

本科毕业论文（设计）

非线性方程组求解的数据驱动

群体智能算法

DATA-DRIVEN EVOLUTION ALGORITHMS FOR
SOLVING NON-LINEAR SYSTEMS OF
EQUATIONS

赵晨君

哈尔滨工业大学

2024 年 5 月

密级：公开

本科毕业论文（设计）

非线性方程组求解的数据驱动
群体智能算法

本 科 生：赵晨君

学 号：200810428

指 导 教 师：张新明

专 业：数据科学与大数据技术

学 院：深圳校区理学院

答 辩 日 期：2024 年 5 月

学 校：哈尔滨工业大学

摘 要

非线性方程组的求解是科学研究以及工程领域重要的研究课题，本文针对非线性方程组的求解场景，将数据驱动的优化算法引入非线性方程组的求解场景中，通过调整、改进模型的结构，优化数据生成算法，提出了 RDDEA-ZPS 算法，该算法在若干算例上呈现出优秀的精确度与性能。

首先，通过对数据驱动优化算法（DDEA），尤其是 WDDEA-DBC 算法进行了分析，指出了其数据生成、代理模型以及优化流程中的缺点，阐述了其在非线性方程组求解场景下的局限性。

此后，针对 WDDEA-DBC 模型的局限性，依次提出了 ZPS 数据扩充算法、逆转 RBFNN 代理模型框架和变更 FPA 优化流程场景等方法，精准解决原算法存在的若干引入误差的问题，提出了逆算子零点采样数据驱动优化算法，即 RDDEA-ZPS 模型。

针对新的模型与流程，在若干算例上进行了数值实验，搜索最优参数。通过实验得到了聚类中心个数、ZPS 数据扩充比例、以及代理模型数量 T 和数据生成波动范围这四个对于算法的性能和效果具有重要影响的参数取值。并通过绘制曲线图、箱线图、热力图的可视化方法，计算均值、方差、置信区间等数值方案，为参数的选择提供了数据支撑。

最终，使用两种算法，WDDEA-DBC 和 RDDEA-ZPS，对相同的非线性方程组算例进行求解，通过对比真解和数值解的欧式距离，计算误差的均值与方差，并进行消融实验，直观地验证了 RDDEA-ZPS 算法求解非线性方程组的精确性与稳定性，证明了改进后的模型框架的有效性。

关键词：数据驱动的优化算法；群体智能算法；非线性方程组；径向基函数网络；数据扩充算法

Abstract

The solution of nonlinear systems of equations is a significant research topic in both scientific and engineering domains. This paper addresses the scenario of solving nonlinear systems of equations by introducing data-driven optimization algorithms. Through adjustments and improvements in model structure and data generation algorithms, the RDDEA-ZPS algorithm is proposed, demonstrating excellent accuracy and performance across various test cases.

Initially, an analysis of data-driven optimization algorithms, particularly the WDDEA-DBC algorithm, is conducted, highlighting drawbacks in data generation, proxy model, and optimization processes, elucidating their limitations in solving nonlinear systems of equations.

Subsequently, in response to the limitations of the WDDEA-DBC model, methods such as the ZPS data augmentation algorithm, the inverted RBFNN proxy model framework, and the modified FPA optimization process are proposed to address several error introduction issues of the original algorithm. This culminates in the development of the inverse operator zero-point sampling data-driven optimization algorithm, namely the RDDEA-ZPS model.

For the new model and processes, numerical experiments are conducted on various test cases to search for optimal parameters. By visualizing results through curve plots, box plots, and heatmaps, and by calculating metrics such as mean, variance, and confidence intervals, empirical support is provided for parameter selection.

Finally, employing both algorithms, WDDEA-DBC and RDDEA-ZPS, to solve the same nonlinear system of equations, the accuracy and stability of the RDDEA-ZPS algorithm in solving nonlinear systems of equations are intuitively validated by comparing the Euclidean distance between true and numerical solutions and calculating the mean and variance of errors, thereby proving the effectiveness of the improved model framework.

Keywords: data-driven evolution algorithm, swarm intelligence algorithms, nonlinear systems of equations, radial basis function network, data augmentation algorithm

目 录

摘 要	I
ABSTRACT	II
第 1 章 绪 论	1
1.1 课题背景	1
1.1.1 非线性方程组	1
1.1.2 非线性方程组的求解方法	1
1.1.3 WDDEA-DBC 算法	2
1.2 研究目的及意义	2
1.3 国内外发展现状	3
1.3.1 非线性方程组求解研究现状	4
1.3.2 基于神经网络求解非线性方程组研究现状	4
1.3.3 数据驱动的优化算法（DDEA）研究现状	5
1.4 本文主要研究内容	6
1.4.1 数据生成方法的研究	6
1.4.2 代理模型的神经网络架构的搭建	6
1.4.3 优化问题的建立	6
1.4.4 非线性方程组的求解	7
第 2 章 WDDEA-DBC 的分析应用	8
2.1 WDDEA-DBC 算法介绍	8
2.1.1 模型结构	8
2.1.2 算法的创新之处	9
2.2 在非线性方程求解场景的应用与扩展	9
2.2.1 数据采样	9
2.2.2 代理模型的建立与优化	10
2.3 WDDEA-DBC 的缺点与局限性	10
2.4 本章小结	11
第 3 章 RDDEA-ZPS 模型建立	12
3.1 代理模型的建立	12
3.1.1 WDDEA-DBC 算法的代理模型	12
3.1.2 修改后的代理模型	13

3.2 数据生成	14
3.2.1 拉丁超立方采样	14
3.2.2 DBC 采样	15
3.2.3 零点采样	16
3.2.4 数据扩充算法流程与结果	17
3.3 代理模型的优化与集成	19
3.3.1 优化算法的转换	19
3.3.2 优化算法的实施	20
3.3.3 模型集成	21
3.4 本章小结	21
第 4 章 数值试验与结果	23
4.1 参数寻优实验	23
4.1.1 RDDEA-ZPS 参数与算例方程组	23
4.1.2 聚类中心数量	24
4.1.3 ZPS 数据扩充比例	26
4.1.4 代理模型数量	28
4.1.5 ZPS 扩充数据的波动范围以及比例组合	29
4.2 模型求解结果对比	31
4.2.1 误差对比	31
4.2.2 求解结果对比	31
4.3 消融实验	33
4.3.1 模型结构实验	33
4.3.2 数据采样实验	34
4.4 本章小结	36
结 论	38
参考文献	40
哈尔滨工业大学本科毕业论文（设计）原创性声明和使用权限	41
致 谢	42

第 1 章 绪 论

1.1 课题背景

1.1.1 非线性方程组

非线性方程组的求解一直是数学界和工程领域的一个重要研究课题。在实际问题中，许多系统的行为往往由非线性方程组描述，例如在工程领域，如控制系统设计、信号处理和优化问题等方面，对非线性方程组的高效求解有着重要的应用价值。而这些方程组往往难以直接求解得到精确解析解。因此，寻找以较小的计算成本和较高的求解精度获取非线性方程组解的方法成为迫切需求。因此，开发出能够在实践中有效处理非线性方程组的求解算法对于提高工程系统的性能和效率具有重要意义。

具体地，一般非线性方程组的结构如式（1-1）所示。

$$\begin{cases} f_1(x_1, x_2, \dots, x_n) = 0 \\ f_2(x_1, x_2, \dots, x_n) = 0 \\ \dots \\ f_n(x_1, x_2, \dots, x_n) = 0 \end{cases} \quad (1-1)$$

其中自变量的数量和方程的数量相等，并且对于不同的输入向量 (x_1, x_2, \dots, x_n) ，方程组内的每个方程都对应一个值。因此方程组的寻优过程即是通过寻找最佳向量 $(x_1^*, x_2^*, \dots, x_n^*)$ ，使得输入这些自变量分量，每个等式右侧得到的值都更接近 0。后续模型建立过程也是基于此前提基础上。

1.1.2 非线性方程组的求解方法

非线性方程组的求解在数学和工程领域中具有重要意义，然而，对于复杂的方程组，其解析求解常常困难或不可行。因此，研究者们转向数值方法来逼近方程组的解。

迭代法是解非线性方程组的一种常见方法。其中，牛顿法利用函数的一阶和二阶导数信息进行迭代，通过不断改进当前解来逼近方程组的解。然而，对于复杂函数或高维问题，牛顿法可能收敛速度较慢，且需要大量计算。拟牛顿法则通过近似 Hessian 矩阵或其逆来替代牛顿法中的精确计算，从而减少计算

量，但在高维问题中可能出现性能下降的情况。

除了传统方法以外，不同的优化方法也可以用于解非线性方程组。例如梯度下降法利用负梯度方向进行搜索，以最小化目标函数，从而求解非线性方程组。智能优化方法^[1]也是求解非线性方程组的方法，基于启发式搜索和优化原理的方法，该类算法通常不依赖于方程组的解析形式，而是通过搜索整个解空间来寻找最优解或近似解。但该类方法多存在收敛性不确定、局部搜索能力不强等问题。

因此，提出一种时间开销小，求解精度高的求解方法对于非线性方程组问题的解决有较大意义。

1.1.3 WDDEA-DBC 算法

数据驱动的优化算法^[2]（Data- Driven Evolutionary Algorithm, DDEA）是一类利用问题领域内已有的数据来指导优化过程的方法。该类方法常常使用不同的数据生成方法进行更多更准确数据的生成，提高寻优效率。在优化过程中，DDEA 算法常常使用群体智能进化算法进行优化。

DDEA 算法通过收集和分析输入输出数据，以更智能、高效的方式搜索和优化问题的解，能够有效提升问题的求解效率以及准确度。

其中 WDDEA-DBC^[3]模型是一种优秀的 DDEA 算法。它由数据生成、代理模型的训练、代理模型的管理与聚合以及优化算法优化等步骤组成。首先它使用聚类方法检测需要生成数据的类别并生成数据，此后以径向基函数网络为代理模型，使用独特高效的模型管理框架聚合若干代理模型，最后使用智能优化算法进行聚合代理模型的优化。

通过将复杂的优化问题转化为代理模型的训练，WDDEA-DBC 算法进一步简化了求解问题的场景，极大的提高了问题的求解效率。这进一步启发我们，能够将 DDEA 算法应用到非线性方程组的求解场所，预期提高解方程的准确度与效率。

但是受限于 WDDEA-DBC 模型的代理模型结构以及优化算法应用的场景，其对于非线性方程组的求解步骤引入了若干误差，导致解的精度不够高。因此如何合理的改进该模型，使得其更适用于非线性方程组求解这一多维优化问题，是值得探讨的问题。

1.2 研究目的及意义

受限于非线性方程组的复杂形式，传统的求解方法常常面临诸多挑战，例

如对初始条件的高度敏感性、易于陷入局部最优解、计算复杂度高以及数值不稳定性等问题，限制了其在实际应用中的效果。鉴此，本课题旨在将数据驱动群体智能算法与非线性方程组的求解结合起来，以期提高算法与问题之间的契合度，从而更高效、更准确地获取方程组的解。

目前，传统优化算法如牛顿法和二分法在初始条件选择上表现出极高的敏感性，并且往往难以在求解精度和计算成本之间达到平衡。与此同时，传统智能优化算法如花授粉算法易于受限于局部最优解，且其收敛速度较慢。为应对这一挑战，一些学者提出了数据驱动的优化算法，即利用历史数据构建代理模型，并采用进化算法来求解优化问题，以实现更直观、直接且快速的非线性方程求解过程。

本课题计划基于 DDEA 算法，尤其是 WDDEA-DBC 模型，将数据驱动的优化算法引入新的问题场景中，对求解非线性方程组的方法进行深入探究。研究方向以及目的包括改进数据生成方法、对网络结构的改造、拓展智能优化算法的应用场景等，以减少学习范数所引入的误差，以优化训练过程。通过这些努力，能够建立一个对于非线性方程组求解具有较高准确性的模型。

1.3 国内外发展现状

非线性方程组的求解一直是数值计算和优化领域的关键研究方向。经典的数值方法，如牛顿法、拟牛顿法和共轭梯度法等，由于其在解决非线性问题方面的有效性和广泛应用而备受重视。此外，全局优化算法如遗传算法、模拟退火算法等也在解决非线性方程组的全局最优问题方面取得了一定成果。近年来，随着深度学习技术的快速发展，其在非线性方程组求解中的应用也逐渐受到关注。例如，利用神经网络等深度学习模型来逼近非线性函数的解，已经成为解决复杂非线性系统问题的有效方法之一。

数据驱动的优化算法是近年来备受关注的研究方向，其核心在于充分挖掘数据背后的信息，利用优化过程中产生的数据进行数据的生成、代理模型的建立、模型的聚合以及智能算法的优化。学术界在探索将数据驱动的思想与传统优化算法相结合的方法，并提出了一系列创新性的研究成果。在应用方面，数据驱动的算法在大数据分析、智能制造等领域展现出了广泛的应用前景。

尽管数值计算和优化领域都取得了一定的进展，但将数据驱动的优化算法应用于非线性方程组的求解仍然是一个待深入探索的领域。

1.3.1 非线性方程组求解研究现状

非线性方程组的求解是数学和计算数学领域的一个关键问题，其目标是找到一组变量的值，使得一组非线性方程被满足。为了解决这一问题，研究者们提出了两大类主要方法，即传统方法和智能优化算法。

