

基于神经网络的多任务优化知识迁移

摘要

知识迁移在进化多任务优化中对于任务优化至关重要。然而，大多数现有的知识迁移方法只能实现表面的知识迁移，缺乏深入挖掘任务之间相似性或关系的能力。这一局限性可能会导致负迁移，从而降低知识迁移性能。由于 KT 效率强烈依赖于任务之间的相似性，基于神经网络的知识迁移方法 (NNKT) 可以用于分析任务之间的相似性，并获取迁移模型，以实现高质量的 KT。本文复现了基于神经网络的知识迁移解决单目标多任务问题的算法，并拓展运用神经网络的技术来解决多目标多任务优化问题。通过基于分解的多目标优化算法 (MOEA/D)，利用神经网络作为桥梁，训练任务之间的迁移模型，NNKT 能够深入挖掘任务间的相似性，并利用源任务的已有知识预测目标任务的潜在解，从而实现任务间更高质量的协同进化。将所提的基于神经网络的多目标多任务算法与 IEEE CEC2017 和 IEEE 2019 基准上的一些多任务算法的实验结果进行比较，证明了 NNMOMTO 的效率和有效性。

关键词：进化计算；进化多任务；知识迁移；神经网络

1 引言

进化计算被广泛应用于解决各种优化问题 [1], [2]，然而，传统的 EC 算法一次只能解决一个优化任务，这可能无法满足同时运行多个任务的现实世界场景的要求，例如多系统包设计 [3] 和储层模型中多个井位的优化 [4]。例如，在优化多个井位时，目的是同时确定不同特定油藏中的最佳井位，以最大限度地提高多个油藏的最终石油采收率。当使用传统的单任务 EA 来解决多个任务时，一次只能解决一个任务，并且每次必须为不同的任务重新启动流程。问题中的多个任务通常共享相似或相关的知识，这些知识可以用来促进不同任务的优化。因此，在 EC 中出现了一种新的优化问题，称为多任务优化 (MTO)。与传统的独立解决任务而不需要知识转移 (KT) 的 STEA 相比，MTO 利用不同任务之间的相似性或关系同时解决多个任务，以促进进化过程中多个任务的优化。

当使用 EC 解决 MTO 问题时，这些算法被称为进化 MTO (EMTO) 算法，通常可分为两类：1) 单种群算法和 2) 多种群算法。单种群算法使用基于多因子 EA (MFEA) 的种群来解决 MTO 问题 [5]，而多种群算法利用多个种群，每个种群对应于优化一个任务。然而，无论类别如何，KT 在促进多任务优化方面都发挥着关键作用。例如，由于不同的问题更喜欢不同的交叉算子，因此具有自适应 KT 的 MFEA (MFEA-AKT) [6] 在 KT 期间根据进化过程中的历史信息自适应地为每个个体分配合适的交叉算子。

尽管许多 KT 方法在一定程度上取得了成功，并被广泛应用于各种 EMTO 算法中，但它们只实现了表面的 KT，这通常会传递最佳 [7] 或随机 [5]、[6] 的解决方案，而没有充分挖

掘任务之间的相似性。由于 KT 效率取决于任务之间的相似性，如果可以挖掘不同任务之间的类似性或关系，就可以实施更有效的 KT 过程

神经网络 [8], [9] 是强大的人工智能技术，理论上可以以任何精度学习两个函数之间的关系。因此，神经网络可用于获得迁移模型，以深入挖掘不同任务之间的相似性。通过使用训练好的神经网络作为迁移模型，知识可以在不同的任务之间有效地转移。基于这一基本原理，本文应用基于神经网络的 KT (NNKT) 方法，通过挖掘任务之间的相似性来增强个体学习，以求解多目标多任务优化问题。

2 相关工作

2.1 EMTO

在不失一般性的情况下，多任务优化问题 (MTOP) 被表述如下。我们假设在一个问题中需要解决 K 个单目标任务，这些任务表示为 T_1, T_2, \dots, T_K 。

