

中图分类号: TP371
学科分类号: 080605

论文编号: 1028716 22-S059

硕士学位论文

基于邻域结构迁移的多目标组合优化 算法的研究

研究生姓名	王康
学科、专业	计算机科学与技术
研究方向	计算智能理论与应用
指导教师	蔡昕烨 副教授

南京航空航天大学

研究生院 计算机科学与技术学院

二〇二一年十一月

Nanjing University of Aeronautics and Astronautics
The Graduate School
College of Computer Science and Technology

Research on Multi-Objective Combination Optimization Algorithm Based on Neighbor Structure Transfer

A Thesis in
Computer Science and Technology

by

Wang Kang

Advised by

Associate Prof. Cai Xinye

Submitted in Partial Fulfillment

of the Requirements

for the Degree of

Master of Engineering

November, 2021

承诺书

本人声明所呈交的硕士学位论文是本人在导师指导下进行的研究工作及取得的研究成果。除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得南京航空航天大学或其他教育机构的学位或证书而使用过的材料。

本人授权南京航空航天大学可以将学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存、汇编学位论文。

（保密的学位论文在解密后适用本承诺书）

作者签名：_____

日 期：_____

摘 要

基于邻域结构迁移的多目标组合优化算法的研究。

关键词：多目标优化，局部搜索，进化迁移优化，邻域结构，最小生成树，邻域结构迁移

ABSTRACT

Research on Multi-Objective Combination Optimization Algorithm Based on Neighbor Structure Transfer.

Keywords: Multi-objective Optimization, Local Search, Evolutionary Transfer Optimization, Neighborhood Structure, Minimum Spanning Tree, Neighborhood Structure Transfer

目 录

第一章 绪论	1
1.1 研究背景及意义	1
1.2 国内外研究进展	2
1.3 本文主要研究内容	4
1.4 本文组织结构	4
第二章 背景介绍	6
2.1 多目标优化问题	6
2.1.1 问题定义	6
2.1.2 相关概念	6
2.2 多目标组合优化算法	9
2.2.1 进化算法	9
2.2.2 基于分解的多目标进化算法	11
2.3 局部搜索	16
2.3.1 基本概念	16
2.3.2 邻域结构	18
2.3.3 邻域动作	19
2.4 进化迁移优化	20
2.5 性能评价指标	21
2.5.1 超体积度量指标	21
2.5.2 反向世代距离指标	22
2.5.3 集合覆盖度量指标	23
2.5.4 距离趋近度指标	24
2.5.5 最优边遗失率指标	24
2.6 测试问题	24
2.6.1 TSP	25
2.6.2 MOTSP	25
2.6.3 MCMST	26
2.7 本章小结	26

第三章 基于最小生成树和欧拉回路的邻域结构生成方法	27
3.1 引言	27
3.2 研究动机.....	28
3.3 算法框架.....	28
3.4 实验结果与分析	28
3.5 本章小结.....	28
参考文献.....	29
致谢.....	33
在学期间的研究成果及学术论文情况	34

图表清单

图 2.1	MOP 中相关概念示意图	8
图 2.2	理想点和边界点示意图	8
图 2.3	多目标进化算法流程图	11
图 2.4	加权和法 (WS) 等高线示意图	12
图 2.5	切比雪夫法 (TCH) 等高线示意图	13
图 2.6	基于惩罚的边界交叉法 (PBI) 等高线示意图	14
图 2.7	邻域结构示意图	18
图 2.8	k-opt 示意图	20
图 2.9	跨问题知识迁移示意图	21
图 2.10	HV 指标示意图	22
图 2.11	IGD 指标示意图	23

注释表

O	时间（空间）复杂度	m	目标数量
$F(\mathbf{x})$	\mathbf{x} 对应的目标函数	$f_i(\mathbf{x})$	第 i 个目标函数
Ω	决策空间	\mathbb{R}^n	n 维实数空间
$<$	支配（强支配）	\nless	非支配
\leq	弱支配	\mathbf{x}^*	Pareto 最优解
PS	Pareto 最优解集	PF	Pareto 前沿
\mathbf{z}^*	理想点，参考点	\mathbf{z}^{nad}	边界点
λ	权重向量	λ^i	第 i 权重向量
λ_j^i	第 i 个权重向量的第 j 分量	\mathcal{EP}	外部集（精英种群）
\mathcal{SP}	子问题	\mathcal{S}	搜索（解）空间
\mathcal{X}	有限（可数无限）变量集	s^*	局部最优解
\mathcal{D}	决策域	\mathcal{C}	约束域
\mathcal{A}	接受策略	\mathcal{N}	邻域结构

缩略词

缩略词	英文全称
MOPs	Multi-objective Optimization Problems
CMOPs	Combinatorial Multi-objective Optimization Problems
LS	Local Search
CO	Combinatorial Optimization
TL	Transfer Learning
ML	Machine Learning
SD	Source Domains
TD	Target Domains
EA	Evolutionary Algorithm
ETO	Evolutionary Transfer Optimization
NS	Neighborhood Structure
MOEA	Multi-objective Evolutionary Algorithm
NSGA	Non-dominated Sorting Genetic Algorithm
NSGA-II	A Fast and Elitist Multi-objective Genetic Algorithm-II
SPEA	Strength Pareto Evolutionary Algorithm
ENS	Efficient Approach to Non-dominated Sorting
ENLU	Efficient Non-domination Level Update Method
IGD	Inverted Generational Distance
HV	Hypervolume
IBEA	The Indicator-based Evolutionary Algorithm
HypE	Hypervolume Estimation Algorithm
MOEA/D	Multi-objective Evolutionary Algorithm based on Decomposition
MOGLS	Multi-objective Genetic Local Search
GLS	Genetic Local Search
PAES	Pareto Archived Evolution Strategy
PLS	Pareto Local Search
MPLS	Multi-restart Pareto Local Search

TSP	Traveling Salesman Problem
VRP	Vehicle Routing Problem
NDSset	Non-dominated Solution Set
PS	Pareto Set
PF	Pareto Front
NDSort	Non-dominated Sort
WS	Weight Sum
TCH	Tchebycheff
PBI	Penalty-based Boundary Intersection
EP	External(Elite) Population
SP	Sub-problem
GS	Global Search
MOTSP	Multi-Objective Traveling Salesman Problem
CSet	Candidate Set
IGD	Inverted Generational Distance
MCMST	Multi-criteria Minimum Spanning Tree
MST	Minimum Spanning Tree
LKH	Lin Kernighan Helsgaun

第一章 绪论

1.1 研究背景及意义

现实生活中,许多问题都是由相互冲突和影响的多个目标组成。人们经常会遇到在给定条件下,尽可能的使多个目标同时达到最佳的优化问题。优化问题存在的优化目标超过一个并需要同时处理,就成为了多目标优化问题 (Multi-objective Optimization Problems, MOPs)。当问题的变量域是有限集合时,我们称该类问题为多目标组合优化问题 (Combinatorial Multi-objective Optimization Problems, CMOPs)。多目标组合优化问题在现实世界中大量存在,它已广泛应用于大部分行业,包括但不限于运输、能源、金融和调度。由于大多数多目标组合优化问题是 NP 难问题,传统的确定性算法难以对这类问题在给定时间内进行求解,因此如何设计相应的多目标组合优化算法来解决这类问题,越来越受到学者们的关注,也有着重要的研究意义。

局部搜索 (Local Search, LS) 是解决组合优化 (Combinatorial Optimization, CO) 的一种有效方法,但是局部搜索的搜索空间可能很大,而且在给定的时间内,对于每一个组合,局部搜索都需要进行一次搜索,这种搜索空间可能会导致算法的运行时间很长。但是,我们不仅可以通过各种高效的搜索策略来减小局部搜索的搜索空间,而且还可以通过对具有显性结构化搜索空间的组合优化问题进行搜索空间的剪枝预处理,从而减小搜索空间,达到提高算法的运行效率的目的。

此外,迁移学习 (Transfer Learning, TL) 是一种利用跨问题领域的有用特征数据来提高学习性能的机器学习 (Machine Learning, ML) 方法。TL 利用从包含大量高质量信息的源域 (Source Domains, SD) 中获得的知识来改进只包含少量知识的目标域 (Target Domains, TD) 的学习模型^[1]。然而,现在的 TL 研究主要局限于机器学习应用,如计算机视觉、自然语言处理和语音识别等。近年来,在进化优化的背景下,基于 TL 中不同领域的信息迁移这一特征,越来越多的研究者开始将进化算法 (Evolutionary Algorithm, EA) 与 TL 结合起来,提出了进化迁移优化 (Evolutionary Transfer Optimization, ETO) 这一新的范式。ETO 通过将 EA 求解器与知识学习和跨领域信息迁移相结合,从而实现更好的优化效率和性能。

当前的研究中,很多工作都是基于启发式的方法在有限时间内来对优化问题进行优化,并给出一个或一组近似的解决方案。鉴于此,本文将在基于分解的多目标优化框架上,融入局部搜索和领域信息迁移的思想。并且,针对具体问题,设计高质量的邻域结构 (Neighborhood Structure, NS),同时将这些邻域结构作为 ETO 中能够被迁移的元信息,从而使得设计的算法能够在提升优化效率的同时能够获得更好的优化效果。

1.2 国内外研究进展

从多目标组合优化问题被提出之后，国内外学者设计了大量启发式算法来对各种多目标组合优化问题进行求解，并取得十分显著的成果。其大多数是基于进化算法，可主要分为三类：基于 Pareto 支配关系，基于指标和基于分解的多目标优化算法：

- **基于 Pareto 支配关系的多目标优化算法：**这类算法的核心思想是通过支配关系和密度估计来比较解之间的优劣关系。1993 年，随着 Srinivas 和 Deb 等人将 Pareto 支配关系融入到多目标进化算法（Multi-objective Evolutionary Algorithm, MOEA）中，提出了基于非支配排序的进化算法（Non-dominated Sorting Genetic Algorithm, NSGA）^[2]，标志着多目标进化算法进入了一个新的阶段，这也是多目标进化算法框架的一个重大进展。由于 NSGA 存在在非支配分层上的计算复杂度高（ $O(N^3)$ ）、无精英保留策略等问题，为此，Deb 等人在 NSGA 的基础上，于 2002 年提出了带精英策略的非支配排序的进化算法（A Fast and Elitist Multi-objective Genetic Algorithm-II, NSGA-II）^[3]，较好的弥补了 NSGA 的缺点。同时，强度 Pareto 进化算法（Strength Pareto Evolutionary Algorithm-II, SPEA2）^[4]也是一种性能比较好的多目标进化算法，其使用了一种外部保存集的机制来保证解的多样性。为了进一步减少非支配排序的时间复杂度，Zhang 等人提出了一种新的非支配排序算法 ENS（Efficient Approach to Non-dominated Sorting）^[5]，Li 等人提出了一种稳态的替换策略 ENLU（Efficient Non-domination Level Update Method）^[6]。
- **基于指标的多目标优化算法：**这类算法的核心思想是通过性能指标（IGD（Inverted Generational Distance）^[7]，HV（Hypervolume）^[8]）对解进行选择。2004 年，Zitzler 等人提出了基于指标的多目标进化算法，该算法主要是基于超体积指标（HV）来对个体进行选择，代表算法有 IBEA（The Indicator-based Evolutionary Algorithm）^[9]、HypE（Hypervolume Estimation Algorithm）^[10]。由于涉及到 HV 的计算，如果问题的目标数变多时，计算复杂度也成指数增高。
- **基于分解的多目标优化算法：**这类算法的核心思想是通过将一个多目标问题分解成一组单目标优化问题，然后对这些单目标问题进行求解。2007 年，Zhang 等人系统的将分解的思想融入到进化多目标优化算法中，提出了基于分解的多目标进化算法（Multi-objective Evolutionary Algorithm based on Decomposition, MOEA/D）^[11]，这一工作的出现使得分解策略被广泛的应用于多目标进化算法。近年来，MOEA/D 得到广泛应用，成为了最具有影响力的 MOEA 之一。

许多多目标启发式方法的一个共同点是使用局部搜索（LS）技术，目前，大部分的多目标组合优化算法中使用 LS 来产生新的个体（解），其原理就是对现有的个体使用邻居搜索策略，从而产生一个或多个新的个体，将产生的高质量的个体替换当前的个体，以此不断迭代来逼近