传统方法包括牛顿法、拟牛顿法和最速下降法等。这些通常是迭代法^[4,5]，从一个初始猜测值开始，通过迭代更新变量的值，直到满足方程组的条件。传统方法具有简单易实现的特点，并且在数值计算等领域得到了广泛的应用。

近年来，智能优化算法（Intelligent Optimization Algorithms, IOAs）在解决非线性方程组方面表现出了显著的优越性^[1]。尤其是针对具有多个局部最优解的非线性方程组，智能优化算法，例如遗传算法^[6]、花授粉算法^[7]、模拟退火算法等，被广泛应用于寻找全局最优解的任务中。与传统的迭代方法相比，这些算法具有更强的全局搜索能力，能够有效避免陷入局部最优解，并且具备一定的自适应性，能够在解决复杂问题时取得更好的效果。其在非线性方程组求解领域的高效性和可靠性，为复杂问题的求解提供了有力的工具和方法。

1.3.2 基于神经网络求解非线性方程组研究现状

随着神经网络的发展，非线性方程组的求解已经成为了神经网络应用的一个重要场景。为了更好地结合非线性方程组求解问题与数据驱动的优化算法，本课题致力于探索非线性方程组的神经网络解法，以便于后续建立更加高效的数值模型。通过将神经网络技术与优化算法相结合，旨在开发出一种更为精确、稳健且高效的非线性方程组求解方法，为相关领域的研究和应用提供更多的可能性。

李盈晖等在 1997 年提出了用人工神经网络求解非线性方程组的基础架构^[8]，以基础的前馈神经网络为基础，对于非线性方程组 $f(\mathbf{x}) = (f_1(\mathbf{x}), \dots, f_n(\mathbf{x}))^T = 0$ ，使用反向传播（Backpropagation, BP）算法，构建了以向量 $f(\mathbf{x})$ 为输入，以自变量向量 \mathbf{x} 为输出的神经网络。通过对求解区间内生成若干自变量 \mathbf{x} 数据，计算因变量向量 $f(\mathbf{x})$ 对应的值，逆转两组变量，作为训练数据进行训练，在一些非线性方程组上得到了较好的结果，奠定了神经网络求解非线性方程组方法的基础。

此后，赵启林等学者于 2000 年提出了一种新的神经网络架构^[9]，该架构基于耦合神经网络，并通过关联网络的权值，为求解非线性方程组提供了一种全新的网络结构。然而，这种方法对于非线性方程组的形式具有一定的要求，不适用于所有类型的非线性方程组。

2002 年，赵华敏等学者提出了一种独特的神经网络模型，其中包含微分形式^[10]，并通过严格的数学证明推导出该网络可以从任意起始点开始进行迭代，并最终收敛到非线性方程组的一个解。这一发现为神经网络方法寻找更广泛解的范围提供了新的思路和方法。类似地，G. Li 等学者提出了另一种神经网络模型^[11]，同样采用迭代计算的方式进行解的优化。与之前的方法不同的是，该模型省去了复杂的微分和积分运算，从而有效提高了求解效率，减少了计算开销。

基于李盈晖在 1997 年提出的求解非线性方程组的基础神经网络架构，孙银慧等于 2009 年提出了一种求解非线性方程组的迭代神经网络算法^[12]。该算法以基础神经网络架构为基础，建立显式的神经网络结构，并通过迭代方法进行求解。在训练过程中，根据逐步扩大解的存在区间搜索范围的训练结果，使得算法能够在给定的初始区间以外寻找方程组的真实解。

基于该方法，2013 年，赵庆兰等学者提出了改进的迭代神经网络算法，用于解决非线性方程组^[13]。该算法针对前述算法可能出现的无穷迭代等问题，提出了记忆最小误差点和限制无穷迭代的方法，从而显著提升了算法的效果。

1.3.3 数据驱动和优化算法（DDEA）研究现状

Yaochu Jin 等人的研究团队建立了在线 DDEA 和离线 DDEA 的框架，阐明了 DDEA 模型的一般流程^[2]。他们依据这一新颖的优化算法，深入研究了多目标优化、昂贵优化等难度较大的优化问题，并且利用他们的研究成果成功解决了许多实际应用问题，取得了显著的成果。

2019 年，Wang Handing 等人在 DDEA 框架的基础上，首次引入了代理模型的 Bagging 管理策略^[14]，在对数据集的噪声进行处理后，通过 Bootstrap 采样，生成新的高质量数据，此后借助机器学习中的集成学习方法，对代理模型的误差进行修正，进一步提高了优化算法的准确率。

2020 年，黎建宇等学者引入了新型 Boosting 集成学习方法到 DDEA 的应用中，建立了 BDDEA-LDG 模型^[15]。该模型在数据拟合效果欠佳的地方生成新的数据，再通过相比于 Bagging 更加快速的 Boosting 方法进行代理模型的拟合。

郭宗良等创新性地提出使用基于聚类信息的数据生成方法，以 $\frac{1}{MSE}$ 为权重信息的 RBFNN 集成 boosting 框架，使用花授粉算法进行求解，此方法简称为 WDDEA-DBC^[3]。该模型在若干函数优化的过程中表现出强大的性能，但对于非线性方程组求解这一优化问题，仍然有提升的空间。因此这也是本文的主要 Baseline 模型，预期对该模型进行改进，从而达到更好的求解效果。

1.4 本文主要研究内容

1.4.1 数据生成方法的研究

数据的质量对于模型的泛化能力和性能具有直接的影响。一个均匀的数据集能够确保模型在不同类别或特征上都有足够的代表性样本，从而避免了因数据偏斜而导致的模型偏差。对于非线性方程组的求解而言，同样也需要生成分布均匀的自变量和因变量数对。这样的数据集对于后续网络搜索求解方程组的解至关重要，因为它们能够更好地反映方程组的特性，从而提高求解过程的准确性和可靠性。因此，确保数据集的均匀性和代表性是非线性方程组求解中不可忽视的关键步骤。

由于本研究的目的是求解非线性方程组，因此模型需要重点学习接近解部分的数据分布与自变量、因变量的对应关系。如何生成均匀、能够覆盖解区间的数据并同时加强对于重点部分数据的生成，也是本课题的重要研究目标之一。

1.4.2 代理模型的神经网络架构的搭建

对于 DDEA 算法而言，代理模型的选择是至关重要的。在 WDDEA-DBC 模型中，选用了 RBFNN 作为代理模型，用于对非线性方程组范数进行拟合，并通过优化算法对模型进行优化。然而，对于该架构而言，范数的拟合进一步引入了误差，可能导致数据的质量下降。

因此，本文需要重新调整代理模型的架构，找到适合于非线性方程组求解问题的代理模型。这一步骤至关重要，因为代理模型的选择直接影响了模型的性能和求解结果的准确性。通过精心设计和调整代理模型的架构，算法可以更好地适应非线性方程组的特性，从而提高模型的拟合能力和求解效果。

1.4.3 优化问题的建立

在原始的 WDDEA-DBC 中，采用了数据扩充的方式对数据集进行增强，并对原始的 RBFNN 模型进行优化，以获取非线性方程组的解。然而，这一步骤引入了误差，并且在某种程度上等同于直接使用智能优化算法来求解非线性方程组，其效果有待进一步提高。

因此，在改进了代理模型后，优化算法的应用方法也需要作出调整，以实现更好的寻优效果。通过对优化算法的重新设计和调整，我们可以更有效地利用改进后的代理模型，进一步提高非线性方程组求解的精度和效率。

1.4.4 非线性方程组的求解

在模型构建完毕后，最终要得到非线性方程组的解。这需要对模型进行结构的整合，数据的处理，最终使用该模型求解出方程组的解，即根据模型直接得到使得 $f(\mathbf{x}) = (f_1(\mathbf{x}), \dots, f_n(\mathbf{x}))^T = 0$ 的解 \mathbf{x} 。

第 2 章 WDDEA-DBC 的分析应用

2.1 WDDEA-DBC 算法介绍

2.1.1 模型结构

WDDEA-DBC 模型作为改进模型的 Baseline，其数据处理方法和结构具有许多创新之处，可以应用到非线性方程组的求解中。

为了能够将 DDEA 算法成功拓展到非线性方程组求解的场景中，并且在具体的场景中分析模型的优缺点、提出改进方案，本章将简要阐述该 Baseline 模型的流程，并分析其在该场景中的局限性，为后续改进模型的建立提供对比与依据。

WDDEA-DBC 模型主要分为数据采样、DBC 聚类扩充数据、代理模型的训练与聚合以及使用花授粉算法进行优化等几个部分。具体而言，WDDEA-DBC 模型的结构流程如图 2-1 所示。

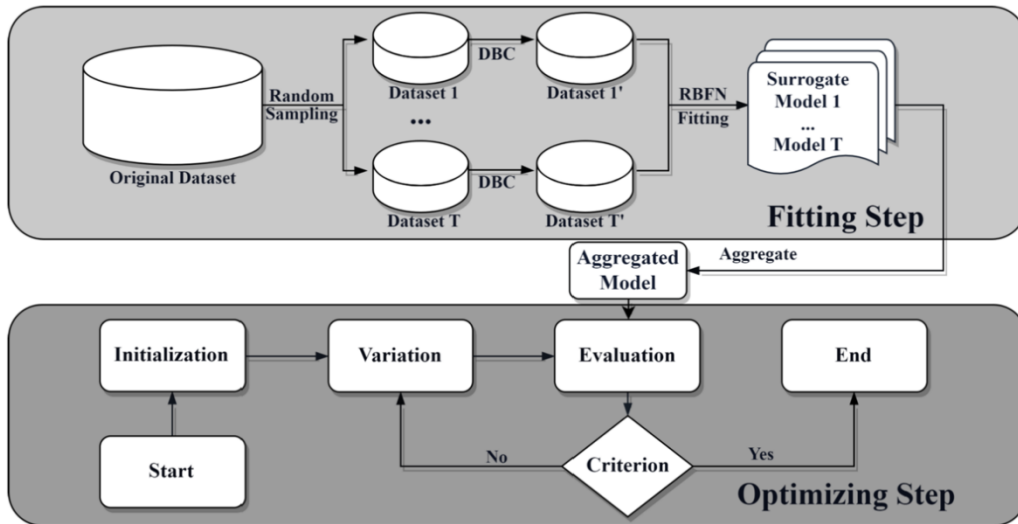


图 2-1 WDDEA-DBC 模型结构^[3]

算法的开始，首先使用拉丁超立方采样（Latin Hypercube Sampling）生成数据定义域中分布较为均匀的数据。此后，再通过基于聚类信息的数据生成^[3]方法（Data Generation Based on Clustering, DBC）来扩充数据集，进一步均匀数据分布、合理扩充数据集。进一步，在扩充后的数据集上进行抽样并训练若干个代理模型，此处的代理模型选择径向基函数网络（Radial basis function neural

network, RBFNN), 各个独立的代理模型采用带 MSE (Mean Squared Error) 权重的管理框架聚合起来。最后, 使用花授粉算法 (Flower Pollination Algorithm, FPA) 对该聚合后的模型进行优化, 实现对原始问题的寻优。

2.1.2 算法的创新之处

WDDEA-DBC 模型的创新之处主要在于其创造性地提出了 DBC 生成数据的算法, 能够在数据空间中薄弱的地方加强数据的质量, 尤其在数据较难获得的场景下扩充数据集十分有效。此外, MSE 倒数加权的方法巧妙削减了误差 (MSE Loss) 较大的模型的预测权重, 从而进一步冲刷掉了大误差结果对最终问题求解的影响。

由于该算法的应用场景极为广泛, 且属于单目标优化算法, 因此将其应用到非线性方程组的求解中, 需要进行模型的调整与构件的合理处理。

2.2 在非线性方程求解场景的应用与扩展

2.2.1 数据采样

非线性方程组的求解是很重要的一类优化问题, 如何合理地将算法进行调整, 使之能够应用于特定场景是需要研究的一类问题。

首先是 WDDEA-DBC 算法中的数据采样部分, 采样的目的是生成足够且均匀的数据集, 为后续代理模型的训练提供高质量数据支持。在非线性方程组的求解场景中, 采样空间为非线性方程组的自变量 \mathbf{x} 空间, 通过对该定义域空间进行拉丁超立方采样, 生成初始的自变量数据集, 通过非线性方程组得到自变量数据集对应的因变量数据集 \mathbf{y} 。此后再对该数据集使用 DBC 算法, 在分布稀疏的地方生成新的数据。如上采样若干次, 得到若干组生成好的数据集, 用于后续若干代理模型的训练。

具体地, 非线性方程组的结构如第一章的公式 (1-1) 所示。首先通过拉丁超立方采样 \mathbf{x} 空间, 再通过 (1-1) 的每个等式计算出对应的右侧值 y_i , 即为第 i 个等式的值, 拼成向量 $\mathbf{y} = (y_1, y_2, \dots, y_n) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_n(\mathbf{x}))$, 作为整体算法的数据集。

由于该算法是单目标优化算法, 在后续优化时, 只能通过搜索多维 \mathbf{x} 空间, 优化单变量的值。因此将右侧的 \mathbf{y} 向量取范数, 即 $F_1(\mathbf{x}) = \|f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_n(\mathbf{x})\|_1$, 将范数值作为适应度进行优化, 其中范数的类型由原算法在测试函数上寻优选定。最小化范数的过程等同于最小化方程组等式右侧值的过程。因此生成数据