多任务优化问题的解定义如下：

$$x_i^* = \arg \min_{x \in X_i} f_i(x), \quad i = 1, 2, \dots, K$$

其中， f_i 是任务 T_i 的目标函数； x_i^* 是任务 T_i 的最优解； X_i 是任务 T_i 的搜索空间。

由于不同任务具有不同的搜索空间，通常定义一个统一的搜索空间，在该空间中可以对所有任务的解进行统一编码。在个体进化和知识迁移 (KT) 过程中，这些解在统一的搜索空间中进行操作。对解进行评估时，会将其解码回原始搜索空间以计算其适应值。

通常，两个任务之间的相似知识可以促进任务的相互优化。因此，挖掘任务之间的相似性或关系对于构建高质量的知识迁移方法是非常有利的。

2.2 相关工作

受多因子遗传模型启发，首个多任务优化算法 MFEA [5] 被提出，利用单一种群在统一搜索空间中解决多个任务，通过交叉操作实现隐式知识迁移，增强任务优化。EMTO 算法分为单种群和多种群方法。单种群方法维护一个种群，采用隐式知识迁移；多种群方法为每个任务维护一个种群，并采用各种知识迁移方法。

在单种群方法中，基于 MFEA 框架的改进算法包括 MFEA2 [10]，该算法通过在线估计任务间协同作用自适应调整知识转移频率；MFEA-AKT [6] 则选择最佳交叉算子；广义 MFEA (G-MFEA) [11] 通过决策变量的平移与重排处理异构任务；基于分组的 MFEA [12] 将相似任务分组，仅在组内进行知识转移。粒子群优化 (PSO) 和差分进化 (DE) 也被集成入 MFEA 框架，形成 MFPSO 和 MFDE [13]。

多种群方法中，如通过自动编码的进化多任务方法采用降噪自动编码器学习任务间映射函数，实现跨任务知识转移 [14]；低置信边界指标用于选择并转移高质量解 [15]；多种知识类型与转移适应框架整合基于进化路径和搜索空间的知识 [16]；基于块级知识转移的差分进化算法将种群划分为块实现知识转移 [17]；领域适应 MTO 采用转移组件分析技术减少解域差异 [18]。

3 本文方法

3.1 本文方法概述

我们假设两个任务 i 和 j 必须同时求解。在知识转移 (KT) 过程中, 利用神经网络 (NNs) 挖掘任务之间的相似性并构建任务间的迁移模型。随后, 通过迁移模型重用任务间的知识, 以预测新的潜在解, 从而促进任务的相互优化。

首先, 我们收集两个任务的解, 并以双向方式将其配对, 作为神经网络的训练数据。其次, 我们初始化并训练神经网络以获得知识转移模型。第三, 利用训练好的神经网络预测目标种群的新解和潜在解, 以辅助其优化。需要注意的是, 为了实现任务 i 和 j 之间的双向知识迁移, 需要训练两组神经网络。神经网络知识迁移 (NNKT) 的四个步骤如下所示。

3.2 收集任务 i 和任务 j 的解:

假设任务 i 的种群有 N_i 个个体, 任务 j 的种群有 N_j 个个体, 其中 $N_i \geq N_j$ 。首先, 我们根据适应度从任务 i 和任务 j 中分别收集前 N_j 个解, 形成集合 A 和 B 。集合 A 和 B 分别作为神经网络的输入和输出训练样本。此外, 训练样本将随时更新, 以确保没有信息延迟。

3.3 配对解 A 和 B :

在神经网络训练中, 配对解是成功实现知识迁移模型的关键步骤。为了提高配对准确性, 本文提出了一种新的适应度归一化相似度配对方法, 通过解的归一化适应度值的距离进行匹配。该方法保留每个任务的个体特征, 从而增强模型的多样性和泛化能力。

我们考虑从任务 B 到任务 A 的解配对。首先, 按适应度值对任务 A 中的解进行排序, 计算归一化适应度值 $f_n(x)$:

$$f_n(x) = \frac{f(x) - f_{A,\min}}{f_{A,\max} - f_{A,\min}}, \quad \forall x \in A$$

其中 $f(x)$ 是 A 解 x 的适应度值, $f_{A,\max}$ 和 $f_{A,\min}$ 分别是任务 A 的最大和最小适应度值。B 中的解按相同方式归一化。