目标问题的最优解。LS 既可以在现有的算法内部混合使用，也可以作为其主要组成部分。

进化算法 (EA) 是进化计算的一个子类，是基于群体搜索策略和群体中个体之间的信息交换的一类元启发式方法，通常用于解决多目标优化问题。EA 通常使用迭代的方式模拟世代进化，使用交叉和变异等算子来改善种群性能 (种群进化)。LS 算子在多目标 EA 中的作用就是替代交叉和变异等算子，以实现种群的进化。Ishibuchi 和 Murata 于 1996 年首次提出了多目标遗传局部搜索算法 (Multi-objective Genetic Local Search, MOGLS)^[12]，该算法将遗传算法和单目标的局部搜索算子进行融合，通过对每次迭代生成的个体进行局部搜索来产生新个体，Jaszkiewicz 于 2002 年提出了另外一种遗传局部搜索算法 (Genetic Local Search, GLS)^[13]，该算法使用从一组可能的权重向量中随机选择一个权重向量对个体进行局部搜索，与前者区别在于选择个体的方式不同。Knowles 和 Corne 提出了一种没有交叉和变异算子、仅仅依赖 LS 算子的 Pareto 解集进化策略 (Pareto Archived Evolution Strategy, PAES)^[14,15]。Paquete^[16] 和 Angel^[17] 同时分别提出了第一个独立的局部搜索算法：Pareto 局部搜索 (Pareto Local Search, PLS) 和双标准局部搜索，这两种算法非常相似，都被统称为 PLS。与 EA 中的种群不同，在 PLS 算法中，所有的个体组成的集合称为存档 (Archive)，并且存档集始终由 Pareto 解集组成。在每次迭代中，PLS 算法都会从存档集中选择没有被搜索过的个体，通过邻域动作，产生其邻居解，并用这些邻居解来更新存档集。除了基于支配关系和基于分解的局部搜索算法之外，还有基于指标的多目标局部搜索算法^[18]，它使用性能指标 (如 HV, IGD) 来选择优质的邻居个体。Drugan 等人提出了 PLS 的多重启算法 (Multi-restart Pareto Local Search, MPLS)^[19]。MPLS 是建立在 PLS 的基础之上的，MPLS 中存档集中的每个个体都会被标记是否休眠，休眠状态的个体不能再次被探索。当个体被探索过后，该个体会被标记为休眠状态，在运行给定迭代次数的 PLS 之后，MPLS 不再考虑新个体，而是从存档中随机选择一个或多个个体，并将其设置为活跃状态，重新启动。

近年来，跨领域的方法相融合的思想越来越受到学者们的关注。就如在机器学习 (ML) 和优化两个领域，有很多研究将迁移学习的思想运用到优化领域当中。TL 是一种利用跨问题领域的知识来提高算法性能的机器学习方法，在进化优化的背景下，基于 TL 中不同领域的信息迁移这一特征，越来越多的研究者开始将 EA 与 TL 结合起来，提出了进化迁移优化 (ETO) 这一新的范式。ETO 通过将 EA 求解器与跨领域信息迁移相结合，从而实现更好的优化效率和性能。

Feng 等人于 2017 年提出可一种具有跨异构问题学习能力，能够自动编码进化搜索的算法框架^[20]，该算法能够针对 MOP，通过将当前的解和过去的解建立跨问题的映射，从而将过去解中蕴含的信息转移到现在的解中，从而达到知识迁移的作用。Yang 等人于 2019 年提出了一种用于解决 MOP 的 ETO 算法^[21]，该算法有两个辅助代理模型，一个是粗粒度代理模型，一个是细粒度代理模型，粗粒度代理模型旨在引导算法在整个搜索空间中快速找到一个蕴含有用信息的子区域，细粒度代理模型则侧重于利用前一个代理模型所寻找的子区域来搜索优质的解，从而达到优化问题的目的。2020 年，Lin 等人提出了一种寻找能够为迁移提供有效信息的、高质

量的解的方法^[22]，在该方法中，从能够实现正迁移的解的邻居中选择可以被用来迁移的解，从而增强其解决任务的收敛性。Liang 等人提出了一种两阶段自适应的知识迁移方法^[23]，该方法通考虑了 EA 中 MOP 的种群分布，从反映整体搜索趋势的概率模型中提取知识来加速和提高任务的收敛性能，在第一转移阶段，使用自适应权重来调整个体搜索的步长，可以减少负转移的影响，在第二阶段，进一步动态调整个体的搜索范围，可以提高种群的多样性，有利于跳出局部最优。

显然，随着 MOP 中目标数量的增加，搜索空间呈指数增长，将被转移的信息也急剧增加^[24]。因此，正确的构建源域和目标域之间的映射和怎样实现源域和目标域之间的有用知识的正迁移更具挑战性。更深入地分析什么信息是有用的信息以及这些信息如何与跨问题域的多个目标相关联，可以帮助我们更好地了解 ETO 如何以及何时在 MOP 中表现良好。

1.3 本文主要研究内容

本论文主要研究使用基于分解的多目标组合优化算法解决超多目标组合优化问题。为解决现有的基于分解的算法在超多目标分解后搜索空间剧增的问题，我们使用局部搜索技术，设计适配问题的邻域结构，提出了基于最小生成树和欧拉回路的邻域结构生成算法。并且与 ML 领域相结合，将迁移学习中的知识迁移的技术应用于多目标组合优化算法当中，提出了基于邻域结构迁移的多目标组合优化算法。本文的主要研究内容如下：

1. 介绍现常用的多目标优化算法和进化算法，分析其在多目标组合优化问题上的优劣。然后介绍常被嵌入到算法中的局部搜索算子，并分别介绍了局部搜索算子中的邻域结构和邻域动作，并且引出了近年来备受关注的跨领域结合的进化迁移优化，这为后续章节的研究提供了一个基础理论支撑。
2. 本论文基于分解的多目标优化框架，使用局部搜索技术，针对具体问题，从问题本身出发，挖掘问题本身的信息，构建邻域结构，提出了相应的邻域结构生成算法，该算法能够快速地为路径类问题（TSP，VRP 等）生成高质量的邻域结构，供局部搜索使用，从而提升问题的优化效率。
3. 基于上述邻居结构，结合 ETO 中信息迁移的思想，提出了基于邻域结构迁移的多目标组合优化算法。两目标相似度模型粗粒度地确定能被迁移的邻域结构的范围，勘探-开发机制细粒度地动态改变邻域结构被选择用来迁移的概率，从而提高在算法中发生正迁移的比重，以达到更好的收敛效果和优化质量。

1.4 本文组织结构

本论文围绕基于分解的多目标优化框架和用于局部搜索的邻域结构，以及基于邻域结构迁移的多目标组合优化算法进行研究。通过分析现有算法的不足，并且针对具有不同规模、类型

和特征的组合优化问题，进行算法设计及实验论证分析。同时，通过对比现有的相关算法来验证设计的算法的合理性及有效性。本论文总共分为五章节，全文组织结构如下：

第一章首先介绍了本文的研究背景及意义，梳理了多目标优化算法，局部搜索和进化迁移等领域的相关发展历程，最后阐述了本文的主要研究内容及组织结构。

第二章首先介绍了多目标优化问题的定义及相关概念，对一些著名的多目标进化算法进行了介绍，并且对基于分解的多目标进化算法进行了详细的阐述。然后介绍了能够嵌入多目标进化算法中的局部搜索算子，并且介绍了局部搜索中的两个重要的概念：邻域结构和邻域动作。接着，介绍了进化迁移优化的一些思想，以及在多目标优化算法中使用进化迁移期望能够解决的问题。最后，介绍了本文中可能会用到的性能评价指标和测试问题。本章的主要目的是为后续章节提供一定的基础理论支撑。

第三章提出了邻域结构生成算法

第四章提出了基于邻域结构迁移的多目标进化算法

第五章对本文的主要工作进行总结，并分析本文工作的不足，为后续研究工作提供进一步可能的研究方向

第二章 背景介绍

2.1 多目标优化问题

多目标优化是指使多个目标在给定条件下同时尽可能达到最佳。多目标优化问题是具有需要被同时优化的多个目标的问题，这些被优化的目标包含最大化目标、最小化目标或者两者混合目标。在实际处理问题时，为了简化被优化的问题，可以将最大化（最小化）问题取反，使得所有的优化目标都转换成最小化（最大化）问题。在本文中，我们统一用最小化问题来描述多目标优化问题。

2.1.1 问题定义

一般地，一个含有 n 个决策变量， m 个目标变量的最小化多目标优化问题可以由如下数学形式描述：

$$\begin{aligned} \text{minimize} \quad & F(\mathbf{x}) = (f_1(\mathbf{x}), \dots, f_m(\mathbf{x}))^T, \\ \text{s.t.} \quad & \mathbf{x} \in \Omega. \end{aligned} \quad (2.1)$$

其中， $\mathbf{x} = (x_1, x_2, \dots, x_n)^T$ 表示决策变量， $\Omega \subset \mathbb{R}^n$ 是 n 维的决策空间， $F : \Omega \rightarrow \mathbb{R}^m$ 为决策域到目标域的映射， \mathbb{R}^m 为 m 维的目标空间。当 Ω 为实数集时，问题 (2.1) 被称为多目标连续优化问题，当 Ω 为有限集合时，问题 (2.1) 被称为多目标组合优化问题。

2.1.2 相关概念

由于多目标优化问题的复杂性，在求解时，我们通常会将各个优化目标的解进行处理，最终得到一组均衡解使得各个优化目标的解尽可能地能被决策者接受。这里我们首先引入可行解的概念。

定义 2.1（可行解）： 满足约束条件的解 $\mathbf{x} \in \Omega$ 称为可行解。

在多目标优化问题中，由于存在多个目标函数，可行解之间无法通过单一的数值比较来进行优劣的评价，因此，我们引入由经济学家 Vilfredo Pareto 于 1987 年提出的 Pareto 支配的概念，Pareto 支配的核心思想是通过比较解之间对应位置的决策变量的值来综合判断其优劣。

定义 2.2（Pareto 支配）： 设 $\mathbf{x}_1, \mathbf{x}_2 \in \Omega$ 是 MOP 的任意两个不同的可行解，若同时满足下列两个条件：

1. 对于所有的 m 个优化目标， \mathbf{x}_1 都不比 \mathbf{x}_2 差，即 $\forall i \in \{1, \dots, m\}$ ，有 $f_i(\mathbf{x}_1) \leq f_i(\mathbf{x}_2)$ ；

2. 至少存在一个子目标, 使得 \mathbf{x}_1 比 \mathbf{x}_2 好。即 $\exists i \in \{1, \dots, m\}$, 使 $f_i(\mathbf{x}_1) < f_i(\mathbf{x}_2)$ 。

则称 \mathbf{x}_1 为非支配的 (非劣的或占优的), \mathbf{x}_2 为被支配的。称 \mathbf{x}_1 Pareto 支配 \mathbf{x}_2 , 记为 $\mathbf{x}_1 < \mathbf{x}_2$, 其中 “ $<$ ” 表示支配关系。上述定义的支配关系是针对决策空间的, 类似地, 目标空间的支配关系可以定义为: 设 $\mathbf{u}, \mathbf{v} \in \mathbb{R}^m$, 称 $\mathbf{u} < \mathbf{v}$, 当且仅当 $\forall i \in \{1, \dots, m\} u_i \leq v_i$, $\exists j \in \{1, \dots, m\} u_j < v_j$ 。

定义 2.3 (弱支配): 若有 $\mathbf{x}_1, \mathbf{x}_2 \in \Omega$, 对于 $\forall i \in \{1, \dots, m\}$, 有 $f_i(\mathbf{x}_1) \leq f_i(\mathbf{x}_2)$, 且 $\exists j \in \{1, \dots, m\}$, 使 $f_j(\mathbf{x}_1) < f_j(\mathbf{x}_2)$ 成立, 则称 \mathbf{x}_1 弱支配 \mathbf{x}_2 , 记作 $\mathbf{x}_1 \leq \mathbf{x}_2$ 。

定义 2.4 (强支配): 若有 $\mathbf{x}_1, \mathbf{x}_2 \in \Omega$, 对于 $\forall i \in \{1, \dots, m\}$, 有 $f_i(\mathbf{x}_1) < f_i(\mathbf{x}_2)$, 则称 \mathbf{x}_1 强支配 \mathbf{x}_2 , 记作 $\mathbf{x}_1 < \mathbf{x}_2$ 。