时，对 \mathbf{x} 向量空间进行采样，将采样的 \mathbf{x} 代入方程组中，得到 \mathbf{y} 向量的范数。采样后的数据点格式可记作 $\{(\mathbf{x}, F_1(\mathbf{x}))\}$ 。

2.2.2 代理模型的建立与优化

代理模型建立部分，采用 RBFNN 作为代理模型，以 \mathbf{x} 为网络的输入， $F_1(\mathbf{x})$ 为网络的输出，训练得到 $\mathbf{x} \rightarrow F_1(\mathbf{x})$ 的隐式对应关系，网络的优化算法采用常见的 Adam 方法。

在模型聚合部分，经过训练若干组模型后，算法得到了每个独立模型的均方误差（MSE）值。接着，算法将 MSE 的倒数作为权值对模型进行加权，以削弱大误差模型、增强小误差模型的预测结果影响力，得到更加准确的函数对应关系。最终的模型将使用这些加权重值来综合考虑各个模型的性能，以提高整体预测准确性。

在模型构建完成后，算法使用花授粉算法（FPA）进行寻优。具体而言，将函数传入 FPA 算法中，通过全局或局部搜索、自然变异、Levy 飞行等方法来寻找更优的解 \mathbf{x} ，使得 $F_1(\mathbf{x})$ 越来越接近最小值 0，即解逐渐逼近非线性方程组的解 \mathbf{x}^* 。经过如上模型程序的运算，最终可以得到非线性方程组的近似解 \mathbf{x}^* 。

为保持对改进后模型叙述的连贯程度，此处的流程仅做简述，对于应用到的 WDDEA-DBC 算法中的一部分方法，将在后文改进算法中详细阐述并予以备注。

2.3 WDDEA-DBC 的缺点与局限性

原算法在若干优化场景中有较优秀的效果，但若需要应用在非线性方程组上，则会存在若干局限性。

首先，代理模型的训练过程属于近似方程组自变量到因变量映射关系的步骤。相比于直接使用原方程组从左到右（即 \mathbf{x} 到 \mathbf{y} 的范数）的对应关系直接作为 FPA 算法的优化函数，神经网络的引入导致原本精确的映射关系变得模糊且有误差，这是引入的第一步误差。

其次，算法受限于单目标优化场景，引入范数作为神经网络的输出。而范数的引入导致神经网络无法具体学习自变量到因变量的对应关系，这一转换导致一些信息被吞噬，引入了第二步误差。

第三，对于最终生成的模型，算法使用 FPA 算法寻优。加权后的集成模型已经是原方程组对应关系的近似，在这一步误差的基础上又进行了智能优化算法的应用，又进一步引入了第三步优化误差。

此外，理论上讲，非线性方程组的数据可以无穷无尽地获取，而 DBC 数据生成算法在数据源欠缺的情况下效果最佳，因此在该场景下，该数据生成算法似乎作用不大。那么该如何更合理地生成数据，能够产生更多更高质量的数据支持，也是改进模型需要考虑的重要方面。

综合以上问题，后续进行模型的改进以及实验的设计，进一步提升 DDEA 算法在非线性方程组求解问题上的适配性和准确度。

2.4 本章小结

综上所述，本章主要介绍了 WDDEA-DBC 模型作为改进模型的 Baseline，在非线性方程组求解中的应用，并对其进行了详细的分析和讨论。通过分析以 WDDEA-DBC 为代表的 DDEA 算法，可以清楚的看见该类模型的数据处理方法和结构具有许多创新之处，可以被应用到非线性方程组的求解中。

尽管 WDDEA-DBC 模型在若干优化场景中表现出良好的效果，但在应用于非线性方程组求解时存在一些局限性。主要包括代理模型训练误差、单目标优化场景限制、智能优化算法引入误差等方面。此外，DBC 数据生成算法在数据源充足时效果有限，需要更有效的数据生成方法来支持模型的应用和改进。

此后，从第三章开始，将进行改进后模型构建过程的阐述，从数据生成到最终结果均属于改进模型的组成部分，包括对 WDDEA-DBC 模型的延伸与对其缺点的依次修改。

第 3 章 RDDEA-ZPS 模型建立

3.1 代理模型的建立

3.1.1 WDDEA-DBC 算法的代理模型

对于 DDEA 算法，代理模型的选择至关重要，WDDEA-DBC 模型中选用了 RBFNN^[16]作为代理模型进行非线性方程组范数的拟合，并用优化算法进行模型的优化。原算法采用的代理模型结构具体如图 3-1 所示。

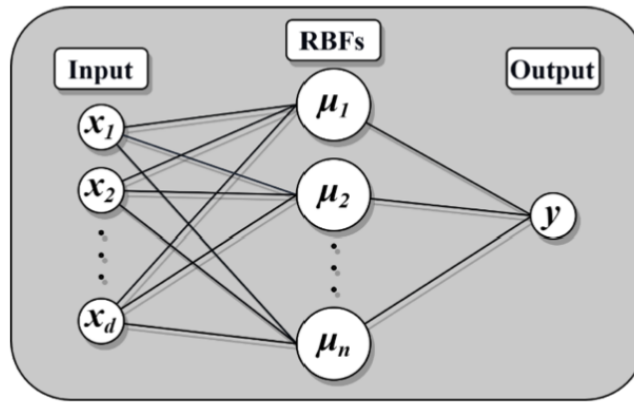


图 3-1 WDDEA-DBC 算法的代理模型结构^[3]

对于该架构，正如上一章所提到的，由于原模型采用花授粉算法（FPA）进行求解，该优化算法是一种单目标优化算法，输出为 y 的范数。然而这样的算法有较大的局限性。

首先，范数的引入导致无法具体学习自变量到因变量的对应关系，这一转换导致一些信息被吞噬，引入了第一步误差。其次，对于最终生成的函数，使用 FPA 算法寻优，函数已经是原方程组的近似，在这一步误差的基础上又进行了智能优化算法的应用，又进一步引入了误差。因此，最终的结果并不是非常理想。

代理模型是整体模型的主要架构，对代理模型的合理修改对整体模型的精度有较大的提升作用。因此考虑修改原算法中的代理模型框架，以提升求解准确度。

3.1.2 修改后的代理模型

针对以上问题，本课题将代理模型的结构进行改进，从拟合非线性方程组的范数转变为拟合方程组的逆算子，即 \mathbf{y} 到 \mathbf{x} 的对应关系。

为了方便对比，控制变量，新的模型仍然使用 RBFNN 作为代理模型基础结构。修改后的 RBFNN 结构如图 3-2 所示。

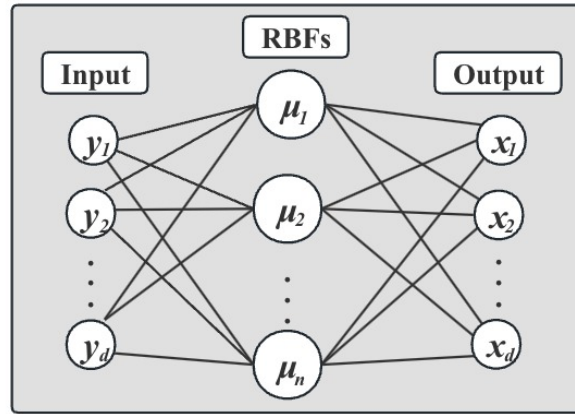


图 3-2 改进模型中代理模型的结构

结构图中，左侧为因变量向量的各个分类，即非线性方程组中的每个等式右侧的值，右侧为因变量对应的自变量向量 \mathbf{x} 的值。该 RBFNN 除输入、输出层以外，中间层为径向基层，从输入层到径向基层的计算公式由式（3-1）所示：

$$\mu_i = \sum_{j=1}^d w_{ji} \phi(\|\mathbf{y} - \mathbf{c}_j\|) \quad (3-1)$$

其中 μ_i 是径向基函数的计算结果，也是中间神经元的激活值； w_{ji} 是输入层神经元到中间层神经元的权重，在使用中常常记为 1； \mathbf{y} 是输入向量， \mathbf{c}_i 是第 i 个神经元的聚类中心。为方便对比效果，本课题采用的径向基函数与原算法一致，为高斯函数，表达式由（3-2）式给出：

$$\phi(x) = \exp\left(\frac{x^2}{-2\sigma^2}\right) \quad (3-2)$$

在具体计算中，式中的 x 即由输入向量 \mathbf{y} 和每个神经元的中心点 \mathbf{c}_i 之间的距离代替。 σ 为控制宽度的参数，由后续数据生成步骤中的中间结果确定。

相比于原代理模型，这里做了两点转变。第一是将数据集的范数还原为向量，这样可以更精确地学习自变量与因变量之间的映射关系，消除引入范数这

一步造成的信息损失。第二是调转了自变量、因变量在神经网络中的输入输出地位，此处把因变量 \mathbf{y} 向量作为输入，自变量 \mathbf{x} 作为输出，进行网络的训练。

经过这样的转变，便可以得到向量 \mathbf{y} 到 \mathbf{x} 的映射关系，在消除范数步骤误差的同时，极大的简化了求解步骤。由于原模型求解非线性方程组时，需要使用RBFNN近似以及FPA寻优，这两个步骤均引入了误差。而修改后的代理模型产生的误差仅仅是拟合神经网络可能引入误差，在求解时，无需使用智能优化算法搜索最优解，而仅需要对训练好的代理模型输入 d 维零向量 $\mathbf{y} = (0, 0, \dots, 0)$ ，输出的解 \mathbf{x}^* 向量即是算法求解结果。

3.2 数据生成

3.2.1 拉丁超立方采样

构建和使用高质量的数据集对于提高模型算法效果至关重要。对于非线性方程组求解这一特殊场景，均匀的数据集能够保证训练集中包含、定位解所在的区间。此外，重点生成解附近的数据集也对模型在解周围寻优精度的提升有较大作用，因此该场景下，如何生成均匀并且有重点的数据集对模型的建立至关重要。

借鉴WDDEA-DBC模型的数据生成方法，首先对定义域空间进行拉丁超立方采样，保证数据的均匀性和广度。

相对于传统的随机抽样方法，拉丁超立方采样可以更有效地利用参数空间，避免了样本之间的相关性，从而更好地探索参数空间的全局特性。

在拉丁超立方采样中，首先将每个参数的取值范围划分成相等的子区间，然后在每个子区间内随机选取一个样本点。这种设计确保了样本点之间的分布均匀且相互独立。

在非线性方程组的求解这一应用场景下，模型将对自变量 $\mathbf{x} = (x_1, x_2, \dots, x_d)$ 的每一维进行采样，此处下标记号变更为 d ，便于后续变量设置的理解。算法流程如下。

算法 3-1 拉丁超立方采样

输入： 维度 d ，采样数据点数量 N_data

输出： 自变量采样后的数据集 \mathbf{X} 。

Begin

For $j=1$ to d **Do**

 将第 j 维度均匀分为 N_data 个区间

For $i = 1$ to N_data **Do**

 在该区间内生成一个随机数，记作 N_{ij}

 将该随机数加入矩阵中

End for

End for

End

最终得到的采样结果为一个矩阵，如式（3-1）所示：

$$\mathbf{X} = \begin{pmatrix} X_{11} & X_{21} & \cdots & X_{N1} \\ X_{12} & X_{22} & \cdots & X_{N2} \\ \vdots & \vdots & \ddots & \vdots \\ X_{1d} & X_{2d} & \cdots & X_{Nd} \end{pmatrix} \quad (3-3)$$

其中每一列为一个数据点 \mathbf{x} ，每一行是不同数据点的同一个维度。这样的方法保证了在每个维度上都进行了均匀且具有代表性的采样，使得采样的结果能够较好地反映参数空间的特性。此后再根据以上的每列的 \mathbf{x} 数据点计算出对应的 \mathbf{y} 向量，得到若干数据对，作为初步的数据集 $\{(\mathbf{x}, \mathbf{y})\} = \{(\mathbf{x}, F_1(\mathbf{x}))\}$ 。

3.2.2 DBC 采样

生成拉丁超立方采样的数据集后，同样受 WDDEA-DBC 算法启发，将进一步使用 DBC 算法对数据进行进一步的生成。DBC 算法为基于聚类信息的数据生成方法，目的是扩充数据集大小，而具体途径是在数据集中数据分布较为稀疏的地方产生合成数据来提高训练效果。

具体来说，拉丁超立方采样后的原数据集可以表示为 $TD = \{(\mathbf{x}_i, \mathbf{y}_i) | i = 1, 2, \dots, N\}$ 。

由于 DBC 的职责是根据 TD 的原始数据生成新的合成数据，因此 DBC 算法流程的首要任务在于确定数据生成的区域范围。该过程涉及调用 K-Means 算法进行聚类，并获得簇及其中心点的信息。

由于在改进后的模型中，作为输入的变为 \mathbf{y} ，因此需要对 \mathbf{y} 进行扩充，使得模型能够学习到充足的 \mathbf{y} 的分布，从而方便进一步寻优。对每个簇，按照其紧密度（Compactness, CP）的大小进行排序。其中，紧密度的计算方法为：

$$CP_j = \sum_{i=1}^{n_j} \frac{\|\mathbf{y}_i - \mathbf{c}_j\|}{n_j}, j = 1, 2, \dots, m \quad (3-4)$$