接下来, 任务 A 中的每个归一化解 q , 在 B 中找到与其最相似的解 p :

$$p = \arg \min_{x \in B} |f_n(x) - f_n(q)|.$$

通过配对所有解来形成一个训练集

$$T = \{(p_1, q_1), \dots, (p_{|T|}, q_{|T|})\}$$

用于训练神经网络, 以实现从任务 j 到任务 i 的知识转移, 其中 p_i 作为输入数据, q_i 作为目标输出数据。

3.4 神经网络设计与训练:

配对完成后, 我们开始训练神经网络 (NNs)。假设知识从任务 i 转移到任务 j , 我们采用 D_j 个三层神经网络, 每个神经网络独立预测一个维度的输出。每个神经网络有一个输入

层（包含 D_i 个神经元）、一个隐藏层和一个输出层（包含一个神经元）。这种方法简化了传统神经网络的结构，并使得并行训练成为可能。

每个神经网络的输入相同，但输出是目标维度的预测值。例如，任务 i 和任务 j 分别有 4 维和 3 维时，从任务 i 到任务 j 需要 3 个神经网络，从任务 j 到任务 i 需要 4 个神经网络。

根据 [19]，隐藏层神经元数量为：

$$S_{\text{hidden}} = \max \left(\frac{|T| \cdot (S_{\text{input}} + S_{\text{output}})}{10}, 1 \right).$$

其中， $|T|$ 为训练样本数， S_{input} 和 S_{output} 为输入层和输出层的神经元数。隐藏层激活函数为双曲正切（tanh）：

$$\phi(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

输出层激活函数为线性函数：

$$\phi(x) = x$$

每个神经网络使用均方误差作为训练误差，公式为：

$$E = \frac{1}{N} \sum_{k=1}^N (X_k - Y_k)^T (X_k - Y_k)$$

其中， N 为训练样本数量， X_k 和 Y_k 分别为输入和输出。为提高效率，采用 Levenberg-Marquardt 算法 [20] 进行优化。每个神经网络独立训练 30 个周期。

3.5 知识重用

训练神经网络（NNs）后，我们获得了源任务 i 和目标任务 j 之间的迁移模型，能够转移有用的知识。在此，假设知识转移（KT）发生在任务 i 到任务 j 之间。首先，从任务 i 中选择若干随机解作为神经网络的输入，生成任务 j 中的新解。然后，神经网络预测的新解与任务 j 的原始种群结合，使用差分进化（DE）算法（DE/rand/1 和二项交叉）进行进化。任务 j 中的解也用于优化任务 i ，通过已训练好的神经网络生成新解，经过评估后参与任务 i 的进化。值得注意的是，神经网络需要及时更新和重新训练，以避免信息延迟。为避免过于频繁的重新训练，我们在每 G 代进行一次重新训练，并在新的神经网络训练前，通过当前神经网络在每 g 代（ $g < G$ ）重用其知识。

4 复现细节

4.1 与已有开源代码对比

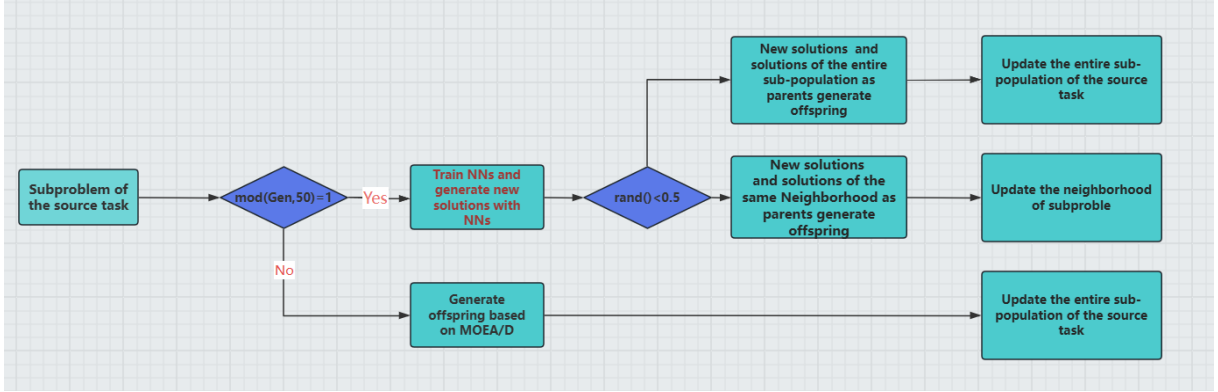


图 1. Flowchart of the main loop of MTEA/D-NN.

