_k) = \left\{ \boldsymbol{g}_k \,\middle|\, f(\boldsymbol{x}_k + \boldsymbol{\epsilon}_k) - f(\boldsymbol{x}_k) \geq \boldsymbol{g}_k^{\mathrm{T}} \boldsymbol{\epsilon}_k,\ \forall \boldsymbol{\epsilon}_k \in \mathbb{R}^n \right\}。$$

2.4 共轭方向法

共轭方向法（Conjugate Direction Method）用于二次函数

$$f(\boldsymbol{x}) = \frac{1}{2} \boldsymbol{x}^{\mathrm{T}} \boldsymbol{A} \boldsymbol{x} - \boldsymbol{b}^{\mathrm{T}} \boldsymbol{x},$$

其中矩阵 $\boldsymbol{A}$ 对称正定。

共轭方向法需要构造一组关于矩阵 $\boldsymbol{A}$ 的共轭方向 $\{\boldsymbol{d}_i\}_{i=1}^n$，使得

$$\boldsymbol{d}_i\boldsymbol{A}\boldsymbol{d}_j = 0，\ i \neq j，$$

沿着这些方向进行线性搜索，即

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \alpha_k\boldsymbol{d}_k，$$

其中 $\alpha_k > 0$ 为学习率，可通过一维搜索得到。

对于一般优化问题 (如公式 (1))，使用如下公式迭代共轭方向：

$$\boldsymbol{d}_{k+1} = -\nabla f(\boldsymbol{x}_{k+1}) + \beta_k\boldsymbol{d}_k，\ \boldsymbol{d}_0 = -\nabla f(\boldsymbol{x}_0)。$$

β_k 的迭代有 FR 法、PR 法等，其中 PR 法较为常用。对于 FR 法，有

$$\beta_k = \frac{\|\nabla f(\boldsymbol{x}_{k+1})\|^2}{\|\nabla f(\boldsymbol{x}_k)\|^2}。$$

对于 PR 法，有

$$\beta_k = \frac{[\nabla f(\boldsymbol{x}_{k+1})]^{\mathrm{T}}[\nabla f(\boldsymbol{x}_{k+1}) - \nabla f(\boldsymbol{x}_k)]}{\|\nabla f(\boldsymbol{x}_k)\|^2}。$$

2.5 共轭梯度法

共轭梯度法（Conjugate Gradient Method）是一种特殊的共轭方向法，使用当前梯度和上一次方向构造共轭方向，避免计算 $\boldsymbol{A}$。对于二次函数，其迭代格式为：

$$\begin{aligned}
&\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \alpha_k\boldsymbol{d}_k，\\
&\alpha_k = \frac{\boldsymbol{r}_k^{\mathrm{T}}\boldsymbol{r}_k}{\boldsymbol{d}_k^{\mathrm{T}}\boldsymbol{A}\boldsymbol{d}_k}，\\
&\boldsymbol{d}_k = \boldsymbol{r}_k + \beta_{k-1}\boldsymbol{d}_{k-1}，\ \beta_{k-1} = \frac{\boldsymbol{r}_k^{\mathrm{T}}\boldsymbol{r}_k}{\boldsymbol{r}_{k-1}^{\mathrm{T}}\boldsymbol{r}_{k-1}}，\\
&\boldsymbol{r}_k = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_k，\ \boldsymbol{r}_0 = -\nabla f(\boldsymbol{x}_0)。
\end{aligned}$$

2.6 变尺度法

变尺度法（Quasi-Newton Method）利用对称正定矩阵 $\boldsymbol{B}_k$ 近似 Hessian 矩阵，从而加速收敛。

变尺度法的迭代公式为

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \alpha_k \boldsymbol{H}_k \nabla f(\boldsymbol{x}_k)，\ \boldsymbol{H}_k = \boldsymbol{B}_k^{-1}。$$

为了构造拟 Newton 条件，

$$\boldsymbol{B}_{k+1}\boldsymbol{s}_k = \boldsymbol{y}_k，$$

还需要两个迭代公式，即

$$\boldsymbol{s}_k = \boldsymbol{x}_{k+1} - \boldsymbol{x}_k，$$
$$\boldsymbol{y}_k = \nabla f(\boldsymbol{x}_{k+1}) - f(\boldsymbol{x}_k)。$$