其中 n_j 是第 j 个簇中数据点的个数， \mathbf{c}_j 是第 j 个簇的中心点。此后会对 CP 值最大的 m 个簇做数据生成，数据生成的具体算法由后续章节给出。

然而，由于目前求解的问题是非线性方程组，调转代理模型输入输出之后，直接对 \mathbf{y} 进行扩充会导致无法计算对应自变量 \mathbf{x} 的情况。因此通过寻找 \mathbf{y} 对应的 \mathbf{x} ，对 \mathbf{x} 做生成，再由新的 \mathbf{x} 计算出对应的 \mathbf{y} 。

DBC 算法的流程由图 3-3 阐述。

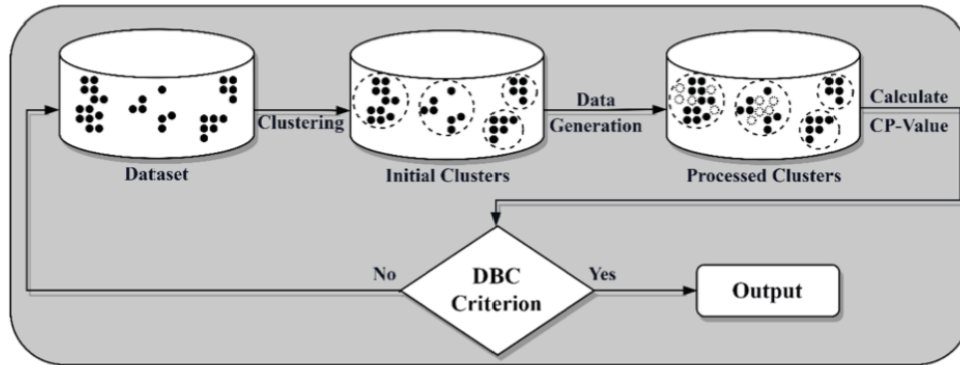


图 3-3 DBC 算法流程^[3]

即对于原数据集进行 K-Means 聚类，计算出 CP 值低的簇，对其进行数据生成，此后再继续检测各个簇的 CP 值，直到达到收敛或者停止条件，DBC 算法停止。

3.2.3 零点采样

DBC 算法在数据源有限的情况下，能生成与原数据相似的数据，提高了数据数量与质量。然而，在求解非线性方程组这个应用场景中，每采样生成一个自变量 \mathbf{x} 都可以获得对应的 \mathbf{y} ，这就能够使得模型获取源源不断的数据，DBC 的优势不再明显。因此，我们需要更高效更有利于模型寻优的数据生成方法。

由于求解非线性方程组的目标为寻找使得每个等式都等于零的 \mathbf{x} 组合，因此对于 \mathbf{y} 接近 $\mathbf{0}$ 附近的数据点，最具有研究价值， \mathbf{y} 与 $\mathbf{0}$ 的接近程度由范数进行衡量。通过对这类数据进行额外采样，称为零点采样（Zero-point Sampling, ZPS）。

具体流程如下。对于 DBC 采样后的数据集，计算出数据集内点的 $\|\mathbf{y}\|$ 并进行排序，选择 $\|\mathbf{y}\|$ 最小的 m 处数据点，对其对应的 \mathbf{x} 进行周边的数据生成。具体地，对选定的该部分数据集 \mathbf{X}_m ，与 DBC 算法采样相同的数据生成方法，即对于每一个 \mathbf{X}_m 中的点 \mathbf{x}_i ，生成其邻域内一个新的数据点 $\mathbf{x}_i + \boldsymbol{\delta}_i$ 。其中 $\boldsymbol{\delta}_i$ 的分布以及参数由（3-5）和（3-6）式给出：

$$\boldsymbol{\delta}_i \sim N_d(\mathbf{0}_d, \mathbf{L}_d), \mathbf{L}_d = l \cdot \mathbf{E}_d \quad (3-5)$$

$$l = \sqrt{\frac{\sum_{j=1}^d (U_j - L_j)^2}{d}} \cdot 10^{-6} \quad (3-6)$$

l 的具体值由文献[15]给出，用于控制生成数据的波动范围，此后也会对 DBC、ZPS 算法中涉及到的参数进行参数寻优实验与讨论。经过数据扩充采样后，新生成的数据集集合为 $K = \{\mathbf{x}_{new}, F(\mathbf{x}_{new}) | \mathbf{x}_{new} = \mathbf{x}_i + \boldsymbol{\delta}_i\}$ ，其中 $F(\cdot)$ 为非线性方程组函数。将原数据与新数据结合后得到最终用于训练的数据集。

3.2.4 数据扩充算法流程与结果

经过 DBC 采样以及 ZPS 数据增强，数据集更加适用于非线性方程组的求解场景。在基础的拉丁超立方采样下，数据扩充算法的总流程描述由算法 3-2 伪代码给出。

选取低维算例进行二维和三维可视化，通过绘制拉丁超立方采样后以及 DBC、ZPS 处理后的数据，将不同的数据分布进行对比，初步检验算法在数据生成以及扩充步骤得到数据集的质量。

具体地，在简单方程组

$$\begin{cases} x_1^2 - x_2^2 = 0 \\ 1 - |x_1 - x_2| = 0 \end{cases} \quad (3-7)$$

进行数据可视化的实验，绘制三维散点图以及点截面图，从而进一步分析数据生成与扩充算法运行后的数据点分布。

算法 3-2 DBC 与 ZPS 数据扩充

输入： TD —原数据集

$N_centers$ —类别数

w —需要生成数据的簇的数量

m —需要生成数据的数量

输出： 生成后的数据集 TD

Begin:

$CPs = KMeans(TD, N_centers)$

选出 CP 值最高的 w 个簇，记为 S

For y_i **in** S **Do**

 找到 y_i 对应的 x_i ，按照 $x_{new} = x_i + \delta_i$ 生成新的数据点并加入 TD 中

 计算出 y_{new} ，将点 (x_{new}, y_{new}) 加入 TD 中

End For

选出 y 范数取值最小的 m 处数据点，记为簇 M

For y_i **in** M **Do**

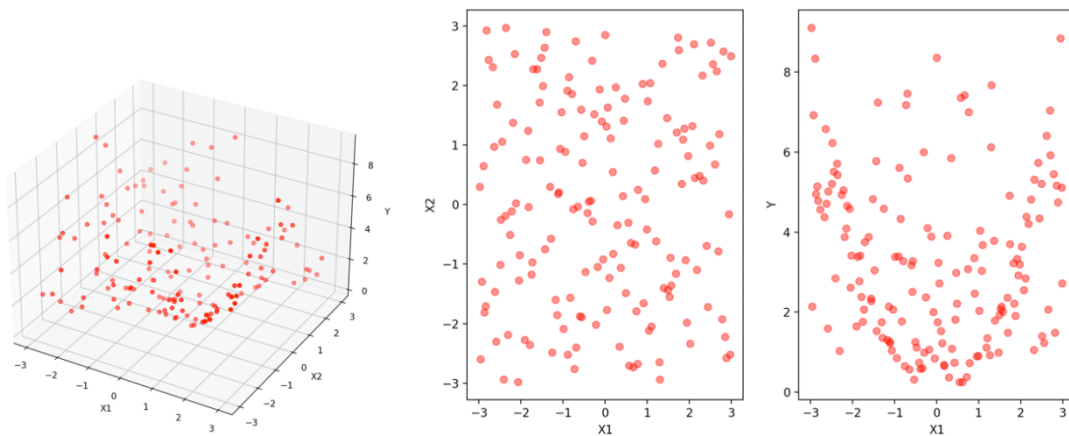
 找到 y_i 对应的 x_i ，按照 $x_{new} = x_i + \delta_i$ 生成新的数据点，并加入 TD 中

 计算出 y_{new} ，将点 (x_{new}, y_{new}) 加入 TD 中

End For

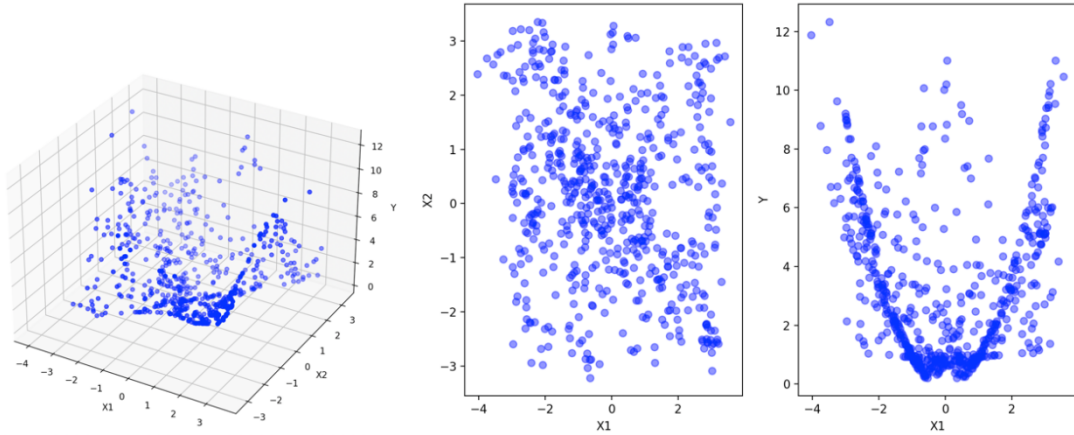
End

得到结果如图 3-4 所示。



(a) 拉丁超立法采样后数据

图 3-4 低维方程数据生成可视化



(b) DBC 与 ZPS 后数据分布

图 3-4（续图）

其中每一行的一组图为进行一个步骤后的采样结果，第一行是仅进行拉丁超立方采样后的数据分布，第二行是经过 DBC 和 ZPS 数据扩充后的数据分布。每行的第一个图为以 $\mathbf{x} = (x_1, x_2)$ 为底部两个坐标， $\|\mathbf{y}\|$ 为纵轴进行绘制的点空间分布图。第二张图是 $\mathbf{x} = (x_1, x_2)$ 在平面上的分布图。第三张图是三维图的纵切面图，可以看出 $\|\mathbf{y}\|$ 的纵向取值分布。

可以看出，经过拉丁超立方采样后， \mathbf{x} 在定义域中的分布较为均匀，可以达到基本覆盖定义域空间的程度。经过 DBC 采样以及零点处采样， \mathbf{y} 在 $\mathbf{0}$ 附近的点数相比于仅进行拉丁超立法采样更为密集，验证了该数据采样与生成算法有一定效果。后续将继续根据生成的数据进行模型的建立、聚合以及方程组的求解，更加精细地验证模型的效果。

3.3 代理模型的优化与集成

3.3.1 优化算法的转换

接第三章模型的改进方案可得，模型从单一输出变为了多维输出。而原 WDDEA-DBC 最终求解方程组所使用的 FPA 算法不再适用于方程直接的求解。

具体地，原模型是在代理模型训练完毕后，借助其近似原始方程组，通过 FPA 算法搜索其输入空间，最小化输出范数，逐步逼近非线性方程组的解。而

模型经过逆转后，直接拟合非线性方程组的拟算子，求解过程只需输入全 0 向量即可得到非线性方程组的数值解而无需智能优化算法的使用。此外，改进后模型输出变成了多维，新模型的输出变为了原模型的输入，即向量 \mathbf{x} ，无法再定义新的适应值，也无法再将其投入 FPA 算法中直接进行定义域的寻优。综上，需要将智能优化算法的应用进行调整，使之适应新的问题求解场景。

因此，本课题将优化算法的应用场景从直接优化拟合后的函数转移到优化神经网络。这种转换能够带来多方面的好处。

首先，智能优化算法通常具备卓越的全局搜索能力，有助于神经网络摆脱局部最优解的束缚，从而发现更优的全局最优解。其次，这些算法还表现出良好的适应性和鲁棒性，在优化过程中能够根据反馈信息灵活调整搜索策略，以应对优化问题的复杂性和环境的不确定性。此外，智能优化算法也适用于非连续、非光滑的优化问题，并且对参数和超参数的依赖较低，因此更容易进行参数调节和优化。

3.3.2 优化算法的实施

确定智能优化算法的应用转换方法后，如何将其应用到神经网络的优化中是值得思考的。为方便对比，本课题同样采用 FPA 算法作为智能优化算法的选择。参考 FPA 算法^[17]以及其相关信息，优化过程大致如下。

首先定义问题，将 RBFNN 的参数作为优化的变量，将神经网络的性能指标作为优化的目标函数，此处为数据集在参数对应网络上计算出的适应值。

此后，初始化了一定数量的花朵个体，每个个体代表了 RBFNN 网络的一个参数组合，包括网络结构和每个神经元的权重和偏置等。然后，对每个待选参数个体，本文利用 RBFNN 网络计算该参数设置下的输出，得到当前参数下神经网络的输出结果 $\hat{\mathbf{x}}$ ，与准确值 \mathbf{x}^* 计算 MSE，作为优化问题的适应值。

接着，本课题使用自花传粉、异花传粉和 Levy 飞行、自然变异等自然法则的数学表达，来更新每个个体的位置，扩大搜索的范围，并增加了算法的随机性和多样性。

在更新位置之后，本课题根据花粉的传播策略来更新每个个体的位置，使得适应值更优的个体更有可能传播其信息给相邻个体。最终，通过一步步迭代，设置停止阈值，得到最佳的 RBFNN 参数组合。