开源的代码是用来解决单目标优化问题,我通过将基于神经网络的知识迁移方法和 MOEA/D 框架结合在一起,用来解决多目标多任务优化问题,填补原文的这一不足。通过基于分解的多目标优化算法 (MOEA/D),利用神经网络作为桥梁,训练任务之间的迁移模型,NNKT 能够深入挖掘任务间的相似性,并利用源任务的已有知识预测目标任务的潜在解,从而实现任务间更高质量的协同进化。

在我们改进的算法中,内部邻域构建了同一优化任务中子问题之间的关系。而外部通过神经网络学习构建了不同任务之间子问题的关系,可以视为定义和挖掘不同任务之间关联关系的桥梁。对于一个子问题,其内部邻域由预定义的权重向量确定,在优化过程中是固定的。相比之下,其外部邻域通过任务间的信息交换进行探索和挖掘,这意味着外部邻域会随着进化过程动态变化。如果源任务中某个子问题的优化能为目标任务中某个子问题的优化提供有用信息,那么该子问题在目标任务中的邻域是值得进一步探索的。

为了创建用于训练神经网络的模型数据,采用基于位移的密度估计 (SDE) [21] 计算适应度值:

$$f_{SDE}(x_i) = \min_{x_j \in P \setminus \{i\}} \left(\sum_{m=1}^M (\max(0, f_m(x_j) - f_m(x_i)))^2 \right)$$

其中, $f_m(x_i)$ 表示粒子 x_i 的第 m 个目标值, M 是目标的数量。通过 SDE 可以计算每一个解的适应度值,通过上述的方法进行配对,构造神经网络的训练数据,改进的代码如图 2 所示。

4.2 实验环境搭建

实验平台: MATLAB R2023a 和 MTO-Platform [22]

4.3 界面分析与使用说明

参考 MARLAB 用户使用手册, MTO-Platform 可以从 GitHub 中下载: [GitHub - intLyc/MTO-Platform](https://github.com/intLyc/MTO-Platform)

```

properties (SetAccess = private)
    Beta = 0.4
    F = 0.5
    CR = 0.5
    MuM = 15
end

methods
    function Parameter = getParameter(Algo) [...]

    function Algo = setParameter(Algo, Parameter) [...]

    function run(Algo, Prob)
        % Initialize

        for t = 1:Prob.T [...]

            gen = 0;
            NN_G = 50;

            ntask = Prob.T;
            dims = Prob.T(1);
            fit_list = cell(1,ntask);
            pop_list = cell(1,ntask);
            PopObj = cell(1,ntask);

            net = cell(ntask,1);
            for i =1:ntask [...]

                while Algo.notTerminated(Prob, population) [...]
            end

            function offspring = Generation(Algo, population) [...]
        end
    end
end

```

图 2. 改进的代码

表 1. 与三种比较算法在 CEC2017 测试实例上基于 IGD 指标性能分析

	MO-MFEA	MO-MFEA-II	EMT-ET	MTEA-D-NN
CEC17-MTMO1-CI-HS-T1	2.1585e-02	1.2797e-02	2.3160e-02	1.2440e-02
CEC17-MTMO1-CI-HS-T2	1.1276e-01	8.4297e-02	1.1779e-01	4.4374e-02
CEC17-MTMO2-CI-MS-T1	1.0827e-02	9.7544e-03	8.8648e-03	3.8219e-03
CEC17-MTMO2-CI-MS-T2	1.7789e-02	1.0537e-01	1.4354e-02	1.4961e-01
CEC17-MTMO3-CI-LS-T1	1.0328e-02	7.4612e-03	1.2720e-02	3.2519e+01
CEC17-MTMO3-CI-LS-T2	5.6047e-03	4.6642e-03	4.8565e-03	1.5323e-01
CEC17-MTMO4-PI-HS-T1	1.7825e-02	2.2161e-02	2.4064e-02	8.6701e-03
CEC17-MTMO4-PI-HS-T2	1.6046e+00	1.9349e+00	1.9464e+00	1.9393e+00
CEC17-MTMO5-PI-MS-T1	1.1340e-01	1.3407e-01	3.1989e-02	1.9157e-02
CEC17-MTMO5-PI-MS-T2	4.1053e+02	4.4022e+02	4.6805e-03	1.3190e+03
CEC17-MTMO6-PI-LS-T1	9.8526e-03	8.4520e-03	7.7402e-03	7.4034e-03
CEC17-MTMO6-PI-LS-T2	2.2864e+00	2.0041e+01	2.8866e-01	2.0000e+01
CEC17-MTMO7-NI-HS-T1	4.8796e+01	7.0539e+01	3.5149e+01	9.1561e+02
CEC17-MTMO7-NI-HS-T2	2.0533e-02	1.4243e-02	1.8533e-02	8.3077e-03
CEC17-MTMO8-NI-MS-T1	4.7261e+01	4.1813e+01	4.3292e+01	2.9989e+00
CEC17-MTMO8-NI-MS-T2	1.1355e+00	9.5626e-01	5.4800e-01	2.5119e-02
CEC17-MTMO9-NI-LS-T1	7.3025e-02	6.9136e-02	6.8835e-02	7.6879e-02
CEC17-MTMO9-NI-LS-T2	2.0295e+01	2.0294e+01	2.0193e+01	2.0294e+01