定义 2.5 (非支配): 设 $\mathbf{u}, \mathbf{v} \in \mathbb{R}^m$, 若 \mathbf{u} 不支配 \mathbf{v} , 且 \mathbf{v} 不支配 \mathbf{u} , 则称 \mathbf{u} 和 \mathbf{v} 是非支配的, 记作 $\mathbf{u} \not< \mathbf{v}$ 或 $\mathbf{v} \not< \mathbf{u}$ 。

定义 2.6 (非支配解集 (Non-dominated Solution Set, NDSset)): 一个可行解集中任意可行解 $\mathbf{x}_1, \mathbf{x}_2 \in \Omega$, 通过对应的函数 F 映射到目标域中所得到的目标向量 $F(\mathbf{x}_1), F(\mathbf{x}_2) \in \mathbb{R}^m$, 有 $F(\mathbf{x}_1) \not< F(\mathbf{x}_2)$, 且 $F(\mathbf{x}_2) \not< F(\mathbf{x}_1)$, 则该可行解集称为非支配解集 (NDSset)。

定义 2.7 (Pareto 最优解): 设 $\mathbf{x}^* \in \Omega$ 为一个可行解, 若 $\nexists \mathbf{x} \in \Omega$, 使得 $\mathbf{x} < \mathbf{x}^*$, 则称 \mathbf{x}^* 为 Pareto 最优解。

定义 2.8 (Pareto 最优解集 (Pareto Set, PS)): 所有满足定义 (2.7) 的可行解 \mathbf{x}^* 所构成的解集, 称为 Pareto 最优解集 (PS)。

定义 2.9 (Pareto 前沿 (Pareto Front, PF)): Pareto 最优解集中的解通过对应的函数 F 映射到目标域中所得到的目标向量的集合称为 Pareto 前沿 (PF), 即

$$PF = \{F(\mathbf{x}) : \Omega \rightarrow \mathbb{R}^m \mid \mathbf{x} \in PS\}. \quad (2.2)$$

为了方便地理解多目标优化, 图 2.1 展示了上述定义中的相关概念。以两个决策变量的两目标最小化 MOP 为例, 图 2.1a 为决策空间中解集的示意图, 图 2.1b 为目标空间中解集的示意图, 且决策空间的解与目标空间中的解一一对应。其中, 在决策空间中, 虚线为 PS, 图中共六个可行解, 其中 $\mathbf{a}, \mathbf{b}, \mathbf{c} \in PS$, 均为 Pareto 最优解, 它们互不支配, 可称为非支配解集 (NDSset)。在目标空间中, 虚线为 PF, 其中 $F(\mathbf{a}), F(\mathbf{b}), F(\mathbf{c}) \in PF$, 包含的支配关系有: $F(\mathbf{a}) < F(\mathbf{d}), F(\mathbf{b}) < F(\mathbf{e}), F(\mathbf{c}) < F(\mathbf{f})$, 其余可行解之间均为非支配关系。

设计多目标优化算法时, 在目标空间中, 还有两个比较重要的概念: 理想点 (Ideal Point) 和边界点 (Nadir Point), 他们限定了 PF 的空间范围。

定义 2.10 (理想点 (Ideal Point)): 目标空间中, 所有可行解 $\mathbf{x} \in \Omega$ 在各个目标上的最小值

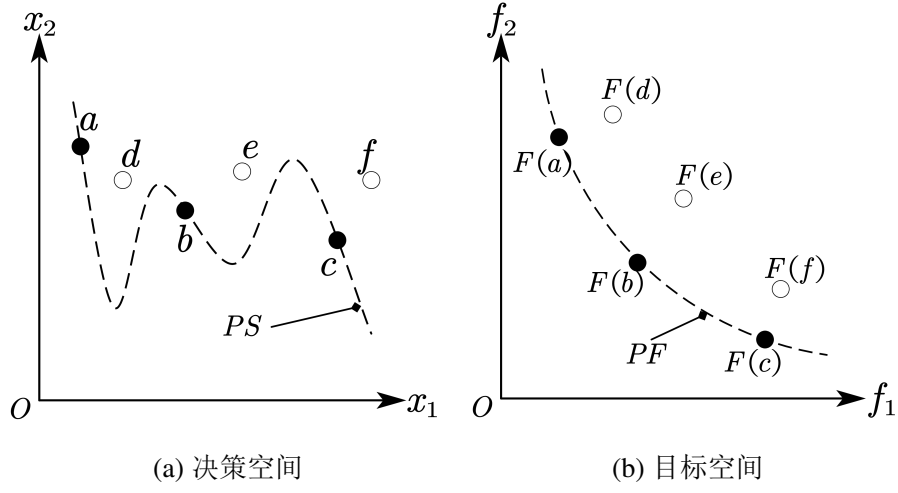


图 2.1 MOP 中相关概念示意图

构成的向量称为理想点（理想目标向量），记作 $\mathbf{z}^* = \{z_1^*, \dots, z_m^*\}^T$ ，即

$$z_i^* = \min_{\mathbf{x} \in \Omega} f_i(\mathbf{x}), \quad i \in \{1, \dots, m\}. \quad (2.3)$$

定义 2.11（边界点（Nadir Point））：目标空间中，PS 中所有可行解 \mathbf{x} 在各个目标上的最大值构成的向量称为边界点（边界目标向量），记作 $\mathbf{z}^{nad} = \{z_1^{nad}, \dots, z_m^{nad}\}^T$ ，即

$$z_i^{nad} = \max_{\mathbf{x} \in PS} f_i(\mathbf{x}), \quad i \in \{1, \dots, m\}. \quad (2.4)$$

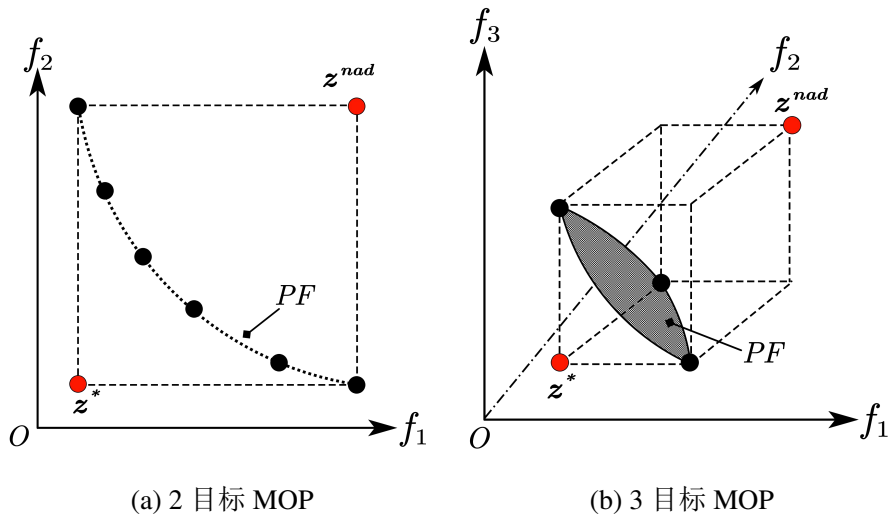


图 2.2 理想点和边界点示意图

如图 2.2 中所示, 图 2.2a 和图 2.2b 分别为两目标和三目标 MOP 理想点和边界点示意图。由图可以看出, 理想点和边界点限定了 PF 所在区域, 分别确定其上下界。

2.2 多目标组合优化算法

在大多数 MOP 中, 都需要同时考虑多个互相冲突的目标, 通常这些目标难以同时都达到最好的效果, 其中一个目标的性能的提升往往会导致问题中的另外一个或多个目标的性能降低。这就需要在优化各个目标时进行权衡, 这也致使算法最终求得的解不是惟一的, 而是一组互不支配的解 (非支配解集)。而这也是进化算法适合求解 MOP 的原因之一, 进化算法正是一种基于群体 (多个解) 的迭代算法。对于进化算法而言, 群体之间的信息遗传 (共享) 能够促进群体间个体的多元性, 并且能够保证群体中的一些优秀信息能被下一代群体继承, 进而使得整个群体进化 (优化)。由此, 本论文基础算法是基于进化算法的, 并且, 在本节中, 首先介绍多目标进化算法的基本框架, 然后会对一些多目标组合优化相关的优化技术进行详细介绍。

2.2.1 进化算法

进化算法 (EA) 是一种无需太多问题的先验假设, 使用种群循环迭代的方式来有效解决大规模问题的随机全局搜索算法。EA 的主要指导思想是基于达尔文的进化论, 通过适者生存、优胜劣汰的遗传机制来使得种群在表现出基因多样性的同时相互竞争, 从而使得种群进化。EA 正是通过模拟自然界物种的行为, 在解空间随机的搜索可行解, 产生后代种群的同时对种群进行适者生存、优胜劣汰的遗传操作, 将高质量的可行解保留, 低劣的解淘汰, 从而保证算法是往最优解的方向逼近直至收敛。由于 EA 本身的高适应性和高扩展性的优点, EA 可以与现有的启发式算法相组合, 被广泛应用于求解各种优化问题。

为了模拟自然界的物种进化, MOEA 之中引入了个体、种群、适应度、遗传操作、选择策略等概念:

- **个体**: 在实际问题抽象成解空间后, 将解编码成能表示其信息的向量 (符号串), 这个向量 (符号串) 就是个体。
- **种群**: 个体的集合, 进化的对象。
- **适应度**: 度量个体对于环境的适应程度 (个体的质量)。个体的适应度越高, 说明该个体的生存能力越强。
- **遗传操作**: 一般包括复制 (Reproduction)、交叉 (Crossover) 和变异 (Mutation) 三种基本形式, 是 EA 的核心部分。在算法中, 通过设置三种算子起作用的概率, 从而来控制整个群体的进化方向。
- **选择策略**: 是从父代中挑选出子代种群的策略, 常用的有非支配排序 (Non-dominated Sort, NDSort)^[2,5,25-27]、拥挤距离 (Crowding Distance)^[28]、精英保留等策略^[3] 或者混合策略。

随着问题的不同，上述 MOEA 中的部件也会有所不同，我们通常需要根据具体的问题，设计出适用于对应问题的算子。其算法框架如下：

Algorithm 1: 多目标进化算法框架

Input:

MOP: 多目标优化问题

params: 种群参数

stop: 终止条件

Output: \mathcal{PS}

```

1  $\mathcal{P} \xleftarrow{\text{初始化}} \text{MOP, params};$ 
2 while not stop do
3    $\mathcal{P}' \xleftarrow{\text{遗传操作}} \mathcal{P};$ 
4    $\mathcal{P}_{\text{new}} \leftarrow \mathcal{P} \cup \mathcal{P}';$ 
5    $\text{fitness} \xleftarrow{\text{适应度评价}} \mathcal{P}_{\text{new}};$ 
6    $\mathcal{P} \xleftarrow[\text{fitness}]{\text{选择策略}} \mathcal{P}_{\text{new}};$ 
7 end
8  $\mathcal{PS} \xleftarrow{\text{NDSort}} \mathcal{P};$ 
9 return  $\mathcal{PS};$ 

```

算法1是多目标进化算法的基本框架，与图 2.3 展示的多目标进化算法流程图相对应。在算法中，首先随机初始化一个种群 \mathcal{P} ， \mathcal{P} 中的个体即为问题的初始解，个体是根据 MOP 而设计的向量或符号串，种群规模根据 **params** 而设置。接下来就是迭代过程，模拟的是种群的进化。然后对种群 \mathcal{P} 进行遗传操作（对于不同的问题，遗传操作也不尽相同。组合优化通常采用局部搜索的方式产生新的个体，局部搜索算子也需要单独设计）产生新的种群 \mathcal{P}' ，接着将父代种群和新产生的种群混合，获取它们所有个体的评价值，然后根据评价值和选择策略选出子代种群（选择策略通常采用的是精英选择策略，是算法中至关重要的一步，它需要确保种群能够向着好的方向进化，即向 PF 逼近，同时还需要兼顾多样性和收敛性这两个关键指标）。最后在末代的时候，采用非支配排序的方式，从种群 \mathcal{P} 中选出非支配种群 \mathcal{PS} 。

由算法1和图 2.3 展示的多目标进化算法的工作流程，可以清晰的知道，如何产生和挑选子代种群是我们需要关注的对象。但是，没有在算法1和图 2.3 中体现的细节，也是我们研究的重点。比如：