3.3.3 模型集成

受到集成学习的启发，得到训练好的各个 RBFNN 模型后，如何进行集成聚合，从而得到更精准的求解结果，是值得探究的问题。此处仍然参考 WDDEA-DBC 模型的集成方法，通过计算单一模型的 MSE，并赋予其权重为 MSE 的倒数。具体地，假设训练得到了 T 个模型，每个模型的表达式分别为 $\hat{g}_1(\mathbf{y}), \hat{g}_2(\mathbf{y}), \dots, \hat{g}_T(\mathbf{y})$ ，MSE 分别为 $MSE_1, MSE_2, \dots, MSE_T$ ，通过式 (3-8) 进行加权。

$$\hat{g}(\mathbf{y}) = \frac{\sum_{i=1}^T \frac{g_i(\mathbf{y})}{MSE_i}}{\sum_{i=1}^T \frac{1}{MSE_i}} \quad (3-8)$$

由于越高的 MSE 代表该模型的偏差越大，因此取倒数加权的操作可以降低大偏差模型的权重，进一步增强了模型求解的准确性。

经过以上数据生成、数据扩充以及代理模型训练、加权的操作后，得到最终的模型 $\hat{g}(\mathbf{y}) = (x_1^*, x_2^*, \dots, x_d^*)$ ，当要求解非线性方程组时，仅需要对该模型输入全 0 向量 \mathbf{y}_d ，得到向量 $\hat{g}(\mathbf{y}_d)$ ，即为最终算法求解的方程组的数值解。

3.4 本章小结

本章主要对原 WDDEA-DBC 模型进行分析并作出修改。

本文首先介绍了该算法的代理模型结构及其在拟合非线性方程组中的作用，随后分析了原算法在代理模型选择和优化方面的局限性以及信息损失和误差累积问题。接着，本文提出将代理模型从拟合范数转变为拟合方程组的逆算子，并通过修改代理模型结构和训练方式，提升算法的准确性和效率。这些改进为解决非线性方程组求解问题提供了新的方法和基础。

同时，本文介绍了数据生成方法的重要性，描述了拉丁超立方采样和数据扩充的两种方法：DBC 采样和零点采样（ZPS），并展示了数据扩充算法的流程及其效果验证，进一步为模型建立和求解提供了可靠基础。

最终，得到 RDDEA-ZPS 算法的整体架构如图 3-5 所示。

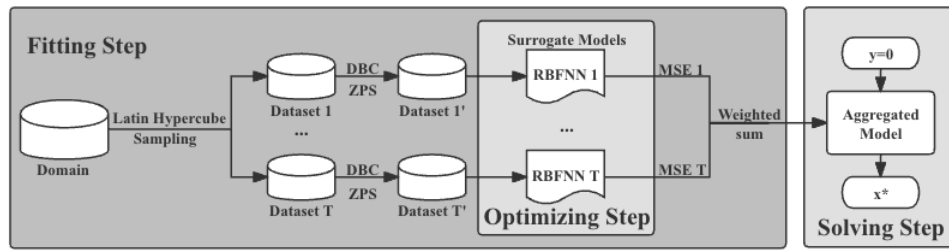


图 3-5 RDDEA-ZPS 算法架构

综合以上论述，通过修改 RBFNN 代理模型和创新性地提出 ZPS 方法，本文提出了改进后的 RDDEA-ZPS 模型。该模型能够有效处理原模型的误差问题，简化求解过程，提高了求解非线性方程组的效率和准确性。

第 4 章 数值试验与结果

4.1 参数寻优实验

4.1.1 RDDEA-ZPS 参数与算例方程组

数据生成、代理模型建立与优化过程中，涉及到若干重要的参数选择问题。这些超参数对于算法的运行效率以及结果有着较大的影响，通过调节超参数，可以控制模型的复杂度，优化模型的性能，加速模型的训练过程，从而提高模型的泛化能力和实用性。

因此在模型框架搭建完成后，如何合理地设置参数需要进一步探究。这里选出了若干对实验结果有重要影响作用的若干参数并对它们进行解释，后续将对这些参数进行数值试验，寻找最优设置。

选出的参数见表格 4-1。

表 4-1 需要优化的参数值及含义

参数符号	参数含义
$N_centers$	K-Means 聚类中心的数量，也是径向基函数网络隐藏层的神经元个数。
w	ZPS 扩充数据时需要生成的数据比例
T	集成模型中，基础代理模型的数量
l	数据扩充时，生成新数据点的波动幅度

沿用原 WDDEA-DBC 算法中的算例，本课题选择了来自论文^[18]中的以下 6 个非线性方程组作为研究与比较的对象，方程组列在表 4-2 中。

其中包含 2 维这类低维度的方程组，便于数据生成、参数寻优时进行可视化分析，也包含维度为 10 及以上的大型非线性方程组，来测试方程组在高维情况下的运行性能。

经过查阅原论文的方程组求解范围，发现原始定义域中存在多个解，考虑到目前两个模型较难求得多解，因此进一步将求解范围进行细化。

表 4-2 非线性方程组算例

编号	方程组形式	求解范围	维度
N1	$\begin{cases} x_1^2 - x_2^2 = 0 \\ 1 - x_1 - x_2 = 0 \end{cases}$	$\begin{aligned} x_1 &\in [0, 3] \\ x_2 &\in [-3, 0] \end{aligned}$	2
N2	$\begin{cases} x_1^2 - x_2 - 2 = 0 \\ x_1 + \sin(\frac{\pi x_2}{2}) = 0 \end{cases}$	$\begin{aligned} x_1 &\in [0.71, 3] \\ x_2 &\in [-3, 0] \end{aligned}$	2
N3	$\begin{cases} x_1 + x_2 - 3 = 0 \\ x_1^2 + x_2^2 - 9 = 0 \end{cases}$	$\begin{aligned} x_1 &\in [0, 1.5] \\ x_2 &\in [1.5, 3] \end{aligned}$	2
N4	$\begin{cases} x_1 - \sin(2x_1 + 3x_2) - \cos(3x_1 - 5x_2) = 0 \\ x_2 - \sin(x_1 - 2x_2) + \cos(x_1 + 3x_2) = 0 \end{cases}$	$\mathbf{x} \in [-1, 0]^2$	2
N5	$\begin{cases} x_1 - 0.25428722 - 0.18324757x_4x_3x_9 = 0 \\ x_2 - 0.37842197 - 0.16275449x_1x_{10}x_6 = 0 \\ x_3 - 0.27162577 - 0.16955071x_1x_2x_{10} = 0 \\ x_4 - 0.19807914 - 0.15585316x_7x_1x_6 = 0 \\ x_5 - 0.44166728 - 0.19950920x_7x_6x_3 = 0 \\ x_6 - 0.14654113 - 0.18922793x_8x_5x_{10} = 0 \\ x_7 - 0.42937161 - 0.21180486x_2x_5x_8 = 0 \\ x_8 - 0.07056438 - 0.17081208x_1x_7x_6 = 0 \\ x_9 - 0.34504906 - 0.19612740x_{10}x_6x_8 = 0 \\ x_{10} - 0.42651102 - 0.21466544x_4x_8x_1 = 0 \end{cases}$	$\mathbf{x} \in [-2, 2]^{10}$	10
N6	$\begin{cases} x_i + \sum_{j=1}^D x_j - (D+1) = 0 & i = 1, \dots, D-1 \\ \left[\prod_{j=1}^D x_j \right] - 1 = 0 \end{cases}$	$\mathbf{x} \in [-1, 1]^{20}$	20

后续参数寻优的过程中，将根据实际情况采用不同的方程进行数值试验。

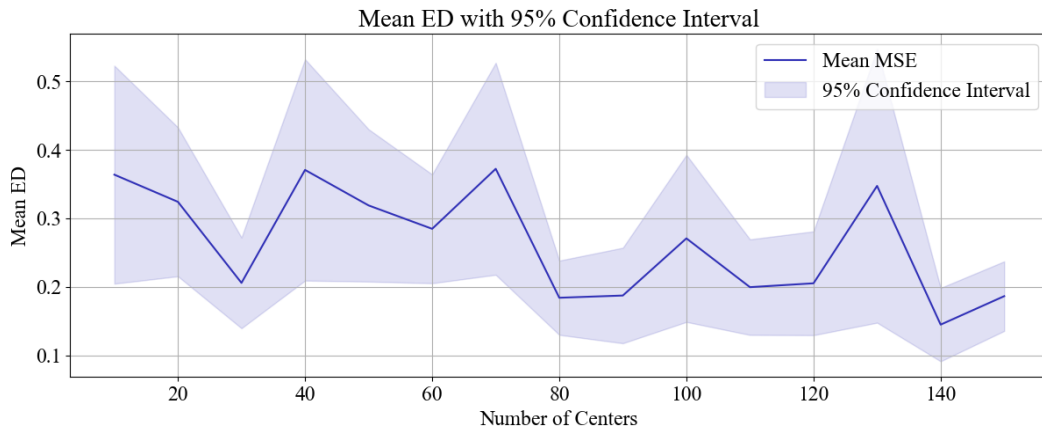
4.1.2 聚类中心数量

该参数是聚类算法 K-Means 中的类别数量 $N_centers$ ，为了节省时间开销，此步骤得到的聚类中心同时也作为径向基层的中心点，因此涉及到隐藏层神经元的数量。

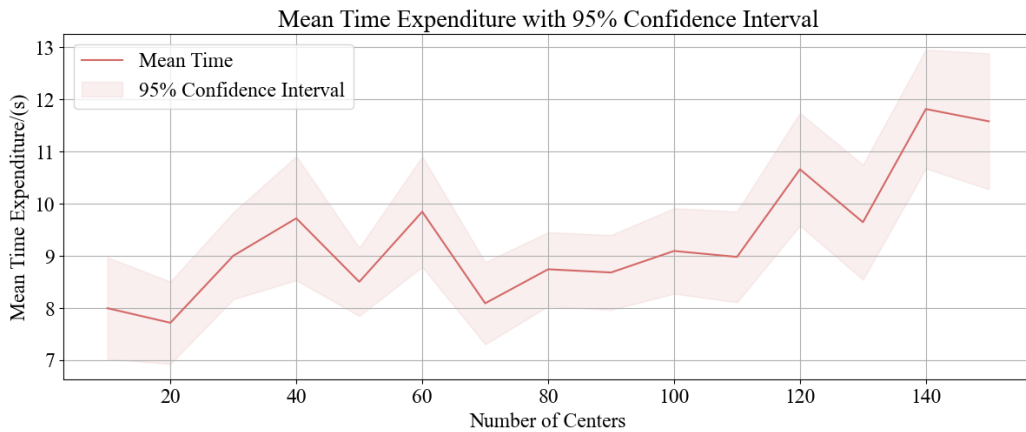
若神经元数量过小，模型过于简单，无法充分表达训练数据的特征，从而导致欠拟合的问题。在这种情况下，模型可能无法很好地捕捉数据的复杂关系，导致训练误差和测试误差都较高。然而若神经元数量过多，会使模型变得过于复杂，容易记住训练数据中的噪声和细节，而忽略了数据中的真实模式和规律，

此时还有可能出现时间开销过大的现象。因此如何选择聚类中心也即神经元的数量，需要进行数值试验的探索。

选定 N1 方程,在其他参数保持一致的情况下,设置测试中心数量分别为 10, 20, ..., 150 进行数值实验。对于每个不同的中心取值,重复 20 次完整实验,记录每个参数、每次实验的误差以及时间开销。实验结果绘制线结果趋势图如图 4-1 所示。其中误差使用预测值与真实解之间的欧式距离 (Euler Distance, ED) 衡量。