在变尺度法中，对 $\boldsymbol{H}_k$ 的迭代有多种方法，例如 DFP 法和 BFGS 法。DFP 法的迭代公式为

$$\boldsymbol{H}_{k+1} = \boldsymbol{H}_k + \frac{\boldsymbol{s}_k \boldsymbol{s}_k^{\mathrm{T}}}{\boldsymbol{s}_k^{\mathrm{T}} \boldsymbol{y}_k} - \frac{\boldsymbol{H}_k \boldsymbol{y}_k \boldsymbol{y}_k^{\mathrm{T}} \boldsymbol{H}_k}{\boldsymbol{y}_k^{\mathrm{T}} \boldsymbol{H}_k \boldsymbol{y}_k}。$$

BFGS 法的迭代公式为

$$\boldsymbol{H}_{k+1} = (\boldsymbol{I} - \rho_k \boldsymbol{s}_k \boldsymbol{y}_k^{\mathrm{T}}) \boldsymbol{H}_k (\boldsymbol{I} - \rho \boldsymbol{y}_k \boldsymbol{s}_k^{\mathrm{T}}) + \rho_k \boldsymbol{s}_k \boldsymbol{s}_k^{\mathrm{T}}，$$

其中 $\boldsymbol{I}$ 为单位矩阵。BFGS 法相比 DFP 法更加稳定，实用性更强，是最常用的变尺度方法之一。

2.7 随机梯度下降法

随机梯度下降法（Stochastic Gradient Descent Method，SGD）在机器学习中常用于大规模样本问题。设

$$f(\boldsymbol{x}) = \frac{1}{N} \sum_{i=1}^{N} f_i(\boldsymbol{x})，$$

SGD 随机选取一个样本或小批次更新，即

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \alpha_k \nabla f_{i_k}(\boldsymbol{x}_k)，$$

其中 $\alpha_k > 0$ 为学习率，i_k 为当前随机选取的样本索引。

2.8 Newton 法

Newton 法使用函数的二阶导数信息，迭代公式为

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \left[\nabla^2 f(\boldsymbol{x}_k)\right]^{\mathrm{T}} \nabla f(\boldsymbol{x}_k),$$

其中 $\nabla^2 f(\boldsymbol{x}_k)$ 为函数 f 在点 $\boldsymbol{x}_k$ 的 Hessian 矩阵。

2.9 阻尼 Newton 法

为了提升稳定性，阻尼 Newton 法在 Newton 的基础上引入阻尼因子，迭代公式为

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \alpha_k \left[\nabla^2 f(\boldsymbol{x}_k)\right]^{\mathrm{T}} \nabla f(\boldsymbol{x}_k),$$

其中 α_k 为阻尼因子，以保证下降并避免震荡。

2.10 交替方向乘子法

交替方向乘子法（Alternating Direction Method of Multipliers，ADMM）用于分解优化具有分离结构的约束问题

$$\begin{cases} \min\limits_{\boldsymbol{x},\boldsymbol{y}} & f(\boldsymbol{x}) + g(\boldsymbol{y}) \\ \text{s.t.} & \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{y} = \boldsymbol{c}。 \end{cases}$$

其 Lagrange 函数为

$$\mathcal{L}(\boldsymbol{x}，\ \boldsymbol{y}，\ \boldsymbol{z}) = f(\boldsymbol{x}) + g(\boldsymbol{y}) + \boldsymbol{z}\mathrm{T}(\boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{y} - \boldsymbol{c}) + \frac{\rho}{2}\|\boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{y} - \boldsymbol{c}\|^2, \quad (2)$$

因此优化目标转化为公式 (2) 。

2.11 Krylov 子空间法

Krylov 子空间法用于大规模线性系统或优化问题的迭代方法，基本思想是利用前几次残差张成的 Krylov 子空间

$$\mathcal{K}_k(\boldsymbol{A}, \boldsymbol{r}_0) = \mathrm{span}\{\boldsymbol{A}^i \boldsymbol{r}_0\}_{i=0}^{k-1},$$