4.4 创新点

KT 被广泛应用于 EMTO 算法中, 以促进不同任务的优化。然而, 大多数 KT 方法只能实现表面的 KT, 缺乏深入挖掘不同任务之间相似性或关系的能力。由于 KT 的效率强烈依赖于任务的相似性, 没有相似性分析的 KT 可能会导致负迁移现象, 从而降低算法的性能。受神经网络强大映射能力的启发, 本文使用 NNKT 方法来深入挖掘任务的相似性。首先, 收集并配对这两项任务的解决方案。然后训练神经网络, 根据配对解决方案发现两个任务之间的相似之处。随后, 应用知识重用, 利用训练好的神经网络根据彼此现有的知识预测新的有前景的解决方案。

5 实验结果分析

将改进的基于神经网络的知识迁移技术用于求解多目标多任务优化问题, 与三种比较前沿的方法进行比较, 即 MO-MFEA [23], MO-MFEA-II [24], EMTIL [25]。其中, MO-MFEA 是多目标多任务 EA 的第一个范式。MO-MFEA-II 是 MO-MFEA 的变体, 它采用数据驱动技术在不同任务之间建立相似的关系模型。该模型用于调整信息传输的频率, 以确保有用信息得到充分利用, 同时丢弃无用信息。EMTIL 中使用增量学习方法来挖掘不同任务之间潜在的有价值信息。上述三种算法中的选择是基于主导策略的。

表 2. Comparison of MTEA/D-NN with other algorithms.

MTEA/D-DN v.s.		MO-MFEA	MO-MFEA-II	EMTIL
IGD	+	13/18	10/18	11/18
	\approx	2/18	4/18	5/18
	-	3/18	4/18	2/18

我们在 9 个多目标多任务基准测试实例 (CEC2017) 和 10 个多目标许多任务基准测试实例 (CEC2019) 上评估了 MTEA/D-DN。对于这九个多任务测试实例, 每个测试实例由两个 MOP 组成, 这些测试实例分别表示为 CIHS、CIMS、CILS、PIHS、PIMS、PILS、NIHS、NIMS 和 NILS [26]。

与多任务测试实例上的最新算法的比较: 表 1 和表 3 分别显示了我们提出的算法及对比算法在九个多任务测试用例上的 IGD+ 均值的统计分析结果。达到最佳性能的算法用红色字体标记。

总体而言, 很明显, 对于 IGD 度量, 统计结果表明我们改进的算法比其他对比算法表现更好。具体来说, 在这九个测试实例中, 我们改进的算法在 IGD 指标获得了 9 个单独任务的最佳结果。在这十个测试实例中, 我们改进的算法在 IGD 指标获得了 14 个单独任务的最佳结果。表 2 和表 4 显示了我们提出的算法和这些最先进的算法在 IGD 度量的 Wilcoxon 秩和检验上的统计分析结果。从该表中可以清楚地看出, 在大多数测试问题上, 我们提出的算法明显优于这些最先进的算法。以 CIHS、CILS、PILS 和 NILS 的测试实例为例, 我们在图 2 中根据中值 IGD+ 度量给出了算法的最终非支配解集。从这个图中可以明显看出, 我们提出的 MTEA/D-NN 总是可以得到很好解的分布。