(a) N1 方程误差

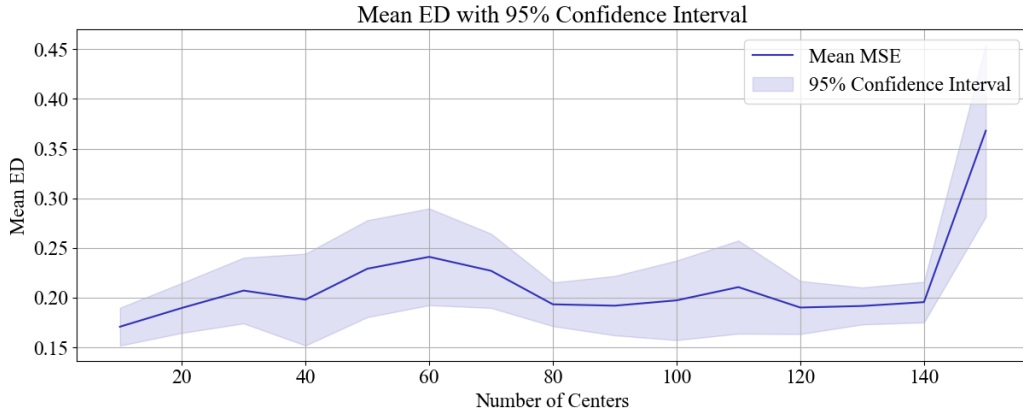


(b) N1 方程平均时间消耗

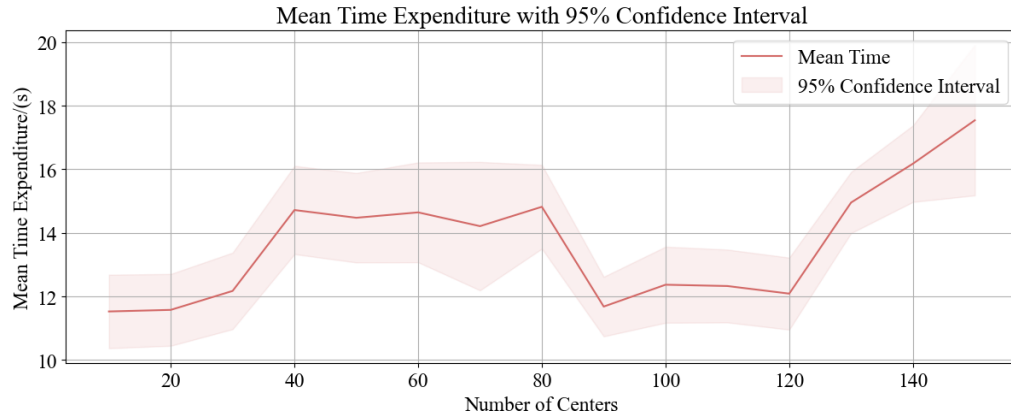
图 4-1 N1 算例上不同中心点数量寻优结果

能够看出,在低维度算例上进行重复实验,随着中心点数量的上升,平均 ED 即误差呈现波动但逐渐下降的趋势,与此同时,求解方程的时间消耗在逐渐上升。

类似地,对较高维算例 N5 进行同样的数值试验,得到结果如下图 4-2 所示。



(a) N5 方程误差



(b) N5 方程平均时间消耗

图 4-2 N5 算例上不同中心点数量寻优结果

在高维算例上，对于不同的中心点数量，模型优化的误差差异没有低维度算例大，排除由于计算不稳定而造成的中心数量 150 处的极端值的影响，随着中心数量的增大整体的趋势但仍然呈现出中心少时误差大时间小、中心多时误差小时间较长的趋势。

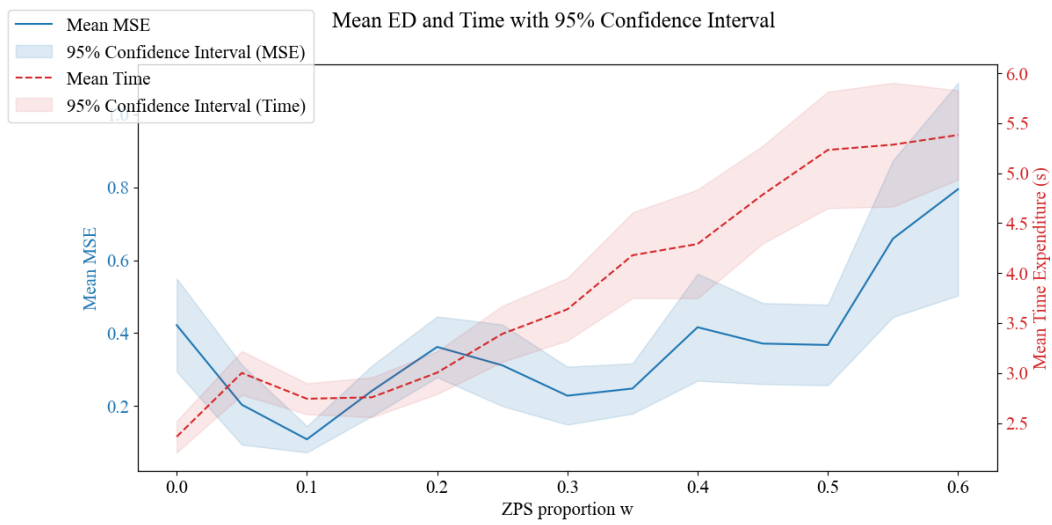
综合两个算例的曲线走向，权衡后选定 $N_centers$ 为 80，为后续求解方程的中心数。

4.1.3 ZPS 数据扩充比例

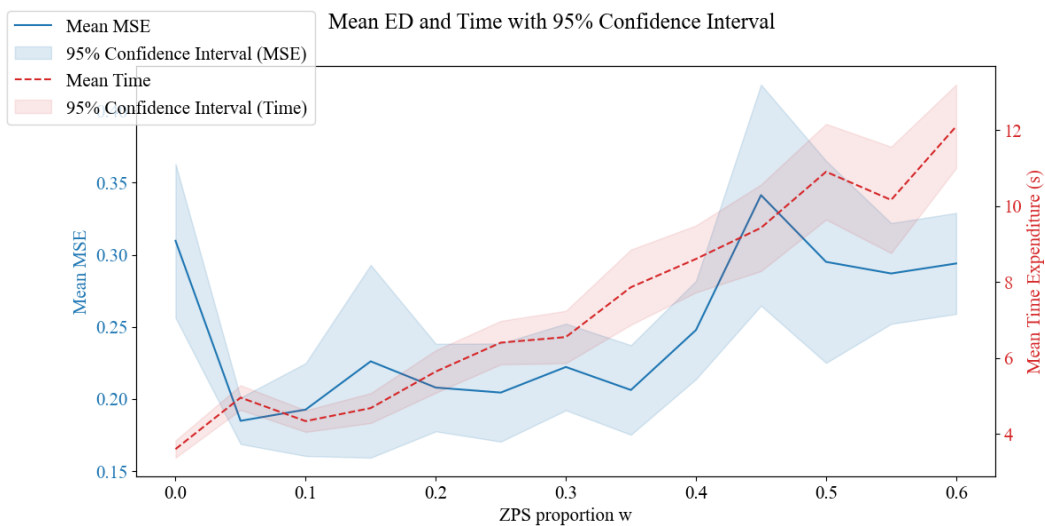
零点采样是扩充有效数据集的重要途径之一，其采样比例 w 的取值也会影响模型求解非线性方程组的准确率与效果。

当采样数量较小时，只有很少的数据经过采样，零点附近的数据仍然不够模型捕捉解附近的变量对应关系。而当有较多的数据经过 ZPS 采样后，会导致大部分的数据都进行了拓展，和普通的 DBC 没有本质差别。因此 w 的选择也是本文需要通过实际数值试验进行斟酌的重要参数。

同样的，在 N1 和 N5 这两个维度不同的方程组上进行数值试验，对于 w 的设置，选取 0, 0.05, 0.1, ..., 0.55, 0.6 这一系列以 0.05 为等差的序列，依次进行试验。在 N1、N5 方程上的实验结果由图 4-3 展示。



(a) N1 方程寻优结果



(b) N5 方程寻优结果

图 4-3 算例上不同采样比例 w 寻优结果

图 4-3 和图 4-2 所表示的变量、横纵坐标相同，4-3 将两个变量同时移到同一个坐标系中，便于通过交点、趋势等更直观地选择参数。

从两个方程的寻优结果图中可以看出，求解得到的解的误差随着采样比例的增加，都呈现先下降后上升的趋势，两张图的误差均在 $w = 0.1$ 附近达到极小值；同时，求解两个方程组的时间开销都随着采样比例的增加而增加。因此，综合误差与时间，最终选择 $w = 0.1$ 作为 ZPS 数据扩充的比例，即对于全部数据集的 10% 进行零点采样进行扩充。

4.1.4 代理模型数量

由于 RDDEA-ZPS 算法同样使用了 Bagging 框架进行代理模型的管理，即集成，因此代理模型的数量 T 的选取对于算法的求解结果也有一定的影响。

集成学习的效果依赖于个体模型之间的差异性。当代理模型数量太少，可能导致集成模型的偏差较大，无法捕捉到数据的复杂性，从而影响最终结果。然而，增加基础模型的数量也意味着增加了集成模型的复杂性。如果基础模型的质量较低，增加数量可能会导致集成模型的方差增加，从而降低整体性能。因此，需要平衡模型数量和质量。

在本节中同样采用重复实验的方式，保持其他参数设置相同的情况下，对 N1、N2 和 N5 三个算例上进行数值实验。在模型数量 T 为 10、20 和 30 这三种情况下进行寻优。寻优的结果绘制箱线图 4-4 进行对比。

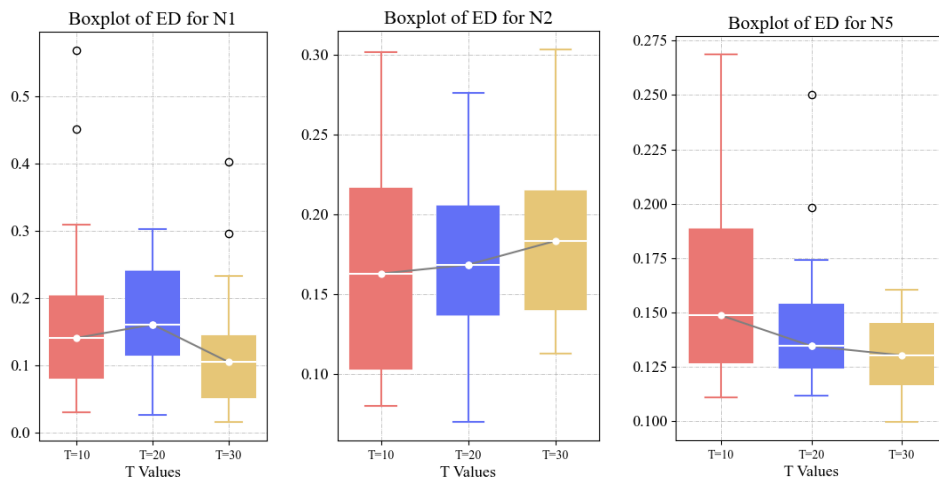


图 4-4 代理模型数量寻优

可以看到，在 N1 和 N2 低维算例中，模型在 $T=10$ 和 $T=20$ 的条件下的 ED 损失相似，在 $T=20$ 时的损失分布更集中； $T=30$ 时，在两个低维算力的表现不稳定，在 N1 中 ED 损失更低，而在 N2 中，该损失较大且分布分散程度也更大。

在高维算例中，ED 损失随着代理模型数量的增加而递减。在 $T=10$ 时损失分布分散程度较大，呈现出不稳定的状态，并且 ED 且均值明显高于其余两种情况； $T=20$ 和 30 的情况下，两者的 ED 损失接近。

综合以上三个方程的优化情况，并综合程序运行效率等因素，后续模型选择 $T=20$ 作为代理模型的数量进行方程组的求解。

4.1.5 ZPS 扩充数据的波动范围以及比例组合

对于数据扩充步骤，除 3.2 节中提到的需要进行数据扩充的比例 w ，数据生成时控制 δ_i 变化范围的系数 l 也是需要通过实验确定的。这是由于 l 直接影响了在目标数据点周围生成的新数据点与原数据的偏离程度，具体表达式由 (3-5) 和 (3-6) 给出。

当 l 较小时，原数据点与新数据点距离近、偏离程度小，因此可能导致生成的数据十分接近原数据，这样一来就不能发挥数据扩充算法的最佳效果；而当 l 较大时，新数据与原数据点之间的分布较远，差异较大，因此新数据可能会落入其他点的邻域范围内，导致新、旧数据的相似度过差，生成的数据便不能很好地近似原有数据点的特点。因此，如何选择范围值 l ，则是需要权衡的又一超参数。

此外， l 和 w 的搭配也至关重要。 w 控制了需要新生成数据的比例， l 控制了生成数据的波动程度，它们共同影响了数据扩充算法生成的新数据点与原数据点之间的相似度和波动程度。

具体来说，当选择的参数 l 较小时，生成的新数据点与原数据点的相似度较高，因为它们之间的距离较近。然而，如果此时使用较大的参数 w 来控制数据扩充的比例，那么可能会导致大量的新数据点与原数据点重叠，丧失了数据扩充算法的优势。当选择的参数 l 较大时，新数据点与原数据点之间的波动程度会增加，因为它们之间的距离较远。在这种情况下，如果仍然使用较大的参数 w 来控制数据扩充的比例，那么可能会导致生成的新数据点过于分散，失去了对原数据分布的良好近似。

诸如以上两类的情况等，对探究 l 与 w 的搭配提出了挑战。本节将选择一系列 l 和 w 的候选值，依据其不同搭配，进行网格寻优，每次搭配重复 20 次训练，并

求出指标的平均值，绘制热力图。结果如图 4-5 所示， l 和 w 的取值选项在横纵坐标中标出。

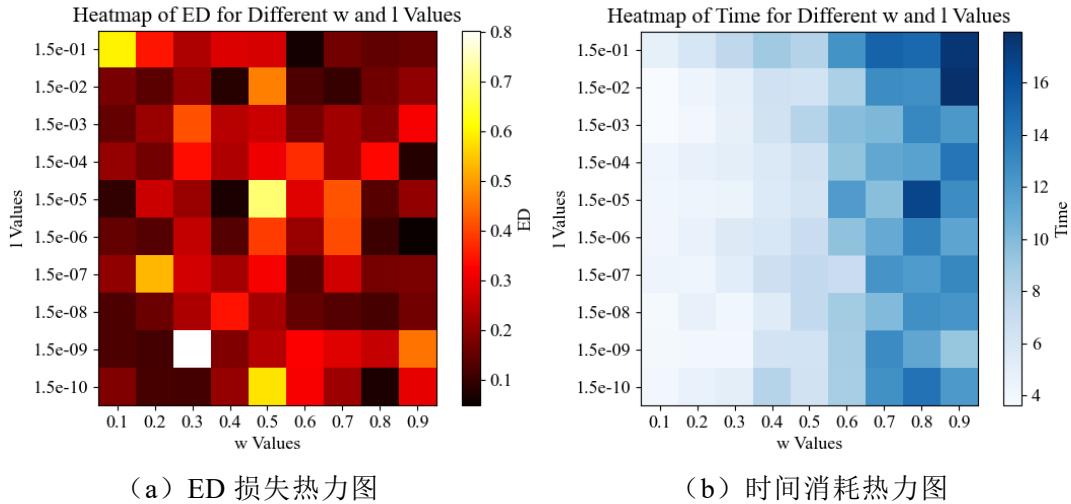


图 4-5 w 和 l 参数搭配 ED 与时间消耗热力图

由图看出，对于 ED 损失，相比于其他位置的热力值，左上角到右下角的对角线上颜色更浅，表示当 w 或 l 中的某一值较大时，都会给实验带来较大的 ED 误差；而左上、右下角位置的热力值较低，表示当两个参数同等程度的增加或减小时，寻优效果较好。然而考虑到当两者均较大时，对于数据生成的范围无法精确到目标范围，反而可能会降低效果、增加程序运行开销。这启示我们应从该热力图的左下方选择参数搭配。