在子空间中寻找最优解近似，如共轭梯度法 (CG) 就是 Krylov 子空间方法在对称正定情形下的实现。非对称情形中可使用 GMRES 方法。

三、实验设置及结果

为了使优化器的通用性和可扩展性，实验基于 PyTorch 的 Optimizer 类进行优化器实现，并分别实现了随机搜索优化器，梯度下降优化器，次梯度下降优化器，共轭方向优化器，共轭梯度优化器，变尺度法优化器（BFGS），随机梯度下降优化器，Newton 优化器，阻尼 Newton 优化器，ADMM 优化器和 Krylov 子空间优化器，并使用 Rosenbrock Banana 函数作为优化器测试函数，即

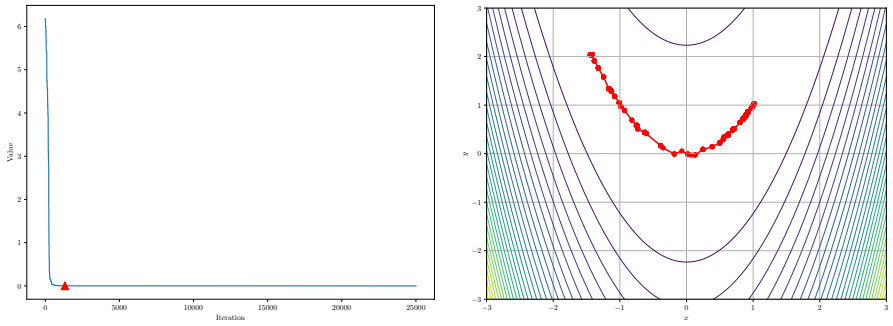
$$f(x_1, x_2) = (a - x_1)^2 - b(x_2 - x_1^2)^2,$$

其中 $a = 1$，$b = -100$。实验结果如表 1 所示，各优化器的最优值收敛曲线和最优点收敛路径如图 1 至图 11 所示。

表 1　实验结果

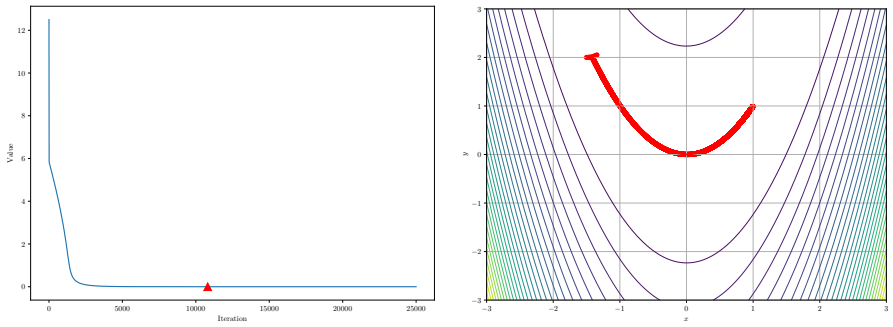
优化器	最优点	最优值	迭代轮次
Random Search	$(1.00, 1.00)$	0.00	1326
Gradient Descent	$(0.99, 1.00)$	0.00	10808
Subgradient Descent	$(0.99, 1.00)$	0.00	10808
Conjugate Direction	$(1.00, 1.00)$	0.00	23366
Conjugate Gradient	$(1.00, 1.00)$	0.00	500
BFGS	$(1.00, 1.00)$	0.00	21983
SGD	$(1.00, 1.00)$	0.00	6350
Newton	$(1.00, 1.00)$	0.00	7
Damped Newton	$(1.00, 1.00)$	0.00	2970
ADMM	$(1.00, 1.00)$	0.00	22184
Krylov	$(1.00, 1.00)$	0.00	6385

从各图可看出，各优化器均在较少的迭代轮次内下降至最优值附近，但各优化器迭代至最优点需要的迭代轮次总和不同。综合表 1 可知，Newton 法所需要的迭代轮次最少。