6 总结与展望

在本文中, 将基于神经网络的知识迁移方法和 MOEA/D 框架结合在一起, 用来解决多目标多任务优化问题, 填补原文的这一不足。通过基于分解的多目标优化算法 (MOEA/D), 利用神经网络作为桥梁, 训练任务之间的迁移模型, NNKT 能够深入挖掘任务间的相似性, 并利用源任务的已有知识预测目标任务的潜在解, 从而实现任务间更高质量的协同进化。将所提的基于神经网络的多目标多任务算法与 IEEE CEC2017 和 IEEE 2019 基准上的一些多任务算法的实验结果进行比较, 证明了 NNMOMTO 的效率和有效性。

尽管目前的工作已经取得了有竞争力的结果, 但仍然存在值得未来研究的挑战。未来, NNMTO 的效率将进一步提高, 特别是解决更具挑战性和复杂性的 MTOP。此外, 神经网络技术将得到改进, 以适应更多的 MTOP 甚至许多任务优化问题。同时, 所提出的 KT 方法将扩展到更复杂的场景, 如约束或超多目标优。此外, NNMTO 将应用于更多现实世界的多任务应用, 如双极平衡问题、机器人手臂控制问题和车辆路线问题。

表 3. 与三种比较算法在 CEC2019 测试实例上基于 IGD 指标性能分析

	MO-MFEA	MO-MFEA-II	EMT-ET	MTEA-D-NN
CEC19-MTMO-CPLX1-T1	1.3655e-02	1.6217e-02	1.9823e-02	4.0998e-03
CEC19-MTMO-CPLX1-T2	6.2201e-02	8.0220e-02	7.1852e-02	3.7323e-02
CEC19-MTMO-CPLX2-T1	8.8651e-03	1.0085e-02	7.9889e-03	4.0581e-03
CEC19-MTMO-CPLX2-T2	6.9842e-02	1.1945e-01	6.0278e-02	8.4177e-02
CEC19-MTMO-CPLX3-T1	9.6271e-02	1.3566e-01	8.2982e-02	8.4895e-02
CEC19-MTMO-CPLX3-T2	4.8329e-02	6.4298e-02	3.6802e-02	2.3625e-02
CEC19-MTMO-CPLX4-T1	9.9477e-02	1.1007e-01	1.9725e-01	3.5894e-01
CEC19-MTMO-CPLX4-T2	1.0798e-01	1.2506e-01	2.0288e-01	1.7687e-01
CEC19-MTMO-CPLX5-T1	3.8976e-02	7.1962e-02	4.3101e-02	2.1625e-02
CEC19-MTMO-CPLX5-T2	1.4807e-01	1.4652e-01	1.3643e-01	1.1169e-01
CEC19-MTMO-CPLX6-T1	7.6047e-02	7.4781e-02	7.0012e-02	3.7310e-02
CEC19-MTMO-CPLX6-T2	1.2907e-01	1.3177e-01	1.2480e-01	2.1703e-01
CEC19-MTMO-CPLX7-T1	6.8382e-02	7.5353e-02	6.7264e-02	4.7856e-02
CEC19-MTMO-CPLX7-T2	5.6506e-02	6.5167e-02	6.5676e-02	3.6547e-02
CEC19-MTMO-CPLX8-T1	4.9920e-02	9.2368e-02	6.7359e-02	2.8296e-02
CEC19-MTMO-CPLX8-T2	2.0408e-01	4.0021e-01	1.4700e-01	6.1941e-02
CEC19-MTMO-CPLX9-T1	1.1678e+00	6.1158e+01	5.4065e+01	3.7454e-01
CEC19-MTMO-CPLX9-T2	3.2802e-01	2.1587e-01	2.0008e-01	1.6530e-01