- 设计容易表达问题的个体
- 算法终止的条件
- 局部搜索算法（产生新个体的方法，设计问题的邻域结构）

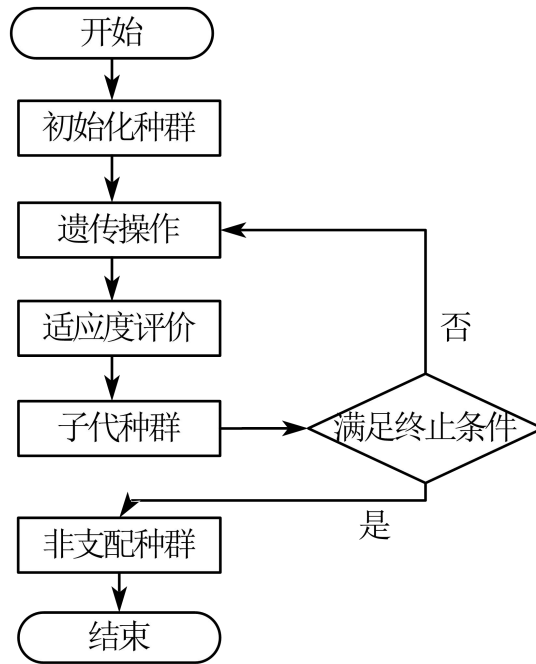


图 2.3 多目标进化算法流程图

2.2.2 基于分解的多目标进化算法

1985 年, Fourman 和 Schaffer 等人在解决多目标优化问题时提出多目标进化算法^[29,30], 经过多年的研究发展, 已经衍生出各种各样的多目标进化算法, 按照不同的策略大致可以归纳为三类算法框架: 基于支配的 (Domination-based)、基于分解的 (Decomposition-based) 和基于指标的 (Indicator-based) 多目标进化算法。尽管基于支配的和基于指标的多目标进化算法框架仍然被广泛的使用, 但是由于各自算法机制的原因, 导致他们在很多方面有着的局限性。基于支配的多目标进化算法框架的缺陷在于它不适用于超多目标 (优化目标数 $m > 3$) 的优化问题, 在超多目标优化问题的解集中, 几乎所有的解之间都是非支配的, 这使得非支配排序算法几乎失效, 导致算法中的选择策略不能为种群进化提供有效的指导, 最终导致算法不能向 PF 逼近, 不能起到优化的作用^[31,32]。同样, 基于指标的多目标进化算法框架的其中一个局限性是, 它依赖于指标的计算, 然而大多数指标的计算复杂度会随着问题目标的数量呈指数级增长, 这导致算法难以在有效时间内使用, 如基于超体积指标的多目标进化算法, 其计算复杂度随目标数呈指数上升, 即使目前有很多能够降低复杂度的近似方法, 也不能满足多目标优化问题的效率需求。

近年来, 在对多目标进化算法的研究中, 基于分解的多目标进化算法越来越受到研究者的关注, 其核心思想是通过分解的方式, 将一个多目标优化问题分解为一组单目标的子问题, 然后使用优化算法对这些单目标子问题同时进行优化, 最后将这些单目标子问题的解组合成一个

多目标优化问题的解集。在早期，分解的思想就在解决多目标优化问题的几个启发式算法中得到了一定程度上的应用^[33-37]。2007 年，Zhang 等人结合分解的思想，系统的提出了基于分解的多目标进化算法 (MOEA/D)^[11]，成为了最热门的基于分解的多目标进化算法之一。近来更是涌现出各种基于 MOEA/D 的算法框架，被用来解决各种各样的多目标优化问题^[38,39]。

在 MOEA/D 中，它把一个多目标优化问题根据一组权重向量分解成多个单目标子问题，然后根据各个子问题之间的联系形成邻居关系，然后利用邻居子问题之间的信息对所有子问题进行优化。对于一个具有 m 个目标的 MOP，我们将其分解成 k 个子问题，首先需要定义 k 个均匀的权重向量，其中一个权重向量 λ 为：

$$\lambda = (\lambda_1, \dots, \lambda_m)^T, \quad \forall \lambda_i \geq 0, \quad \sum_{i=1}^m \lambda_i = 1. \quad (2.5)$$

然后，针对不同类型的问题，我们可以是使用不同的分解方法，下面将介绍三种常用的分解方法（聚合函数）：

1. 加权和法 (Weight Sum, WS)：WS 是一种常用的线性多目标优化问题的聚合方法^[40]，它的目标函数聚合形式可定义为：

$$\begin{aligned} \text{minimize} \quad & g^{ws}(\mathbf{x} | \lambda) = \sum_{i=1}^m \lambda_i f_i(\mathbf{x}), \\ \text{subject to} \quad & \mathbf{x} \in \Omega. \end{aligned} \quad (2.6)$$

其中 \mathbf{x} 是决策向量， λ 为权重向量。

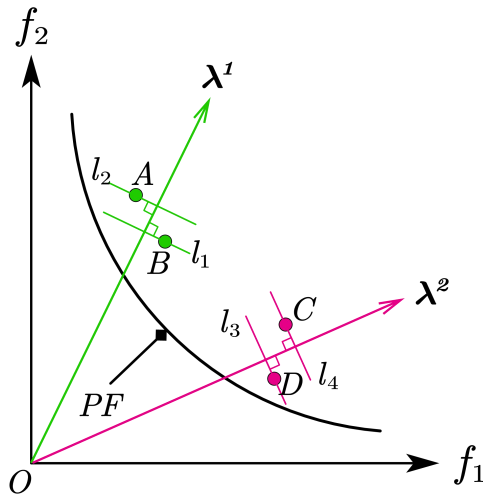


图 2.4 加权和法 (WS) 等高线示意图
 λ^1 和 λ^2 分别为两个子问题的权重向量， l_1, l_2, l_3, l_4 分别为对应子问题的等高线

图 2.4 展示的是一个二目标优化问题使用加权和法的等高线示意图。容易看出，加权和法的等高线是一簇与权重向量 λ 垂直的平行直线。个体在权重向量上的投影长度即为该个体在所属 λ 上的子问题的聚合值。在图 2.4 中，个体 A 和 B 是互不支配的，但是，在 λ^1 所属子问题上用加权和法后，个体 B 要优于个体 A，因为个体 B 在 λ^1 上的投影长度要比个体 A 短。同理，个体 D 优于个体 C。

2. 切比雪夫法 (Tchebycheff, TCH): TCH 是一种非线性多目标聚合方法^[35]，它的目标函数聚合形式可定义为：

$$\begin{aligned} \text{minimize} \quad & g^{tch}(\mathbf{x} | \lambda, \mathbf{z}^*) = \max_{1 \leq i \leq m} \lambda_i |f_i(\mathbf{x}) - z_i^*|, \\ \text{subject to} \quad & \mathbf{x} \in \Omega. \end{aligned} \quad (2.7)$$

其中 \mathbf{x} 是决策向量， λ 为权重向量， \mathbf{z}^* 为理想点（定义 2.10）。

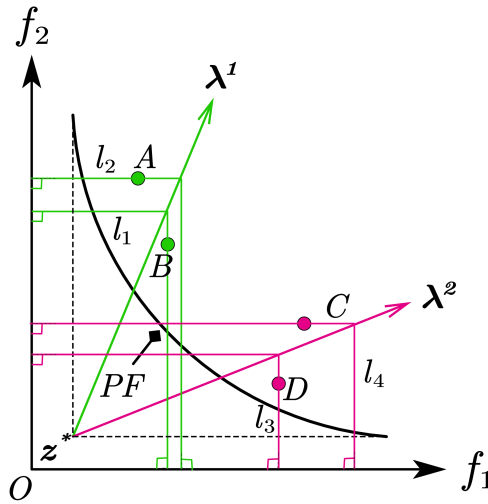


图 2.5 切比雪夫法 (TCH) 等高线示意图

λ^1 和 λ^2 分别为两个子问题的权重向量， l_1, l_2, l_3, l_4 分别为对应子问题的等高线

图 2.5 展示的是一个二目标优化问题使用切比雪夫法的等高线示意图。从图 2.5 中可以看出，对于 λ^1 所属子问题，个体 B 优于个体 A，因为个体 B 的切比雪夫聚合值要优于个体 A。同理，个体 D 优于个体 C。从图中还可以分析出，在连续 PF 时，用 TCH 分解方法得到的子问题的最优解是权重向量与 PF 面的交点。在非连续 PF 时，由于权重向量可能和 PF 面没有交点，所属不同权重向量的子问题可能具有相同的最优解。相比于加权和法，TCH 分解方法的等高线沿着权重向量呈直角锯齿形，因此，用 TCH 方法得到的子问题具有更小的收敛区域（搜索空间），这在处理高维问题时，TCH 分解方法能够很好的限制收敛区域，因此能够很好的保证种群的收敛性。

3. 基于惩罚的边界交叉法 (Penalty-based Boundary Intersection, PBI): PBI 是由 Zhang 等人于 2007 年提出的一种分解方法^[11], 它的目标函数聚合形式为:

$$\begin{aligned}
 & \text{minimize} \quad g^{pbi}(\mathbf{x} \mid \boldsymbol{\lambda}, \mathbf{z}^*) = d_1 + \theta d_2, \\
 & d_1 = \frac{\|(\mathbf{z}^* - F(\mathbf{x}))^T \boldsymbol{\lambda}\|}{\|\boldsymbol{\lambda}\|}, \\
 & d_2 = \|F(\mathbf{x}) - (\mathbf{z}^* - d_1 \boldsymbol{\lambda})\|, \\
 & \text{subject to} \quad \mathbf{x} \in \Omega.
 \end{aligned} \tag{2.8}$$

其中 \mathbf{x} 是决策向量, $\boldsymbol{\lambda}$ 为权重向量, \mathbf{z}^* 为理想点 (定义 2.10), d_1 表示目标向量 $F(\mathbf{x})$ 到权重向量 $\boldsymbol{\lambda}$ 的投影长度, d_2 表示目标向量 $F(\mathbf{x})$ 到权重向量 $\boldsymbol{\lambda}$ 的垂直距离, θ 为惩罚因子, 用于调节 d_1 和 d_2 之间的平衡, 控制种群的分布性和收敛性。

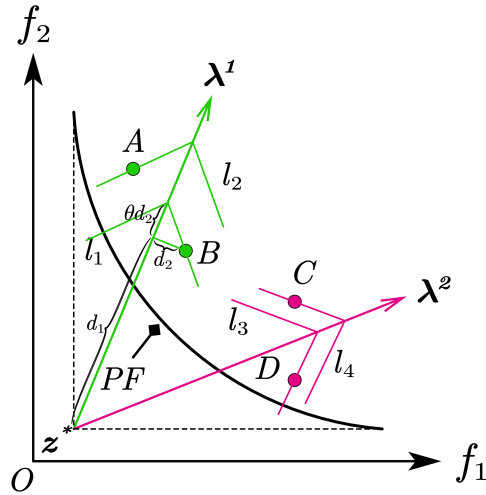


图 2.6 基于惩罚的边界交叉法 (PBI) 等高线示意图
 $\boldsymbol{\lambda}^1$ 和 $\boldsymbol{\lambda}^2$ 分别为两个子问题的权重向量, l_1, l_2, l_3, l_4 分别为对应子问题的等高线

图 2.6 展示的是一个二目标优化问题使用基于惩罚的边界交叉法的等高线示意图。在图中, d_1 是个体 B 到权重向量 $\boldsymbol{\lambda}^1$ 的投影长度, d_2 是个体 B 到权重向量 $\boldsymbol{\lambda}^1$ 的垂直距离, $d_1 + \theta d_2$ 即为 PBI 聚合值。显然, 对于 $\boldsymbol{\lambda}^1$ 所属子问题, 个体 B 的 PBI 聚合值要优于个体 A, 即个体 B 优于个体 A。同理, 个体 D 优于个体 C。通过式 2.8 和图 2.6 可以知道, PBI 能够通过调节 θ 参数来控制 d_1 和 d_2 之间的平衡, 从而控制种群的分布性和收敛性。由于能够控制分布性和收敛性之间的平衡这一特点, PBI 分解方法在处理高维目标 (超多目标) 问题时具有很大的优势。但是, 参数 θ 的设置也影响着算法的性能表现, 这同时也成为 PBI 分解方法的缺点之一。