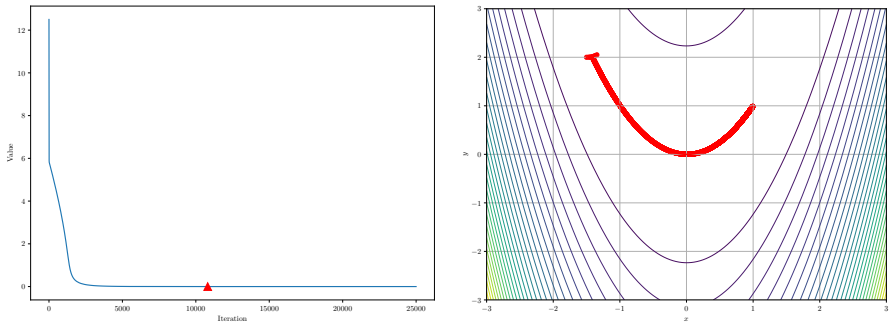
(a) 最优值收敛曲线　　(b) 最优点收敛路径

图 1　Random Search 的最优值收敛曲线与最优点收敛路径



(a) 最优值收敛曲线　　(b) 最优点收敛路径

图 2　Gradient Descent 的最优值收敛曲线与最优点收敛路径

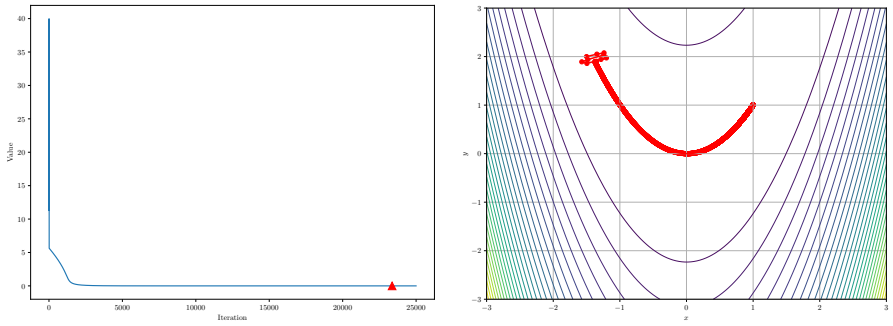


(a) 最优值收敛曲线　　(b) 最优点收敛路径

图 3　Subgradient Descent 的最优值收敛曲线与最优点收敛路径

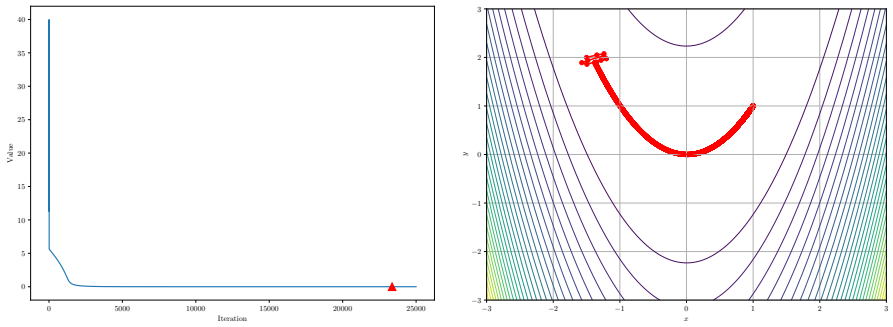
鉴于 Random Search 优化器对初始随机种子较为敏感，实验对多个随机种子进行了实验比较，结果显示使用种子 42 时所需迭代轮次较少，因此在实验中固定采用该种子。