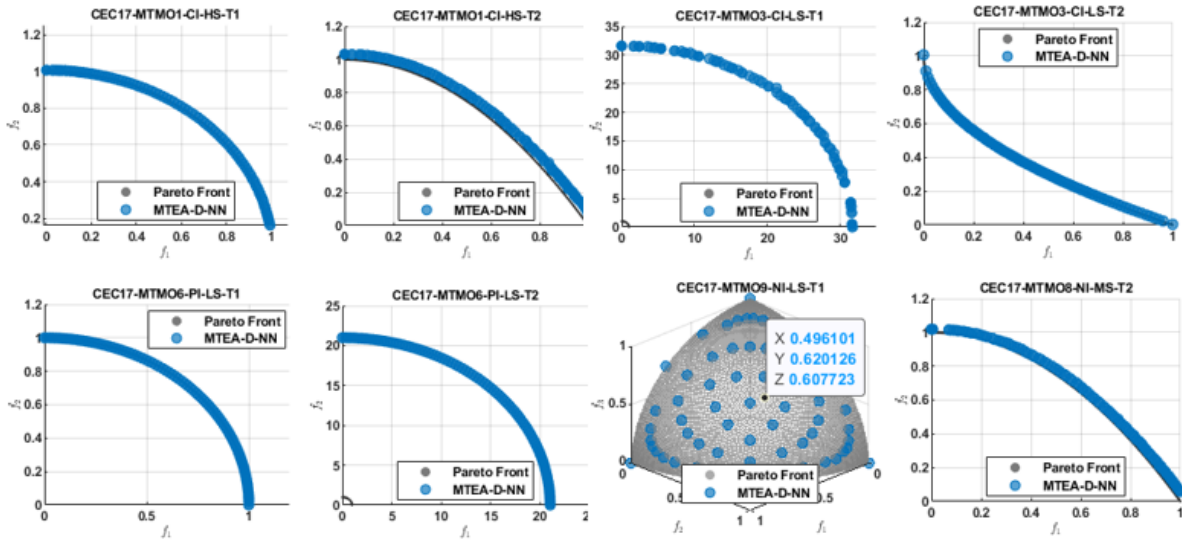


图 3. 实验结果示意

表 4. Comparison of MTEA/D-DN with other algorithms.

MTEA/D-DN v.s.		MO-MFEA	MO-MFEA-II	EMTIL
IGD	+	15/18	15/18	14/18
	\approx	2/18	2/18	2/18
	-	1/18	1/30	2/18

参考文献

- [1] Ke Li, Sam Kwong, and Kalyanmoy Deb. A dual-population paradigm for evolutionary multiobjective optimization. *Information Sciences*, 309:50–72, 2015.
- [2] Zhengping Liang, Hao Dong, Cheng Liu, Weiqi Liang, and Zexuan Zhu. Evolutionary multitasking for multiobjective optimization with subspace alignment and adaptive differential evolution. *IEEE Transactions on Cybernetics*, 52:2096–2109, 2020.
- [3] Weijing Dai, Zhenkun Wang, and Ke Xue. System-in-package design using multi-task memetic learning and optimization. *Memetic Computing*, 14:45–59, 2022.
- [4] Xiaoming Xue, Kai Zhang, Kay Chen Tan, Liang Feng, Jian Wang, Guodong Chen, Xinggang Zhao, Liming Zhang, and Jun Yao. Affine transformation-enhanced multifactorial optimization for heterogeneous problems. *IEEE Transactions on Cybernetics*, 52(7):6217–6231, 2020.
- [5] Abhishek Gupta, Yew-Soon Ong, and Liang Feng. Multifactorial evolution: Toward evolutionary multitasking. *IEEE Transactions on Evolutionary Computation*, 20(3):343–357, 2015.
- [6] Lei Zhou, Liang Feng, Kay Chen Tan, Jinghui Zhong, Zexuan Zhu, Kai Liu, and Chao Chen. Toward adaptive knowledge transfer in multifactorial evolutionary computation. *IEEE transactions on cybernetics*, 51(5):2563–2576, 2020.
- [7] Jian-Yu Li, Zhi-Hui Zhan, Kay Chen Tan, and Jun Zhang. A meta-knowledge transfer-based differential evolution for multitask optimization. *IEEE Transactions on Evolutionary Computation*, 26(4):719–734, 2021.
- [8] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989.
- [9] KS Tang, CY Chan, KF Man, and S Kwong. Genetic structure for nn topology and weights optimization. In *First International Conference on Genetic Algorithms in Engineering Systems: Innovations and Applications*, pages 250–255. IET, 1995.