MOEA/D 算法的核心思想是将 MOP 分解成一系列单目标子问题或者多个多目标子问题，然后利用这些子问题的邻域相似度，通过协作的方式来同时优化所有子问题，从而向 PF 逼近。通常子问题有权重向量通过聚合函数确定，这些子问题之间的邻域关系是通过各个子问题的权重向量之间的相似度（欧式距离）来度量的。与其他的多目标进化算法不同的是，MOEA/D 算法注重从当前子问题的邻域子问题中选择父个体，通过遗传算子（交叉、变异算子）来生成新个体，并且按照一定的策略来更新邻域中的种群。因此，一个优秀的基于邻域的优化策略能够确保 MOEA/D 算法的搜索效率。在算法的迭代进化过程中，一旦搜索到针对某个子问题的高质量解，这个解的优质信息就会迅速的被邻域内的其他个体继承，从而加快整个种群进化（收敛）的速度。下面是 MOEA/D 算法的一个基本框架流程：

Algorithm 2: MOEA/D 算法框架

Input:
MOP: 多目标优化问题
N: 子问题个数
T: 邻域大小
 λ : 一组均匀分布的权重向量 $\{\lambda^1, \dots, \lambda^N\}$
stop: 终止条件
Output: \mathcal{EP} : 外部集（精英种群）

```

1  $\mathcal{SP} = \{SP_1, \dots, SP_N\} \xleftarrow[N]{\lambda} \text{MOP};$ 
2  $\mathbf{B}(i) = \{i_1, \dots, i_T\}, i \in \{1, \dots, N\}$ , 其中  $\{\lambda^{i_1}, \dots, \lambda^{i_T}\}$  是最接近  $\lambda^i$  的  $T$  个权重向量;
3  $\mathcal{P} \xleftarrow{N}$  随机生成大小为  $N$  的初始种群;
4  $\mathbf{z}^* \leftarrow$  通过式 2.3 更新理想点;
5 while not stop do
6   for  $i = \{1, \dots, N\}$  do
7      $\mathbf{x}' \xleftarrow{\text{遗传算子}}_{\text{由 } SP_i \text{ 的邻居 } \mathbf{B}(i) \text{ 确定 } \mathcal{P} \text{ 中的两个个体};}$ 
8      $\mathbf{z}^* \leftarrow$  用  $F(\mathbf{x}')$  更新理想点;
9     通过聚合函数（式 2.6, 2.7, 2.8），用  $\mathbf{x}'$  更新  $SP_i$  的邻居解;
10     $\mathcal{EP} \leftarrow$  用  $F(\mathbf{x}')$  更新外部集;
11   end
12 end
13 return  $\mathcal{EP}$ ;
    
```

算法2 可以分为三个阶段：初始化、遗传生成和更新阶段。在初始化阶段，首先将 MOP 通过权重向量分解成 N 个子问题 $\mathcal{SP} = \{SP_1, \dots, SP_N\}$ ，并且初始化每个子问题的邻居，然后随机生成初始种群 \mathcal{P} ，并得到理想点 \mathbf{z}^* 。在遗传生成阶段主要是从邻居中选出父代个体，然后通过遗传操作来生成子代个体 \mathbf{x}' 。最后的更新阶段就是用生成的子代个体 \mathbf{x}' 来更新邻居解、理想点和外部集。应该注意的是，算法中邻域的大小和替换个体的次数会对种群的分布性和收敛性产生影响。并且在更新阶段，外部集 \mathcal{EP} 中保存着所有被搜索到的非支配解，为了节省资源，可以用一些已有的密度估计方法来控制外部集 \mathcal{EP} 的大小。

2.3 局部搜索

搜索算法是有目的的穷举一个问题解空间中部分或者所有的可能情况，直到搜索到问题的解的一种方法，实际上是根据初始条件和扩展规则构造一颗解答树并寻找符合目标状态节点的过程。从宏观上看，所有的搜索算法都可以划分为两个部分：控制结构（扩展点的方式）和生成系统（扩展节点），而大多数的算法优化和改进都主要是通过修改控制结构来完成的。

在全局搜索（Global Search, GS）中，需要将所有搜索到的路径都记录下来，直到搜索到最优解（满意解），其中，到达目标状态的搜索路径就是问题的解。但是，在大多数问题中，绝大多数的搜索路径与最终目标状态并不是紧密相关的。因此，没有必要将所有搜索到的路径都记录下来，只需要关键路径即可，这样能够节省大量的空间和时间。与 GS 记录所有搜索路径相对，只维护当前目标状态和记录部分关键搜索路径的搜索算法就是局部搜索算法（Local Search, LS）。当问题的状态空间（决策空间）很大时，全局搜索算法就不适用了，其找到最优解所需的时间和空间都将成指数形式的增长，因此，设计效率高的局部搜索算法也是我们研究的重点。并且，本文主要的研究载体是多目标组合优化问题，这类问题的决策空间是离散的，并且十分巨大，导致往常用的遗传算子（交叉、变异等）生成新个体的效率低下。因此，应对这类问题，局部搜索算子就成了算法中必不可少的一部分。

2.3.1 基本概念

对于一个组合优化问题，局部搜索算法就是在该问题的候选解空间（邻域结构）上进行搜索，其搜索过程是从选择初始的候选解开始，然后在当前候选解的邻域结构中进行搜索，如此迭代直到找到一个满意的解或者搜索到了局部搜索算法的终止条件。在搜索过程中，每次对候选解是否被选择的决策仅基于有限的局部信息，并且算法中的决策和最开始的初始化可以是随机的。因此，基于以上局部搜索算法过程，可以给定以下定义^[41]：

定义 2.12: 给定一个最小化 CO 问题 $P = (S, f)$ ，变量 $\mathbf{x} = \{x_1, \dots, x_n\} \in X$ ， X 是离散有限（可数无限）变量集，其决策域为 $\mathcal{D} = \{D_1, \dots, D_n\}$ ，约束域为 C ，则有

$$\begin{aligned} \text{minimize} \quad & f : D_1 \times \dots \times D_n \rightarrow \mathbb{R}^+, \\ \text{s.t.} \quad & C. \end{aligned} \tag{2.9}$$

则所有可能的可行解可表示为：

$$\begin{aligned} S = \{s = \{(x_1, v_1), \dots, (x_n, v_n)\}\}, \\ \text{s.t.} \quad v_i \in D_i, x_i \in \mathbf{x} \in X, s \xrightarrow{s.t.} C. \end{aligned} \tag{2.10}$$

其中, \mathcal{S} 被称为搜索 (解) 空间, 因为集合中的每个元素都可以看作是一个候选解。要解决 CO 问题, 需要在搜索空间 \mathcal{S} 中找到能够使得目标函数 f 值最小的解 $s^* \in \mathcal{S}$, 即 $f(s^*) \leq f(s), \forall s \in \mathcal{S}$ 。 s^* 称为 (\mathcal{S}, f) 的全局最优解。

定义 2.13: 在局部搜索算法中, 邻域动作是一个函数:

$$\mathcal{N} : \mathcal{S} \rightarrow 2^{\mathcal{S}}. \quad (2.11)$$

对于解空间中任意一个解 $\forall s \in \mathcal{S}$, 有

$$\mathcal{N}(s) \subseteq \mathcal{S}. \quad (2.12)$$

其中, $\mathcal{N}(s)$ 就是 s 的邻域。

定义 2.14: 对于一个邻域动作 $\mathcal{N} : \mathcal{S} \rightarrow 2^{\mathcal{S}}$, 在搜索空间 \mathcal{S} 中, 有解 \hat{s} , 使得 $\forall s \in \mathcal{N}(\hat{s}) : f(\hat{s}) \leq f(s)$, 则称 \hat{s} 为局部最优解。如果 $f(\hat{s}) < f(s), \forall s \in \mathcal{N}(\hat{s})$, 则称 \hat{s} 为严格局部最优解。

Algorithm 3: 局部搜索算法框架

Input:
 $P = (\mathcal{S}, f)$
 $\mathcal{N} : \mathcal{S} \rightarrow 2^{\mathcal{S}}$
 \mathcal{A} : 接受策略
stop: 终止条件
Output: s^*

```

1  $s^* = (s \leftarrow \text{初始化});$ 
2 while not stop do
3   foreach  $s' \in \mathcal{N}(s)$  do
4      $s^* \xleftarrow{s' < s^*} s';$ 
5     if  $\mathcal{A}(s', s)$  then
6        $s \leftarrow s';$ 
7     end
8   end
9 end
10 return  $s^*;$ 
    
```

通过定义 2.12, 定义 2.13, 定义 2.14 和算法3 可以知道, 通过邻域动作和邻域, 可以产生对应解的邻居解集合, 在迭代优化时, 可以通过一定的策略来判断是否接受邻居解集合中的某个解, 从而更新当前解。假设每次都接受比当前解更优的邻居解, 那么当前解每次都能得到改进, 这能够使得算法快速收敛到局部最优。

2.3.2 邻域结构

由章节2.3.1 可以知道邻域决定一个局部搜索算法的搜索效率和最终解的质量。邻域越小，搜索空间也会变得很小，局部搜索算法只需要搜索部分解空间就会很快收敛，这很大程度上节省了计算时间。但是，这并不能保证最终的局部最优解也足够的好，因为当邻域足够小时，解空间很大可能会不包含全局最优解或者优质的局部最优解，这使得算法不能搜索到优质的解。若邻域足够大，解空间包含了全局最优解或者优质局部最优解，但是算法需要花大量的计算量和时间去搜索到这个优质解。因此，综合以上原因，我们的目的是要根据具体的问题，找到一个邻域，使得这个邻域不仅足够的小，而且尽可能多的包含优质的局部最优解，我们称这样的邻域为邻域结构。

一般地，邻域结构是建立在具体问题的基础之上的。并且，本文研究的主要内容也与邻域结构相关。因此，为保证一致性，在本文中，具体的问题和测试用例统一使用旅行商问题（Traveling Salesman Problem, TSP）和多目标旅行商问题（Multi-Objective Traveling Salesman Problem, MOTSP）（参考章节2.6.2）。对于 TSP 问题，它的邻域是一个（全连接图）邻接矩阵，其中每个顶点的邻域是所有与该顶点相连的顶点。这个邻域中还包含了其他顶点的邻域，这些邻域中的顶点可以是与该顶点相邻的顶点，也可以是其他顶点的邻域。下面给出邻域结构的定义：

定义 2.15（邻域结构（Neighborhood Structure, NS））：邻居结构是每个顶点只与少数几个顶点相连（候选集（Candidate Set, CSet））的稀疏图，记作 \mathcal{N} 。

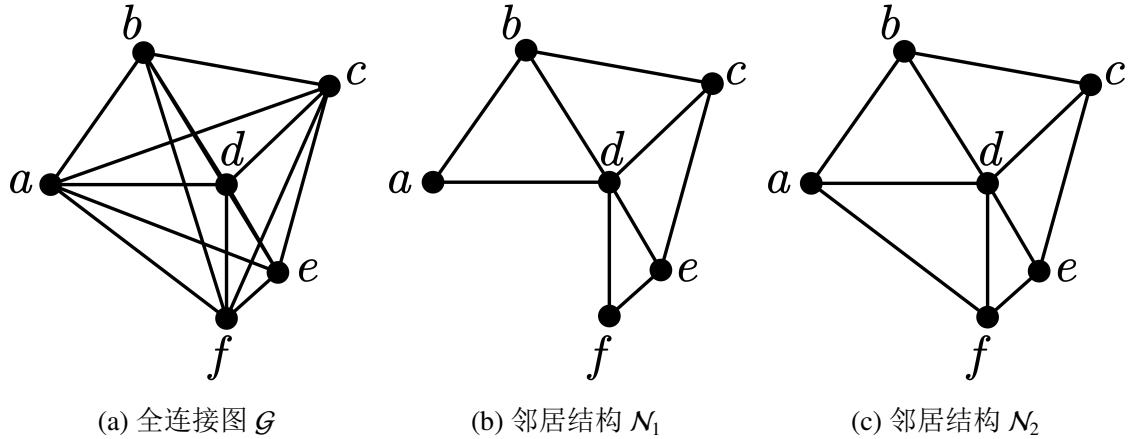


图 2.7 邻域结构示意图

如图 2.7 所示，给定一个 TSP 问题 \mathcal{P} 和问题 \mathcal{P} 的全连接图 \mathcal{G} ，由定义 2.15 可知，当每个顶点的候选集由剩余所有顶点组成时，那么问题 \mathcal{P} 的邻域结构就是 \mathcal{G} 。在图 2.7 中，图 2.7a 就是一个全连接图。由算法 3 可知，当邻域结构是全连接图时，这使得算法需要搜索整个搜索空间，显然不符合题意。因此，我们需要对邻域结构进行定制。如图 2.7b 所示，虽然图 2.7b 的邻域结