从图 1 中可看出，Random Search 优化器也在较少的迭代轮次下收敛到最优点。



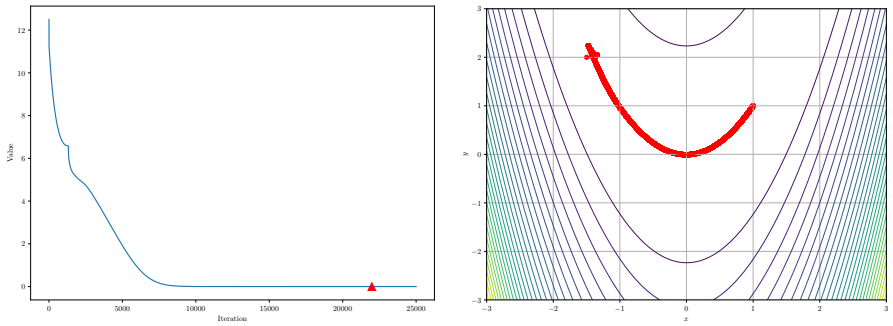
(a) 最优值收敛曲线　　(b) 最优点收敛路径

图 4　Conjugate Direction 的最优值收敛曲线与最优点收敛路径



(a) 最优值收敛曲线　　(b) 最优点收敛路径

图 5　Conjugate Gradient 的最优值收敛曲线与最优点收敛路径

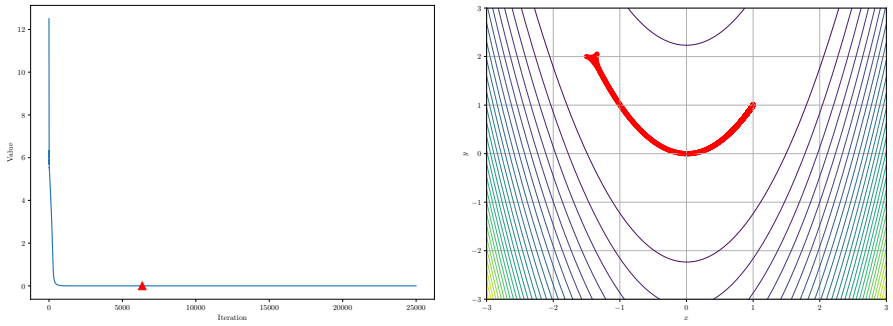


(a) 最优值收敛曲线　　(b) 最优点收敛路径

图 6　BFGS 的最优值收敛曲线与最优点收敛路径

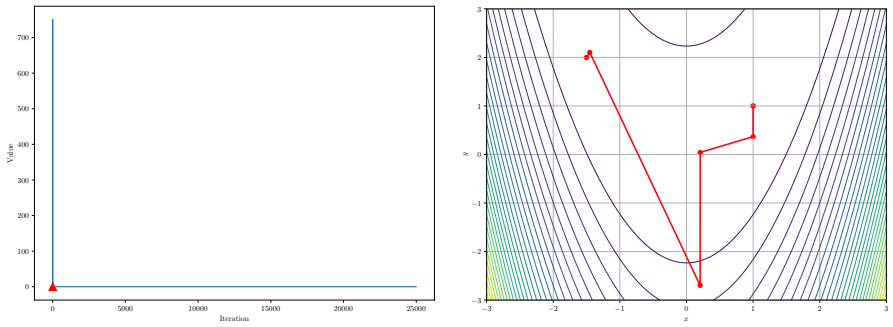
四、结论

本次实验系统地梳理了多种优化方法的原理与特性，并通过 PyTorch 框架实现了这些算法。使用 Rosenbrock 函数对各法进行了性能评估，从图 1 至图 11 与表 1 中可以深入分析各法的表现。