对于时间消耗，可以看到热力图左右两侧的对比较为明显。横向来看，对于不同的 w 值，随着其值的增加，时间消耗显著增加；纵向来看，对于 l 的不同取值，时间消耗上没有明显差异。这一现象也与实际情况相符合，因为 l 只控制生成数据的范围而非多少，但 w 控制了需要生成数据的多少并直接与算法运行时间相关。

综合以上两个指标，为了保持数据生成范围的精确性和一定的广度，同时最大限度地节省运行时间，最终选择 $w = 0.1$ 和 $l = 1.5 \times 10^{-3}$ 作为最终的参数值。这个选择与 4.1.3 中的参数一致。

4.2 模型求解结果对比

4.2.1 误差对比

在最佳参数设置下，对原模型 WDDEA-DBC 以及改进后的模型 RDDEA-ZPS 进行若干次实验，并进行 ED 损失的对比。对于每个算法，在 N1~N6 算例上分别运行 20 次，记录每个模型的 ED 误差在表 4-3 中，计算均值与方差。

表 4-3 两种算法 ED 误差对比

算例编号	RDDEA-ZPS	WDDEA-DBC
N1	0.0693±0.0427	0.8200±0.8502
N2	0.1896±0.0359	0.6857±0.4842
N3	0.1896±0.673	1.3836±0.3781
N4	0.0425±0.0235	0.3507±0.1502
N5	0.1426±0.0185	0.4752±0.1778
N6	0.3794±0.0043	1.6690±0.4905

结果显示，在所有六个算例中，RDDEA-ZPS 算法的平均误差均明显小于 WDDEA-DBC 算法，并且由标准差的大小可知，RDDEA-ZPS 的稳定性要远远强于 WDDEA-DBC 算法。这直接表明了改进后算法求解的强大能力。

4.2.2 求解结果对比

最终使用原模型 WDDEA-DBC 以及改进后的模型 RDDEA-ZPS 进行若干次实验，两个模型给出在新的定义域范围的求解结果，如表 4-4 所示。

表 4-4 两种算法求解结果对比

编号	自变量	真解	RDDEA-ZPS 解	RDDEA-ZPS 误差	WDDEA-DBC 解	WDDEA-DBC 误差
N1	x_1	0.5000	0.5004	0.0530	0.6620	0.2149
	x_2	-0.5000	-0.4469		-0.6412	
N2	x_1	1.0000	0.9301	0.1737	1.1477	0.1741
	x_2	-1.0000	-1.1590		-0.9078	
N3	x_1	0.0000	0.1359	0.1365	0.5755	0.5864
	x_2	3.0000	3.0134		2.8875	

表 4-4（续表）

编号	自变量	真解	RDDEA-ZPS 解	RDDEA-ZPS 误差	WDDEA-DBC 解	WDDEA-DBC 误差
N4	x_1	-0.1733	-0.1809	0.0124	-0.4115	0.2401
	x_2	-0.2561	-0.2659		-0.2264	
N5	x_1	0.2578	0.1739	0.1621	0.1294	0.2983
	x_2	0.3811	0.3708		0.1551	
	x_3	0.2787	0.3433		0.2451	
	x_4	0.2007	0.2651		0.2067	
	x_5	0.4453	0.4795		0.4550	
	x_6	0.1492	0.1556		0.1412	
	x_7	0.4320	0.4801		0.3043	
	x_8	0.0734	0.0796		0.0389	
	x_9	0.3460	0.3776		0.3184	
	x_{10}	0.4273	0.5064		0.4703	
N6	x_1	1.0000	1.0378	0.1889	0.9255	0.3752
	x_2	1.0000	0.9774		0.9088	
	x_3	1.0000	1.0391		0.9426	
	x_4	1.0000	1.0000		0.9147	
	x_5	1.0000	0.9035		0.8805	
	x_6	1.0000	1.0245		0.9265	
	x_7	1.0000	1.0119		0.9207	
	x_8	1.0000	1.0285		0.9368	
	x_9	1.0000	0.9911		0.9099	
	x_{10}	1.0000	1.0681		0.9433	
	x_{11}	1.0000	0.9975		0.9223	
	x_{12}	1.0000	0.9120		0.8889	
	x_{13}	1.0000	1.0406		0.9398	
	x_{14}	1.0000	0.9913		0.9109	
	x_{15}	1.0000	1.0057		0.9248	
	x_{16}	1.0000	0.9284		0.8855	
	x_{17}	1.0000	0.9935		0.9161	

表 4-4（续表）

编号	自变量	真解	RDDEA-ZPS 解	RDDEA-ZPS 误差	WDDEA-DBC 解	WDDEA-DBC 误差
	x_{18}	1.0000	0.9865		0.9101	
	x_{19}	1.0000	0.9874		0.9143	
	x_{20}	1.0000	1.0402		0.9376	

结果显示，RDDEA-ZPS 模型在全部六个算例上相较于原模型有明显提升，验证了新的模型框架的有效性。

4.3 消融实验

4.3.1 模型结构实验

RDDEA-ZPS 算法在改进中对代理模型的结构、优化的过程等架构方面进行了较大的变化，如何验证此类修改的有效性，可以通过变换模型部分架构来验证新模型的总体有效性。

在本节，模型将进行拆解，用做对比的模型分为以下几个：仅使用 FPA 寻优求解方程组、RDDEA-ZPS 仅使用 Adam 优化算法、RDDEA-ZPS 仅使用单一代理模型而不加权。

对于第一个对比模型，即仅使用 FPA 进行优化的模型，具体来说去掉后续 RBFNN 这一神经网络拟合非线性方程组逆算子的情况。这一模型是为了验证算法提出的拟合逆算子神经网络对求解方程组的有效性和准确性的影响。

对于第二个 RDDEA-ZPS 仅使用 Adam 优化算法这一模型，在原算法的基础上去掉了 FPA 神经网络寻优这一步骤，而更改为仅仅使用传统神经网络优化算法 Adam 算法进行网络参数的寻找，同 WDDEA-DBC 模型中优化神经网络的方法一样。这一步在 WDDEA-DBC 算法的基础上，确保未引入新的参数寻优算法的同时，主要检验 FPA 算法在神经网络寻优方面的效率，以及对最终的解方程结果的影响。

对于第三个模型，RDDEA-ZPS 仅使用单一代理模型而不加权，指的是去掉 MSE 倒数加权这一集成方法，而仅仅采用单一代理模型进行求解。这一步的目的是为了验证 RDDEA-ZPS 算法中模型集成这一模块的作用。

对于以上提到的三个模型，和完整的模型，在后续表格中分别记作 FPA，Adam，Single 和 RDDEA-ZPS，在六个算例上运行以上不同的模型 20 次，得到

的结果如表 4-5 所示。

表 4-5 模型消融 ED 误差对比

编号	FPA	Adam	Single	RDDEA-ZPS
N1	0.1080±0.0631	0.5132±0.4043	0.4003±0.3772	0.0693±0.0427
N2	1.2970±0.0060	0.1284±0.0594	0.2158±0.0959	0.1896±0.0359
N3	1.6770±0.0010	0.2945±0.1725	0.3576±0.3286	0.1896±0.673
N4	0.0584±0.0428	0.6986±0.0605	0.0852±0.0867	0.0425±0.0235
N5	0.8898±0.0536	0.7853±0.0321	0.2963±0.1264	0.1426±0.0185
N6	0.9186±0.1548	0.2175±0.0297	0.3859±0.0194	0.3794±0.0043

从不同模型的实验结果可以看出，完整的 RDDEA-ZPS 模型在六个算例上的整体误差最小，效果最佳。

对于仅使用 FPA 而未应用神经网络代理模型方法的非线性方程组求解方法，可以看出在六个算例上的结果均较差，除了在 N1、N4 上的误差接近于完整模型以外，在其余的算例上的欧拉距离误差均较大。这证明了使用神经网络拟合逆算子的方法对于求解非线性方程组这一问题来说有较明显的作用。

对于仅仅使用一个代理模型而未经多个模型集成的算法，即表格中的 Single，在六个算例上的整体表现较仅使用 FPA 的算法好，但准确率仍然总体低于完整模型。此外，由于未经集成，模型呈现出较大的不稳定性，具体体现在 20 次误差的方差较大。这一消融模型证明了 MSE 倒数加权这一集成模型的方法能够有效减小误差，提高算法的准确率，同时证明了集成后模型算法的稳定性。

对于仅使用 Adam 优化神经网络参数的模型，可以看到其在 N2 和 N6 两个算例上的表现略好于完整模型，而完整模型在其他算例上的误差较小。这一步的对比可以体现出相较于传统的 Adam 优化法，FPA 在参数寻优方面可以一定程度上更精细地在有效参数附近搜寻更优的参数组合，提高算法的准确率。

通过以上四个不同架构的算法对比，在整体上可以证明 RDDEA-ZPS 模型的各个架构在求解非线性方程组问题的必要性。

4.3.2 数据采样实验

本模型中提到的数据采样方法在一定程度上也能提高算法的准确度，为了能够验证每一部分的作用，本课题对于数据生成部分的各个模块也进行了消融实验。

正如第四章提到的内容，数据集的生成主要由拉丁超立方采样和数据扩充两部分，其中后者又分为 DBC 和 ZPS 数据扩充方法，以下将分别对 DBC、ZPS 两部分进行消融实验，检测数据扩充中两种新算法的作用。

以下仅使用拉丁超立方采样的模型记作 Latin，去掉 ZPS 的模型记作 Latin+DBC，去掉 DBC 模块的模型记作 Latin+ZPS，完整采样模型记作 Full。保持其他参数相同的同时，对不同的模型设计进行 15 次，记录平均 ED 误差和其标准差，结果如表格 4-6 所示。

表 4-6 数据采样消融 ED 误差对比

编号	Latin	Latin+DBC	Latin+ZPS	Full
N1	0.2247±0.2041	0.2914±0.0857	0.1633±0.1069	0.0769±0.0456
N2	0.1690±0.0705	0.1154±0.0494	0.1633±0.0657	0.1948±0.0377
N3	0.1389±0.0805	0.2642±0.1087	0.0827±0.1232	0.1814±0.0627
N4	0.2335±0.0612	0.1386±0.0503	0.0589±0.0377	0.0451±0.0233
N5	0.1582±0.0454	0.2549±0.7901	0.1991±0.1035	0.1458±0.0200
N6	0.4177±0.0669	0.4051±0.0237	0.4007±0.0054	0.3787±0.0044

从结果可以看出，采用完整的数据采样流程的算法在大部分算例上的表现较好，在 N2、N3 这两个算例上表现不是最佳但相应的误差数据相差不大，并且完整模型的误差标准差最低，模型最稳定。

由第一列和最后一列的完整模型对比，仅仅采用拉丁超立方采样的算法在全部六个算例上的误差均偏大。这表明后续的包含 DBC 和 ZPS 的数据扩充步骤对于整体算法准确性的提高有重要作用。

对于使用采用拉丁超立方采样和 DBC 采样的算法，相比于仅使用拉丁超立方采样的数据，准确性并未有较大提升，ED 误差和误差的方差接近。这证明了对于非线性方程组求解的问题，DBC 采样虽然可以使得数据分布更加均匀，但无法有效增强数据最重要的部分，实验结果与第四章的分析一致。

对于使用采用拉丁超立方采样和 ZPS 采样的算法，相比于以上两种算法，在准确性上有着较大的提升，甚至部分算例上的结果超过了使用了 DBC 扩充方法的完整模型。这证明了本课题新提出的 ZPS 在增强重要数据、提高算法准确率上十分有效。