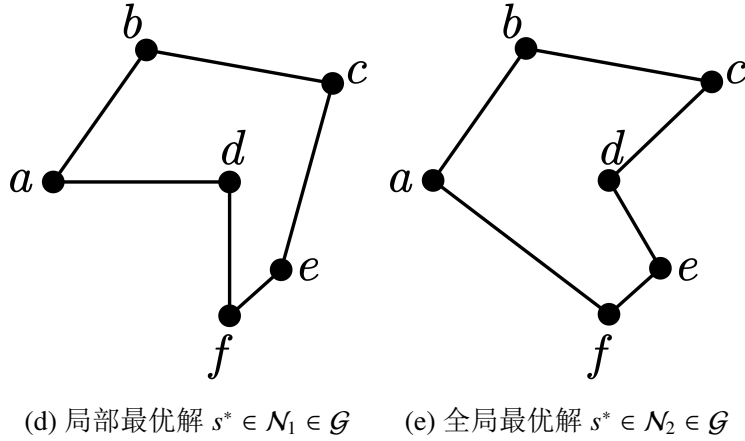


图 2.7 邻域结构示意图（续）

构足够的小，但是，它并没有将图 2.7e 所示的最优路径中的所有边（如边 af ）都包含进去，当算法使用这个邻域结构，不用其他策略时，无论如何也不能搜索到最优解（只能搜索到局部最优解，如图 2.7d）。所以，我们设计的邻域结构（如图 2.7c）不仅要小，还需要保证足够的质量，以确保算法能够搜索到满意的解。

2.3.3 邻域动作

由定义 2.13 可知，邻域动作是一个函数，通过这个函数，算法能够在当前解的邻域中，生成对应的邻居解集合。由章节 2.2.1 可知，遗传操作（交叉、变异算子）是进化算法中最重要的组件之一，它的作用就是产生新个体。又因为多目标组合优化问题的决策空间是离散且巨大的，这导致交叉、变异等算子生成新个体的效率和质量都达不到要求。因此，将邻域动作作为进化算法的一部分，替代遗传操作在算法中的功能，以局部搜索的思想来产生优质的子代个体。

在组合优化问题中，邻域动作的规则不仅与问题的解的编码形式相关，更是在邻域结构的基础上进行搜索产生新解的。因此，为保证与邻域的一致性，统一使用 TSP 和 MOTSP。对于 TSP 问题，其解的编码形式为一个所有顶点组成的序列路径。针对序列路径，产生新个体的方式有很多。比如，有一序列路径 $s = \langle v_1, v_2, v_3, v_4, v_5 \rangle$ ，任意交换两个顶点（ v_2, v_3 ）的位置^[42]，这样就在 s 的基础上，产生了一个 s 的邻居解 $s' = \langle v_1, v_3, v_2, v_4, v_5 \rangle$ 。其中“交换两个顶点”这个动作就是邻域动作。

图 2.8 是一种名为 k-opt 的邻域动作规则^[43]，其中 k 代表要交换的边数， k 越大，代表新个体对原解的改动越大，并且，搜索所有解的判断时间也是 k 的指数倍。图 2.8a 中是 2-opt 移动示意图，原序列路径为 $s = \langle v_1, v_2, v_4, v_3 \rangle$ ，在经由 2-opt 后， s 删除 x_1, x_2 两条边，连接 y_1, y_2 两条边得到 $s' = \{v_1, v_4, v_2, v_3\}$ ，相当于将 s 中的 x_1, x_2 两条边替换成 y_1, y_2 两条边，由此生成 s' 。同理，图 2.8b 为 3-opt，图 2.8c 为 4-opt。从图中很容易知道，一条序列路径 s 经由 k-opt 操作

后,就是将 s 中的 k 条边,使用特定的交换策略,替换成不属于原序列路径的不同的 k 条边,这样就可以产生了 s' 。并且,从图中可以看出, k -opt 是在 $(k-1)$ -opt 的基础上再进行一次换边操作的,在使用 k -opt 时,我们期望 k 越大,获得的结果越好,并且对于足够大的 k ,我们期望序列路径 s 经由 k -opt 操作得到的 s' 应该是最优的。但是,随着 k 的增大,测试所有可能交换的边所需的操作次数增长得非常快。并且,我们并不知道给定问题到底需要设置多大的 k 才能获得可接受的优化结果,这也是使用 k -opt 算子无法避免的一个缺点。

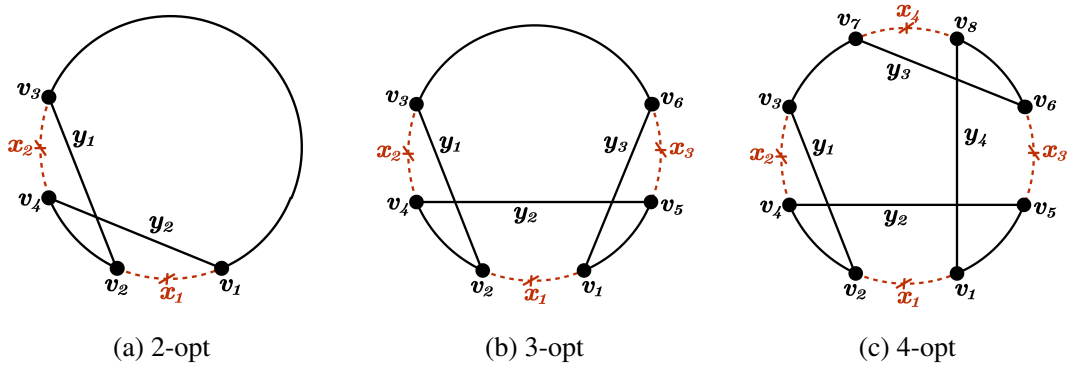


图 2.8 k -opt 示意图

2.4 进化迁移优化

迁移学习使用从源域 (Source Domains, SD) 中获取的知识来提升不同 (但相关) 目标域 (Target Domains, TD) 的学习能力^[1]。相似地,进化迁移优化 (Evolutionary Transfer Optimization, ETO) 是一种将 EA 求解器与知识学习和跨领域信息转移相结合,以实现更好的优化效率和性能的范式。Kay 在文中提到^[24], ETO 虽然是一个兴起不久的优化范式,但是它已经在许多优化领域 (比如:多任务优化、多目标优化和机器学习等) 发挥了重要的作用。

针对多目标优化问题来说, ETO 旨在通过学习或迁移解、结构化信息等与问题相关的有用特征,在提高解的质量或搜索速度等方面来提高算法的性能。从算法设计的角度来说,现有的 ETO 方法可以分为同构 (Homogenous) ETO 和异构 (Heterogeneous) ETO:

- **同构 ETO:** 共享相同搜索空间的跨问题的知识迁移。
- **异构 ETO:** 跨具有不同搜索空间的问题之间的知识迁移。

此外,同构 ETO 已通过以不同子问题之间交换 PS 中解的形式实现跨子问题的知识迁移的方式应用于多目标和超多目标优化问题中^[24]。如图 2.9 所示,多目标和超多目标优化中的有用知识可以采用非支配解集、代理模型等形式来实现跨问题的知识迁移。如果 ETO 取得了在有知识迁移的情况下比没有知识迁移更好 (更差) 的表现,则称为正 (负) 迁移。

直观地说,随着多目标优化问题中目标数量的增加,要迁移的非支配解的数量将急剧增加。

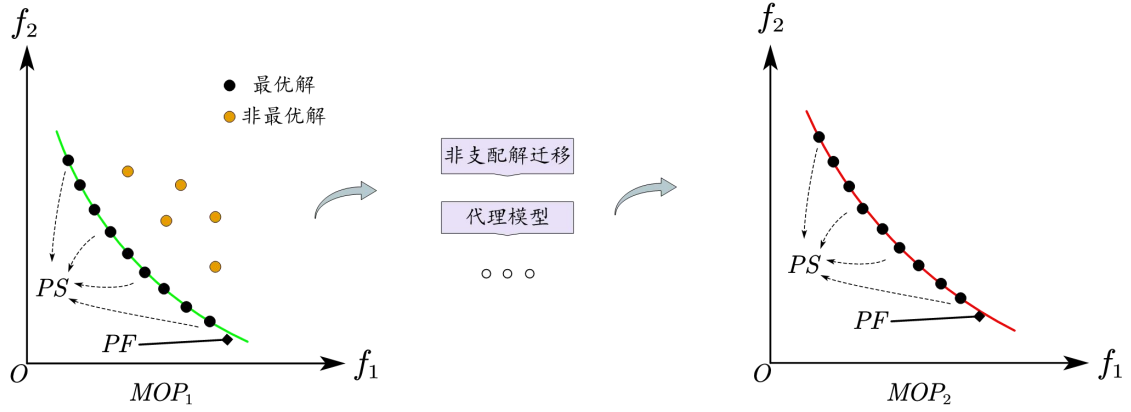


图 2.9 跨问题知识迁移示意图

因此，与只迁移 MOP 问题中非支配解的 ETO 相比，正确地构建源域和目标域中可迁移知识之间的映射关系是我们应该研究的重点。并且我们需要深入分析，在 MOP 中，什么信息是有用的，以及这些信息如何与跨问题域的多个目标相关连的。这可以更好地了解 ETO 在多目标和多目标优化中怎样并且在什么时候能够实现正迁移。

2.5 性能评价指标

对于一个算法，抛开其本身的复杂性而言，其表现好坏则是通过这个算法所求得相应问题的最终结果的优劣而定。对于 CMOPs 来说，其最终结果以非支配解集（NDS_{Set}，定义 2.6）的形式呈现。所以，一个 MOEA 的性能通常从两个方面进行评估：

1. 群体解集的收敛性：NDS_{Set} 与真实 Pareto 前沿（PF）的贴合程度。
2. 群体解集的多样性：NDS_{Set} 在空间上分布的均匀程度和广泛程度。

Knowles 等人于 2006 年在文中^[44]介绍了很多评估 NDS_{Set} 性能的评价指标。例如：超体积度量指标（Hypervolume, HV）^[8]、反向世代距离指标（Inverted Generational Distance, IGD）^[45]和集合覆盖度量指标（Set Coverage, C-Metric）^[8]等。在本文中，除了需要对 MOEA 的性能进行评估外，还需要对邻域结构（最优边遗失率指标^[46]）和 TSP 问题的解进行质量评估（距离趋近度指标^[46]），下面将分节详细介绍。

2.5.1 超体积度量指标

超体积度量指标 HV 是一种通过计算参考点与 NDS_{Set} 围成空间的超体积实现对算法综合性能评价的方法，其数学形式如下：

$$HV = \mathcal{L}\left(\bigcup_{i=1}^{|S|} v_i\right). \quad (2.13)$$

其中, \mathcal{L} 为 Lebesgue 测度, \mathcal{S} 代表 NDSet, $|\cdot|$ 代表集合中元素的个数, v_i 为非支配个体 $s_i \in \mathcal{S}$ 和参考点 z^* 构成的超体积。

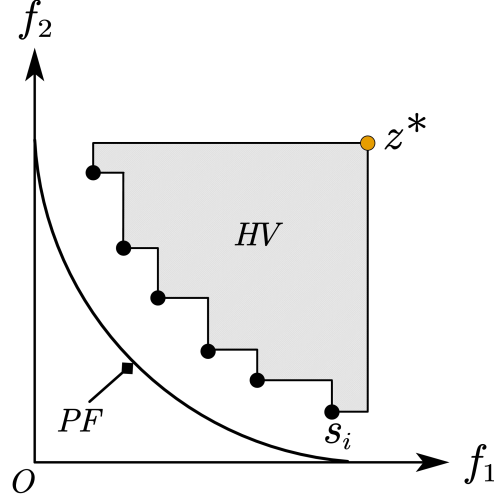


图 2.10 HV 指标示意图

如图 2.10 所示是一个二目标问题的非支配解集的超体积度量示意图, 其中超体积度量值为 NDSet 和参考点 z^* 构成的阴影区域的面积。 z^* 的选择方式有多种, 常用的有: NDSet 每维上的最大值组成的向量和松散形式的最差点^[47]。从图中容易知道, HV 指标值同样遵循 Pareto 支配 (定义 2.2) 原则, 以最小化 MOP 为例, 如果个体 $s_1 < s_2$, 则 s_1 的 HV 度量值一定大于 s_2 , 同理, 可推广到两个解集之间的支配关系。并且, 计算 HV 指标值时, 真实 PF 不是必需的, 所以 HV 指标具有更好的实用性。但是, 计算 HV 指标的时间复杂度非常大, 而且参考点的选择也会影响 HV 指标的正确性^[48]。