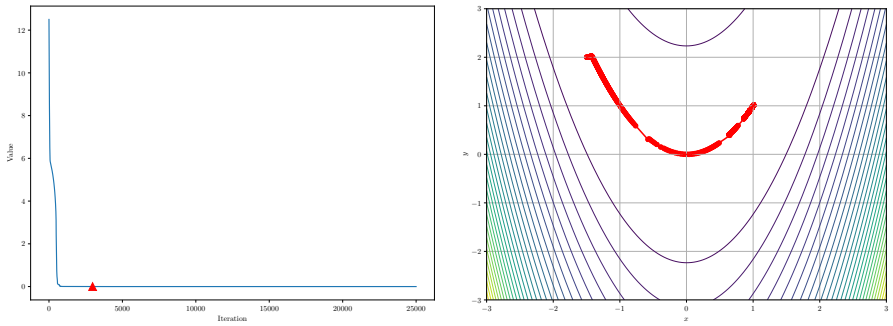
(a) 最优值收敛曲线　　(b) 最优点收敛路径

图 7　SGD 的最优值收敛曲线与最优点收敛路径



(a) 最优值收敛曲线　　(b) 最优点收敛路径

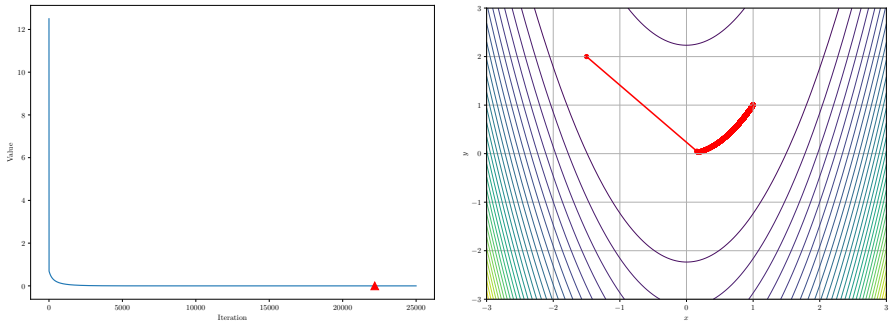
图 8　Newton 的最优值收敛曲线与最优点收敛路径



(a) 最优值收敛曲线　　(b) 最优点收敛路径

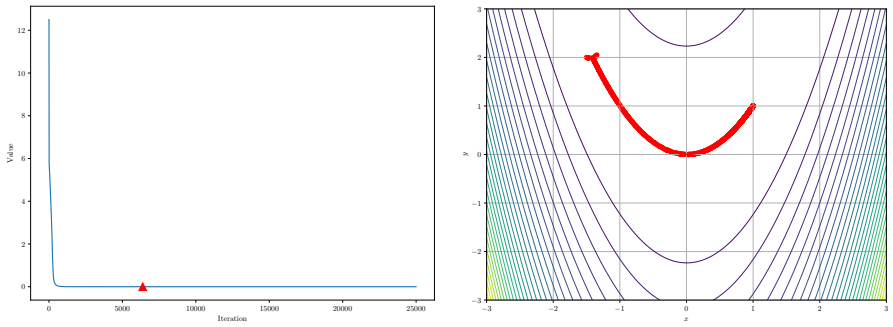
图 9　Damped Newton 的最优值收敛曲线与最优点收敛路径

从各法的最优值收敛曲线来看，大部分方法在迭代初期能够迅速降低目标函数值。然而，具体收敛速度随着时间推移呈现出明显差异。例如随机搜索法在初期表现较为波动，但随着迭代次数增加，其目标函数值逐渐稳定并接近最优值。而梯度下降法与次梯度下降法在初期的收敛速度相对缓慢，但随着迭代次数增加，逐渐展现出稳定的下降趋势，最终逼近最优解。



(a) 最优值收敛曲线　　(b) 最优点收敛路径

图 10　ADMM 的最优值收敛曲线与最优点收敛路径



(a) 最优值收敛曲线　　(b) 最优点收敛路径

图 11　Krylov 的最优值收敛曲线与最优点收敛路径

在最优点收敛路径方面，不同法呈现出独特的路径特征。共轭方向法与共轭梯度法表现出较为直接的收敛路径，能够在较少的迭代步骤中找到最优解附近的位置。相较于一阶方法，BFGS 法通过近似二阶信息，在迭代过程中展现出更快的收敛速度，其路径也更为平滑。

从表 1 中可以看出，Newton 法在所有方法中所需的迭代轮次最少，这得益于其充分利用了二阶导数信息，从而实现对目标函数的二次逼近，展现出极高的收敛效率。阻尼 Newton 法虽然在一定程度上增加了计算复杂度，但通过阻尼因子的引入，有效避免了 Newton 法可能出现的发散问题，同时保持了较快的收敛速度。

在 SGD 法的表现中，我们可以观察到其在大规模样本问题中的潜力。虽然在本次实验中使用的是固定的目标函数，但 SGD 通过对随机样本的更新策略，在迭代过程中依然能够逐步接近最优解，显示出其在处理大规模数据集时的适用性。

ADMM 法主要用于分布式优化和带有约束条件的问题。从实验结果来看，ADMM 在处理具有分离结构的优化问题时，能够有效地分解原问题，通过交替更新变量和乘子，最终实现对原问题的求解。其收敛路径表现出一定的波动性，但在迭代过程中仍能逐步逼近最优解。

从整体的实验结果来看，各法均能够有效地在一定迭代次数内将目标函数值降至接近最优值的水平。但从收敛速度和迭代轮次的角度分析，Newton 法及其改进形式（阻尼 Newton 法）展现出明显的优势。然而，这种优势的取得是以计算二阶导数信息为代价的，因此在实际应用中需要权衡计算复杂度和收敛速度之间的关系。

在面对高维大规模问题时，Krylov 子空间方法等迭代策略展现出了其独特的优势。这种基于子空间投影的优化策略，能够在较低的计算成本下逐步逼近最优解，为高维问题的求解提供了有效的途径。从随机搜索法的表现中，我们也可以得到一些启发。尽管其收敛速度相对较慢，但在某些特定场景下，例如目标函数难以求导或导数信息不可靠时，随机搜索法作为一种无需梯度信息的优化方法，仍具有其独特的应用价值。