- [10] Kavitesh Kumar Bali, Yew-Soon Ong, Abhishek Gupta, and Puay Siew Tan. Multifactorial evolutionary algorithm with online transfer parameter estimation: Mfea-ii. *IEEE Transactions on Evolutionary Computation*, 24(1):69–83, 2019.
- [11] Jinliang Ding, Cuie Yang, Yaochu Jin, and Tianyou Chai. Generalized multitasking for evolutionary optimization of expensive problems. *IEEE Transactions on Evolutionary Computation*, 23(1):44–58, 2017.
- [12] Jing Tang, Yingke Chen, Zixuan Deng, Yanping Xiang, and Colin Paul Joy. A group-based approach to improve multifactorial evolutionary algorithm. In *IJCAI*, pages 3870–3876, 2018.
- [13] L Feng, W Zhou, L Zhou, SW Jiang, JH Zhong, BS Da, ZX Zhu, and Y Wang. An empirical study of multifactorial pso and multifactorial de. In *2017 IEEE Congress on evolutionary computation (CEC)*, pages 921–928. IEEE, 2017.
- [14] Liang Feng, Lei Zhou, Jinghui Zhong, Abhishek Gupta, Yew-Soon Ong, Kay-Chen Tan, and Alex Kai Qin. Evolutionary multitasking via explicit autoencoding. *IEEE transactions on cybernetics*, 49(9):3457–3470, 2018.
- [15] Zhenzhong Wang, Lulu Cao, Liang Feng, Min Jiang, and Kay Chen Tan. Evolutionary multitask optimization with lower confidence bound-based solution selection strategy. *IEEE Transactions on Evolutionary Computation*, 2024.
- [16] Yanchi Li and Wenying Gong. Multiobjective multitask optimization with multiple knowledge types and transfer adaptation. *IEEE Transactions on Evolutionary Computation*, 2024.
- [17] Yi Jiang, Zhi-Hui Zhan, Kay Chen Tan, and Jun Zhang. Block-level knowledge transfer for evolutionary multitask optimization. *IEEE Transactions on Cybernetics*, 54(1):558–571, 2023.
- [18] Xiaoling Wang, Qi Kang, MengChu Zhou, Siya Yao, and Abdullah Abusorrah. Domain adaptation multitask optimization. *IEEE Transactions on Cybernetics*, 53(7):4567–4578, 2022.
- [19] Martin T Hagan, Howard B Demuth, and Mark Beale. *Neural network design*. PWS Publishing Co., 1997.
- [20] Martin T Hagan and Mohammad B Menhaj. Training feedforward networks with the marquardt algorithm. *IEEE transactions on Neural Networks*, 5(6):989–993, 1994.
- [21] Miqing Li, Shengxiang Yang, and Xiaohui Liu. Shift-based density estimation for pareto-based algorithms in many-objective optimization. *IEEE Transactions on Evolutionary Computation*, 18(3):348–365, 2014.

- [22] Yanchi Li, Wenyin Gong, Fei Ming, Tingyu Zhang, Shuijia Li, and Qiong Gu. Mtop: A matlab optimization platform for evolutionary multitasking. *arXiv preprint arXiv:2312.08134*, 2023.
- [23] Abhishek Gupta, Yew-Soon Ong, Liang Feng, and Kay Chen Tan. Multiobjective multifactorial optimization in evolutionary multitasking. *IEEE transactions on cybernetics*, 47(7):1652–1665, 2016.
- [24] Kavitesh Kumar Bali, Abhishek Gupta, Yew-Soon Ong, and Puay Siew Tan. Cognizant multitasking in multiobjective multifactorial evolution: Mo-mfea-ii. *IEEE transactions on cybernetics*, 51(4):1784–1796, 2020.
- [25] Jiabin Lin, Hai-Lin Liu, Bing Xue, Mengjie Zhang, and Fangqing Gu. Multiobjective multitasking optimization based on incremental learning. *IEEE Transactions on Evolutionary Computation*, 24(5):824–838, 2019.
- [26] Yuan Yuan, Yew-Soon Ong, Liang Feng, A Kai Qin, Abhishek Gupta, Bingshui Da, Qingfu Zhang, Kay Chen Tan, Yaochu Jin, and Hisao Ishibuchi. Evolutionary multitasking for multiobjective continuous optimization: Benchmark problems, performance metrics and baseline results. *arXiv preprint arXiv:1706.02766*, 2017.