2.5.2 反向世代距离指标

反向世代距离指标 IGD 也是一种兼顾收敛性和多样性的综合评价指标。世代距离是指 NDSet 中所有个体到 Pareto 最优解集 (PS) 的平均距离, 而反向世代距离则是 PS 中的个体到所求 NDSet 的平均距离。由此, 其计算公式可表述如下:

$$IGD = \frac{1}{|PS|} \sum_{\mathbf{x}^* \in PS} \min_{s \in \mathcal{S}} d(\mathbf{x}^*, s). \quad (2.14)$$

其中, PS 为 Pareto 最优解集 (真实 PF 的真子集), $|\cdot|$ 代表集合中元素的个数, \mathcal{S} 为非支配解集, $d(\mathbf{x}^*, s)$ 代表个体 \mathbf{x}^* 与个体 s 的欧几里得距离。IGD 值越小, 说明算法的综合性能越好。

此外, 由图 2.11 可以直观地理解 IGD 指标对解集的综合性能评价。从图中可以看到, NDSet 距离 PF 的距离远近能够直观地表现收敛性的好坏。同时, 可以观察到, 当 NDSet 的分布性不

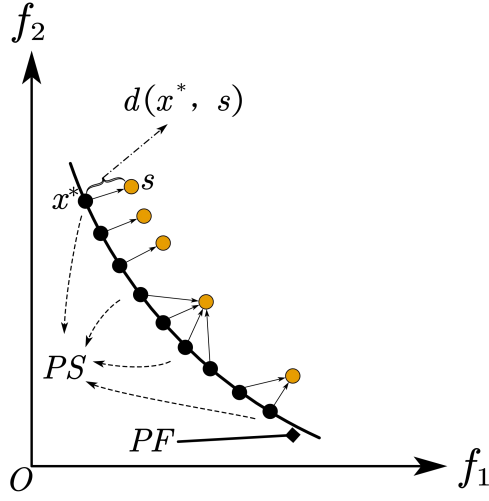


图 2.11 IGD 指标示意图

好时，由式 2.14 计算可得 IGD 评价值会变大，并且在图中也可以看出，当 NDSet 分布均匀且广泛时，IGD 评价值会很小。因此可以说明 IGD 评价指标不仅能够反映解集的收敛性，而且能够刻画解集的多样性（广泛且均匀）。但是，IGD 指标需要通过真实 PF 来计算指标值，而现实问题大多都不知道其真实最优解，所以，这也成为其缺陷之一。

2.5.3 集合覆盖度量指标

集合覆盖度量指标 C-Metric 是一种两个解集相对覆盖率比较的方法，其利用解与解互相间的支配关系来判断两个解集之间的收敛性能的优劣。在 C-Metric 中，设在目标空间中有两个非支配解集 $S_a, S_b \subseteq S$ ，C-Metric 则将 (S_a, S_b) 映射到 $[0, 1]$ 之间，得到 S_a 与 S_b 两解集之间的覆盖率，其数学形式如下：

$$C(S_a, S_b) = \frac{|\{b \in B \mid \exists a \in A, a < b\}|}{|S_b|} \times 100\%. \quad (2.15)$$

其中 $|\cdot|$ 代表集合中元素的个数， $C(S_a, S_b)$ 代表 S_b 中所有能被 S_a 中的解支配的解的数目占 S_b 中解数目的百分比。

由上述定义可知，如果 S_a 中的解都弱支配 S_b 中的解，那么 C-Metric 的值等于 1，反之为 0。一般的，由于 S_a 和 S_b 的交集并不为空，所以在评估解集的质量时，需要同时考虑 $C(S_a, S_b)$ 和 $C(S_b, S_a)$ 。要注意 C-Metric 方法并不是解集之间的距离上的比较，而是不同解集在支配关系上的相对比较。C-Metric 的优点在于计算简单，但是计算复杂度取决于非支配排序算法，当目标数很大时，非支配排序算法的复杂度会很高，会耗费大量时间计算该指标。

2.5.4 距离趋近度指标

距离趋近度指标是一种用来评判单目标解质量的方法。对于最小化优化问题而言，距离趋近度指标利用解的适应值超过已知最优解的适应值的百分比来评判解的质量。假设有解 s 和最优解 s^* ，适应度函数 φ ，则其数学形式如下：

$$Gap = \frac{|\varphi(s) - \varphi(s^*)|}{\varphi(s^*)} \times 100\%. \quad (2.16)$$

其中 $|\cdot|$ 代表集合中元素的个数，一般地，我们很难知道具体问题的最优解，所以 s^* 一般可以用目前已知最优解来代替。

由式 2.16 可知，当适应度函数为线性函数时，其适应值就可以表现为解的值，当算法求得的解距离最优解越近，其 Gap 值就越小，代表该解质量越好。因此，可以利用距离趋近度指标来评估不同解的质量。

2.5.5 最优边遗失率指标

最优边遗失率指标是一种评估邻域结构质量的方法。对于图优化问题（如 TSP、MST）而言，它们的解都是一个序列路径（节点之间可以构成边）构成。所以，最优边遗失率指标计算的是，邻域结构中，缺失的最优边（最优解中的边）的数目占最优解中边的数目的百分比。其数学描述如下：

$$Missing = (1 - \frac{|\{e \mid e \in \mathcal{E}(s^*) \cap \mathcal{E}(\mathcal{N})\}|}{|\mathcal{E}(s^*)|}) \times 100\%. \quad (2.17)$$

其中 $|\cdot|$ 代表集合中元素的个数， s^* 为最优解（一般用目前已知最优解替代）， \mathcal{N} 为邻域结构， $\mathcal{E}(\cdot)$ 获取解或者邻域结构中的边。

由式 2.17 可知，最优边遗失率指标仅能描述邻域结构的质量，并不能表示邻域结构的冗余度。当邻域结构是完全图（包含整个问题）时，那么它一定包含最优解中的所有边，Missing 值为 0，但这并不代表这个邻域结构是我们所期望的。所以，在用最优边遗失率指标对邻域结构进行评估时，应该建立在多个被评估的邻域结构拥有同样的稀疏度（出入度）的基础上。

2.6 测试问题

在本节中，将会介绍适用于本文的测试问题。由于本文提出了一种针对旅行商问题（Traveling Salesman Problem, TSP）的邻域结构生成的方法，为后续的基于邻域结构迁移的多目标进化算法提供一个可用于迁移的知识模型，所以，在本文将会用到单目标的 TSP 问题作为邻域结构生成的测试用例。又因为，邻域结构具有专属性（一种结构模型可能只适用于专属的问题），

所以本文将使用多目标旅行商问题（Multi-Objective Traveling Salesman Problem, MOTSP）作为基于邻域结构迁移的多目标进化算法的测试用例。同时，在最后还会介绍多目标最小生成树问题（Multi-criteria Minimum Spanning Tree, MCMST），该问题具有和 MOTSP 类似的邻域结构，可以使用 MOTSP 的邻域结构尝试解决 MCMST。本文提出的是解决具有图邻域结构组合优化问题的算法框架，有兴趣的读者可以尝试用该框架解决其他类型的多目标组合优化问题，介绍 MCMST 仅为读者提供思路启发。

2.6.1 TSP

旅行商问题（TSP）是一种 NP 难的组合优化问题^[49-51]，最早于 1959 年 Dantzig 等人提出，并且在最优化领域中得到了深入的研究，同时许多优化方法都是用 TSP 作为一个测试基准。经典的 TSP 可以描述为：一个商品推销员想要去若干个城市推销商品，该推销员从其中一个城市出发，不重复的经过所有城市，最后回到出发地。推销员应该制定怎样的旅行路线，使得总行程最短。

从图论的角度而言，TSP 实质是在一个带权完全无向图中，找一个权值最小的哈密顿回路。该问题可抽象如下：给定一个完全图 $\mathcal{G} = (V, E)$ ，其中 $V = \{v_1, \dots, v_n\}$ 为包含 n 个元素的点集， $E = \{e_{i,j} \mid v_i, v_j \in V, i \neq j\}$ 为包含 $n^2 - n$ 个元素的边集，寻找一条路径序列 $\mathcal{T} = \langle t_1, \dots, t_n \rangle$ ， $t_i \in V$ （所有点都被访问且仅访问一次）使得权重总和 $f(\mathcal{T})$ 最小：

$$\text{minimize } f(\mathcal{T}) = d(t_n, t_1) + \sum_{i=1}^{n-1} d(t_i, t_{i+1}). \quad (2.18)$$

其中， $d(v_i, v_j)$ 为边 $e_{i,j}$ 的距离（权重）。

2.6.2 MOTSP

多目标旅行商问题（MOTSP）相当于是 TSP 的一个扩展。TSP 中，边权仅考虑了距离，所以在计算的时候只要求总距离最短的路径序列。对于 MOTSP 而言，边权不仅只考虑距离，还会考虑旅行时间、费用等多种类型的权重，每多一种权重就相当于问题多了一个目标。所以，在求解 MOTSP 时，我们需要对多个权重（目标）进行权衡，从而给出一组非支配解作为 MOTSP 的解决方案。与 TSP 类似，MOTSP 同样是选择一条路径序列 $\mathcal{T} = \langle t_1, \dots, t_n \rangle$ ， $t_i \in V$ ，使得各目标权重总和最小，其数学描述如下：

$$\text{minimize } f_k(\mathcal{T}) = d_k(t_n, t_1) + \sum_{i=1}^{n-1} d_k(t_i, t_{i+1}), \quad (2.19)$$

$$k \in \{1, \dots, m\}.$$

其中, $d_k(v_i, v_j)$ 为边 $e_{i,j}$ 第 k 个目标上的权重。

2.6.3 MCMST

最小生成树 (Minimum Spanning Tree, MST) 问题是一个经典的组合优化问题, 其目标是构造一个带全图的最小权值总和的生成树。设有一个带权无向图 $\mathcal{G} = (V, E, W)$, 其中 $V = \{v_1, \dots, v_n\}$ 为点集, $E = \{e_{i,j} \mid v_i, v_j \in V, i \neq j\}$ 为边集, $W = \{w_{i,j} \mid e_{i,j} \in E, i \neq j\}$ 为边权集, 寻找一个包含所有点的树 $\mathcal{T} = (V, E_{\mathcal{T}}, W_{\mathcal{T}})$, $E_{\mathcal{T}} \subset E, W_{\mathcal{T}} \subset W$, 使得 $W_{\mathcal{T}}$ 最小, 则称 \mathcal{T} 为图 \mathcal{G} 的最小生成树。

与 MOTSP 类似, 当考虑多种类型的权重时, 该问题就是一个多目标最小生成树 (Multi-criteria Minimum Spanning Tree, MCMST) 问题。所以, MCMST 的解同样为一组非支配解, 其中每个解都是一个最小生成树的边集 $E_{\mathcal{T}}$, 则其数学描述如下:

$$\begin{aligned} \text{minimize } f_k(\mathcal{T}) &= \sum_{e_{i,j} \in E_{\mathcal{T}}} w_{i,j}^k, \\ k &\in \{1, \dots, m\}. \end{aligned} \quad (2.20)$$

其中, $w_{i,j}^k$ 为边 $e_{i,j}$ 第 k 个目标上的权重。容易知道解 \mathcal{T} 中包含 n 个顶点, $n-1$ 条边。

2.7 本章小结

本章的主要目的是为后续章节提供一定的基础理论支撑。本章在一开始就介绍了多目标优化问题的定义及相关概念, 并由此对多目标进化的基础算法进行了简要的介绍, 同时详细介绍了基于分解的多目标进化算法。然后针对进化算法在组合优化中遗传算子效率的问题, 引出了在基于分解的多目标进化算法中使用局部搜索来对多目标组合优化问题进行优化的思想。接着介绍了与其他领域相结合的进化迁移优化相关概念。最后, 介绍了一些性能评价指标和经典的测试问题。