经过以上实验和分析，可以验证数据生成与扩充步骤各个模块的有效性。接着，将以上各个模型的 15 次实验结果绘制成箱线图，更直观地展示模型之间的

差异，得到的结果如图 4-6 所示。

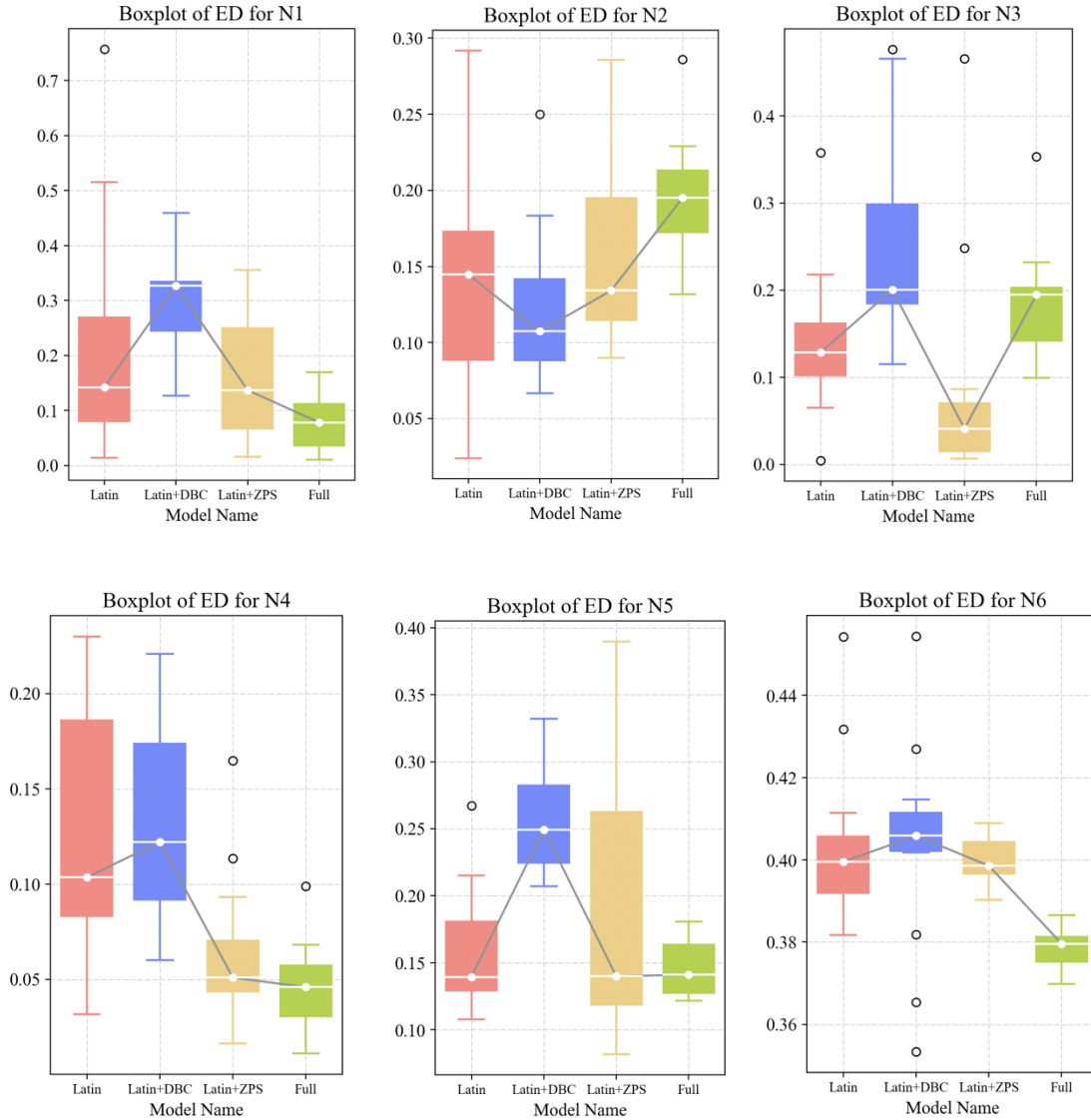


图 4-6 不同模型的 ED 误差箱线图

从图中可以看出，完整模型在大部分算例上的误差最小，其次是使用了拉丁超立方采样和 ZPS 采样模型的算法，这与前文的分析相一致，进一步证明了含 ZPS 方法在内的数据采样与扩充算法的有效性。

4.4 本章小结

本章通过对 RDDEA-ZPS 算法中四个关键参数的优化选择及验证实验，提升了算法的性能。本文研究了聚类中心个数($N_centers$)、ZPS 数据扩充比例(w)、

代理模型数量 (T) 和数据生成波动范围 (I) 的选择对算法性能的影响, 并确定了最优参数组合。这些参数优化提升了模型的表达能力和泛化能力, 确保了算法的有效性。

进一步, 通过对比两种算法在非线性方程组求解中的表现, 结果显示 RDDEA-ZPS 模型在所有算例上均优于原模型, 验证了新模型框架的有效性。在改进过程中, 本文对代理模型的结构和优化过程进行了调整, 通过消融实验验证了各个改进措施的必要性。实验结果表明, 完整的 RDDEA-ZPS 模型在整体误差和稳定性方面表现最佳。

综上所述, 这些研究和实验验证了 RDDEA-ZPS 模型及其数据采样与扩充算法在求解非线性方程组问题上的优越性和必要性。

结 论

本课题主要针对非线性方程组的特点，通过拓展 DDEA 算法到新的求解场景，提出了一种新的适配于非线性方程组求解的数据驱动智能优化算法，RDDEA-ZPS 算法。

课题主要围绕着 WDDEA-DBC 模型及基于其的改进版本 RDDEA-ZPS 展开，系统性地探讨了该模型在非线性方程组求解中的应用及优化过程中的关键问题与解决方案。

首先，对 WDDEA-DBC 模型作为改进模型的 Baseline 进行了详细介绍和分析。通过分析 WDDEA-DBC 算法，展示了其数据处理方法和结构的创新之处，可用于非线性方程组的求解。此后，本文着重分析了该模型应用于非线性方程组场景下的问题。针对原算法存在的问题，如代理模型选择、数据集的格式以及优化算法的局限性等，提出了修改代理模型结构和训练方式的方案。

此后，本文开始构建针对 WDDEA-DBC 算法问题而改进后的 RDDEA-ZPS 算法。在代理模型构建中，本文逆转了原 RBFNN 网络的输入输出，并且还原了原始数据集，减少了因引入范数、使用 FPA 算法而造成的误差。数据生成的过程中，在拉丁超立方采样、DBC 采样的基础上，提出了零点采样（ZPS）的方法，拓展了数据生成的方法，更精确地生成高价值的数据，并为模型建立和求解提供了可靠的数据基础。随后，本文详细讨论了 RDDEA-ZPS 模型的优化以及加权过程，及其相对于原始模型的优势和改进之处，为解决非线性方程组求解问题提供了新的思路和方法。

提出完整的改进算法框架后，本文通过在算例上进行数值模拟实验，确定了 RDDEA-ZPS 算法中关键参数的选择，在后续求解过程中提高了算法的性能和效果。最终使用最佳参数进行非线性方程组的求解与效果对比，发现改进后的模型求解精度更高，效果明显。

最终通过消融实验，对模型的不同架构进行拆解，进一步验证了模型各个部分的有效性。

本论文的主要创造性工作归纳如下：

1. 将全新领域的算法——数据驱动的优化算法引入非线性方程组的求解中，拓宽了算法应用的空间，为非线性方程组的求解提供了新思路。
2. 提出了新的数据生成算法，在非线性方程组求解场景下较好地提高了求解精度。

3. 针对非线性方程组问题的求解特点，创造性地修改了代理模型的主要框架，通过逆转输入输出、转换智能优化算法应用场景等方法，极大的减少了引入的误差。

4. 新算法从单一目标优化转化为了多目标优化，为 DDEA 算法在多目标优化场景下提供了思路。

5. 优化后的模型极大的简化了非线性方程组的求解步骤以及难度。

由于时间、本人能力等限制，目前工作仍然缺少一定深度，今后将在后续几个方面继续深入研究：

1. 目前的优化算法仍然只能定位一个解，对于定义域区间内有多个解的情况，算法无法很好地解决。

2. 目前算法仅可以求解自变量和方程数相同的非线性方程组，对于数目不匹配的情况无法处理。

3. 模型的性能还有提升的空间，未来将通过修改模型框架、研究比较时间开销等方法继续完善优化算法。

参考文献

- [1] GONG W, LIAO Z, Mi X, et al. Nonlinear Equations Solving with Intelligent Optimization Algorithms: A Survey [J] . Complex System Modeling and Simulation, 2021, 1(1): 15-32.
- [2] JIN Y C, WANG H D, CHUGH T, GUO D, MIETTINEN K. Data-Driven Evolutionary Optimization: An Overview and Case Studies [J] . IEEE Transactions on Evolutionary Computation, 2019, 23(3): 442-458.
- [3] GUO Z L, LIN S K, SUO R Z, ZHANG X M. An Offline Weighted-Bagging Data-Driven Evolutionary Algorithm with Data Generation Based on Clustering [J] . Mathematics, 2023, 11(2): 431.
- [4] BROYDEN C G. A class of methods for solving nonlinear simultaneous equation [J] . Mathematics of Computation, 1965: 577-593.
- [5] RAMOS H, MONTEIRO M. A new approach based on the newtons method to solve systems of nonlinear equations [J] . Journal of Computational and Applied Mathematics, 2017: 3-13.
- [6] 吴龙, 任红民, 毕惟红. 遗传算法求解非线性方程组研究综述 [J] . 电子科技, 2014 (4) :173-178.
- [7] YANG X S. Flower Pollination Algorithm for Global Optimization [J] . Unconventional Computation and Natural Computation, 2012: 240-249.
- [8] 李盈晖, 邱菀华. 用人工神经网络求解非线性方程组的一种方法 [J], 系统工程与电子技术, 1997, 19 (5): 3.
- [9] 赵启林, 卓家寿. 非线性方程组的耦合神经网络算法 [J] . 河海大学学报(自然科学版), 2000, (5): 38-40.
- [10] 赵华敏, 陈开周. 解多元非线性方程组的神经网络方法 [J] .西安电子科技大学学报, 2000, (04): 480-482.
- [11] LI G M and ZENG Z Z. A Neural-Network Algorithm for Solving Nonlinear Equation Systems [C] . 2008 International Conference on Computational Intelligence and Security, Suzhou, China, 2008: 20-23.
- [12] 孙银慧. 求解非线性方程组的迭代神经网络算法 [J] . 计算机工程与应用, 2009, 45 (6): 55-56.
- [13] 赵庆兰, 李文, 董晓丽. 改进的求解非线性方程组的迭代神经网络算法 [J] .现代电子技术, 2013, 36 (08): 20-22.
- [14] WANG H D, JIN Y C, SUN C L, DOHERTY J. Offline Data-Driven

- Evolutionary Optimization Using Selective Surrogate Ensembles [J] . IEEE Transactions on Evolutionary Computation, 2019, 23(2): 203-216.
- [15] LI J Y. Boosting Data-Driven Evolutionary Algorithm with Localized Data Generation [J] . IEEE Transactions on Evolutionary Computation, 2020, 24(5): 923-937.
- [16] WANG M. Offline data-driven evolutionary optimization algorithm using k-fold cross [C]. In Advances in Swarm Intelligence, Proceedings of the International Conference on Sensing and Imaging; Springer: Cham, Switerlands, 2022: 305–316.
- [17] CAO H Q, NGUYEN H X, TRAN T N-C, TRAN H N, JEON J W. A Robot Calibration Method Using a Neural Network Based on a Butterfly and Flower Pollination Algorithm[J]. IEEE Transactions on Industrial Electronics, 2022, 69(4): 3865-3875.
- [18] LIAO Z, GONG W, YAN X, WANG L and HU C. Solving Nonlinear Equations System with Dynamic Repulsion-Based Evolutionary Algorithms [J] . IEEE Transactions on Systems Man and Cybernetics Systems, 2018: 1-12.

本科毕业论文（设计）原创性声明

作者签名: _____ 日期: _____ 年 _____ 月 _____ 日

本人知悉本科毕业论文（设计）的使用权限，并将遵守有关规定。

作者签名: _____ 日期: _____ 年 _____ 月 _____ 日

导师签名: _____ 日期: _____ 年 _____ 月 _____ 日

致 谢

当写下最后一个字符时，我充满了对大学四年的感激。

感谢我的导师，给我算法的启发与灵感，让我在遇到困难时，仍然有前进的方向。每一次与他的交流都是我学习和成长的机会，他是我学术道路上的明灯，启迪人，没有他的悉心教诲，便没有这篇文章的完成。

其次，我要衷心感谢我的舍友们。她们不仅是我的同窗，更是我大学生活中最亲密的伙伴和支柱。在她们身上我看到了友爱和包容，她们与我同甘共苦，每一次的相处都让我受益匪浅。她们的陪伴和理解让我在异乡求学时感到温暖和安心，是我生活中的一笔珍贵财富。

除此之外，我还要感谢我大学中认识的所有朋友们。他们与我同行，见证了我大学四年的点点滴滴。在我遇到困难时，他们始终在我身旁给予我无私的支持和鼓励，是我人生中不可或缺的一部分。

最后，我要特别感谢 Taylor Swift。她的音乐伴随着我度过了大学时光中的每一个重要时刻，她的歌曲如一股清泉，给我心灵上的滋养和力量。她的音乐不仅是我生活中的一部分，更是我追求梦想的动力和信仰。

在此，我要衷心感谢所有支持和帮助过我的人，是你们的关爱和支持让我成为了今天的自己。我将永远怀着感激之情，铭记于心。