第三章 基于最小生成树和欧拉回路的邻域结构生成方法

3.1 引言

Fourman 等人于 1985 年在解决多目标优化问题时提出多目标进化算法 (MOEA)^[29,30], 其实质就是进化算法 (EA) 在多目标优化领域的一种扩展。由章节 2.2 介绍可知, EA 是一种受达尔文进化论启发而产生的、基于种群且不断迭代进化的一大类元启发式算法, 其中进化规则中的遗传操作正是 EA 的核心部分。EA 通过模拟自然界物种适者生存、优胜劣汰的遗传机制, 从而使得种群在表现出基因多样性的同时相互竞争, 以致整个种群得以进化。而遗传机制其中一个特征就是, 通过复制、交叉、变异操作从父代种群个体的基因特征中产生新的子代个体。然而对于大多数多目标组合优化问题而言, 其决策空间 (基因编码形式) 是离散并十分巨大的, 导致了这些遗传操作 (复制、交叉、变异算子等) 生成子代个体的基因特征变化不大, 生成新个体的质量和效率都会大大降低, 从而使得种群的进化较为缓慢。因此, 我们需要采用一些策略或算法来替代遗传操作的部分甚至全部功能, 嵌入到多目标进化算法当中, 从而改善生成新个体质量和效率低的问题。

通过章节 2.3 的介绍可知, 局部搜索算子是能够改善甚至替代多目标进化算法中遗传操作来生成新个体的方法之一。对于一个组合优化问题而言, 局部搜索算子就是在该问题的候选解空间上进行搜索的一种方法, 它依赖邻域结构来生成高质量的邻居解 (新个体), 这正是遗传操作的特征之一。而邻域结构正是局部搜索算法中最核心的部分之一。一个质量好的邻域结构, 不仅能够帮助局部搜索算子搜索到高质量的解, 而且能够加速整个算法的运行速度, 使得 MOEA 收敛得更好且更快。比如, Lust 等人^[52] 采用非支配排序的方式为二目标 TSP 问题构建了一个稀疏图 (邻域结构), 从而加速 2PPLS^[53] 算法求解该问题的运行速度。并且, LKH (Lin Kernighan Helsgaun)^[54] 是目前求解 TSP 问题最有效的方法之一, 其在算法中也用到了名为 α -nearest^[55,56] 的方法生成的邻域结构, 使用该邻域结构的 LKH 不仅提高了算法整体的运行效率, 同时, 算法中更是采用了一些特殊的规则与邻域结构相配合, 使得算法能够快速得搜索到高质量的候选解。这些足以说明邻域结构在算法中能够发挥的作用。

邻域结构除了能够作为局部搜索算子的一部分, 还能够表征对应问题的内在信息被用来在不同域 (SD, TD) 中迁移的知识。由章节 2.4 可知, 在 MOPs 中, 目前进化迁移优化 (ETO) 都是将目标空间的解 (非支配解, 非最优解) 当作被迁移的知识。在本文中, 我们将邻域结构当作被迁移的知识, 提出一种基于邻域结构迁移的多目标组合优化算法。

然而, 邻域结构的构建与我们所要解决的问题紧密相关。邻域结构展现的形式需要我们对具体的组合优化问题有深刻的理解, 并且能将该问题的内在信息用某种结构表示出来。这也正

是我们所言研究的重点。我们将通过本章节对 TSP 问题的邻域结构的生成方法进行详细地介绍, 并期望研究者能够从该工作中得到启发, 设计出更具泛化性(适用于多种组合优化问题)的邻域结构。

3.2 研究动机

3.3 算法框架

3.4 实验结果与分析

3.5 本章小结

参考文献

- [1] Pan S J, Yang Q. A survey on transfer learning[J]. IEEE Transactions on knowledge and data engineering, 2009, 22(10):1345–1359.
- [2] Srinivas N, Deb K. Multiobjective optimization using nondominated sorting in genetic algorithms[J]. Evolutionary computation, 1994, 2(3):221–248.
- [3] Deb K, Pratap A, Agarwal S, et al. A fast and elitist multiobjective genetic algorithm: NSGA-II[J]. IEEE transactions on evolutionary computation, 2002, 6(2):182–197.
- [4] Zitzler E, Laumanns M, Thiele L. SPEA2: Improving the strength Pareto evolutionary algorithm[J]. TIK-report, 2001, 103.
- [5] Zhang X, Tian Y, Cheng R, et al. An efficient approach to nondominated sorting for evolutionary multiobjective optimization[J]. IEEE Transactions on Evolutionary Computation, 2014, 19(2):201–213.
- [6] Li K, Deb K, Zhang Q, et al. Efficient nondomination level update method for steady-state evolutionary multiobjective optimization[J]. IEEE transactions on cybernetics, 2016, 47(9):2838–2849.
- [7] Bosman P A, Thierens D. The balance between proximity and diversity in multiobjective evolutionary algorithms[J]. IEEE transactions on evolutionary computation, 2003, 7(2):174–188.
- [8] Zitzler E, Thiele L. Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach[J]. IEEE transactions on Evolutionary Computation, 1999, 3(4):257–271.
- [9] Zitzler E, Künzli S. Indicator-based selection in multiobjective search[C]. Proceedings of International conference on parallel problem solving from nature. Springer, 2004. 832–842.
- [10] Bader J, Zitzler E. HypE: An algorithm for fast hypervolume-based many-objective optimization[J]. Evolutionary computation, 2011, 19(1):45–76.
- [11] Zhang Q, Li H. MOEA/D: A multiobjective evolutionary algorithm based on decomposition[J]. IEEE Transactions on evolutionary computation, 2007, 11(6):712–731.
- [12] Ishibuchi H, Murata T. Multi-objective genetic local search algorithm[C]. Proceedings of Proceedings of IEEE international conference on evolutionary computation. IEEE, 1996. 119–124.
- [13] Jaszkiewicz A. Genetic local search for multi-objective combinatorial optimization[J]. European journal of operational research, 2002, 137(1):50–71.
- [14] Knowles J, Corne D. The pareto archived evolution strategy: A new baseline algorithm for pareto multiobjective optimisation[C]. Proceedings of Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406), volume 1. IEEE, 1999. 98–105.
- [15] Knowles J D, Corne D W. Approximating the nondominated front using the Pareto archived evolution strategy[J]. Evolutionary computation, 2000, 8(2):149–172.
- [16] Paquete L, Chiarandini M, Stützle T. Pareto local optimum sets in the biobjective traveling salesman problem: An experimental study[M]. . Proceedings of Metaheuristics for multiobjective optimisation. Springer, 2004: 177–199.
- [17] Angel E, Bampis E, Gourvés L. Approximating the Pareto curve with local search for the bicriteria

- TSP (1, 2) problem[J]. Theoretical Computer Science, 2004, 310(1-3):135–146.
- [18] Basseur M, Burke E K. Indicator-based multi-objective local search[C]. Proceedings of 2007 IEEE Congress on Evolutionary Computation. IEEE, 2007. 3100–3107.
- [19] Drugan M M, Thierens D. Stochastic Pareto local search: Pareto neighbourhood exploration and perturbation strategies[J]. Journal of Heuristics, 2012, 18(5):727–766.
- [20] Feng L, Ong Y S, Jiang S, et al. Autoencoding evolutionary search with learning across heterogeneous problems[J]. IEEE Transactions on Evolutionary Computation, 2017, 21(5):760–772.
- [21] Yang C, Ding J, Jin Y, et al. Offline data-driven multiobjective optimization: Knowledge transfer between surrogates and generation of final solutions[J]. IEEE Transactions on Evolutionary Computation, 2019, 24(3):409–423.
- [22] Lin J, Liu H L, Tan K C, et al. An effective knowledge transfer approach for multiobjective multitasking optimization[J]. IEEE transactions on cybernetics, 2020, 51(6):3238–3248.
- [23] Liang Z, Liang W, Xu X, et al. A two stage adaptive knowledge transfer evolutionary multitasking based on population distribution for multi/many-objective optimization[J]. arXiv preprint arXiv:2001.00810, 2020..
- [24] Tan K C, Feng L, Jiang M. Evolutionary transfer optimization-a new frontier in evolutionary computation research[J]. IEEE Computational Intelligence Magazine, 2021, 16(1):22–33.
- [25] Jensen M T. Reducing the run-time complexity of multiobjective EAs: The NSGA-II and other algorithms[J]. IEEE Transactions on Evolutionary Computation, 2003, 7(5):503–515.
- [26] Tang S, Cai Z, Zheng J. A fast method of constructing the non-dominated set: arena's principle[C]. Proceedings of 2008 Fourth International Conference on Natural Computation, volume 1. IEEE, 2008. 391–395.
- [27] McClymont K, Keedwell E. Deductive sort and climbing sort: New methods for non-dominated sorting[J]. Evolutionary computation, 2012, 20(1):1–26.
- [28] Kundu D, Suresh K, Ghosh S, et al. Multi-objective optimization with artificial weed colonies[J]. Information Sciences, 2011, 181(12):2441–2454.
- [29] Fourman M P. Compaction of symbolic layout using genetic algorithms[C]. Proceedings of Proceedings of the 1st international conference on genetic algorithms, 1985. 141–153.
- [30] Schaffer J D. Multiple objective optimization with vector evaluated genetic algorithms[C]. Proceedings of Proceedings of the first international conference on genetic algorithms and their applications, 1985. Lawrence Erlbaum Associates. Inc., Publishers, 1985.
- [31] Ishibuchi H, Tsukamoto N, Nojima Y. Behavior of evolutionary many-objective optimization[C]. Proceedings of Tenth International Conference on Computer Modeling and Simulation (UKSIM 2008). IEEE, 2008. 266–271.
- [32] Giagkiozis I, Fleming P J. Methods for multi-objective optimization: An analysis[J]. Information Sciences, 2015, 293:338–350.
- [33] Ishibuchi H, Murata T. A multi-objective genetic local search algorithm and its application to flowshop scheduling[J]. IEEE transactions on systems, man, and cybernetics, part C (applications and reviews), 1998, 28(3):392–403.
- [34] Jin Y, Okabe T, Sendho B. Adapting weighted aggregation for multiobjective evolution strategies[C]. Proceedings of International Conference on Evolutionary Multi-Criterion Optimization. Springer, 2001. 96–110.

- [35] Jaskiewicz A. On the performance of multiple-objective genetic local search on the 0/1 knapsack problem-a comparative experiment[J]. IEEE Transactions on Evolutionary Computation, 2002, 6(4):402–412.
- [36] Paquete L, Stützle T. A two-phase local search for the biobjective traveling salesman problem[C]. Proceedings of International Conference on Evolutionary Multi-Criterion Optimization. Springer, 2003. 479–493.
- [37] Hughes E J. Multiple single objective Pareto sampling[C]. Proceedings of The 2003 Congress on Evolutionary Computation, 2003. CEC'03., volume 4. IEEE, 2003. 2678–2684.
- [38] Ke L, Zhang Q, Battiti R. MOEA/D-ACO: A multiobjective evolutionary algorithm using decomposition and antcolony[J]. IEEE transactions on cybernetics, 2013, 43(6):1845–1859.
- [39] Ke L, Zhang Q, Battiti R. Hybridization of decomposition and local search for multiobjective optimization[J]. IEEE transactions on cybernetics, 2014, 44(10):1808–1820.
- [40] Hillermeier C, et al. Nonlinear multiobjective optimization: a generalized homotopy approach[M], volume 135. Springer Science & Business Media, 2001.
- [41] Blum C, Roli A. Metaheuristics in combinatorial optimization: Overview and conceptual comparison[J]. ACM computing surveys (CSUR), 2003, 35(3):268–308.
- [42] Borges P C, Hansen M P. A basis for future successes in multiobjective combinatorial optimization[J]. Technical 5eport, 'epartment of Mathematical Modelling, Technical University of'enmark, 1998..
- [43] Helsgaun K. An effective implementation of K-opt moves for the Lin-Kernighan TSP heuristic[D]. Roskilde University. Department of Computer Science, 2006.
- [44] Knowles J D, Thiele L, Zitzler E. A tutorial on the performance assessment of stochastic multiobjective optimizers[J]. TIK-report, 2006, 214.
- [45] Czyżżak P, Jaskiewicz A. Pareto simulated annealing—a metaheuristic technique for multiple-objective combinatorial optimization[J]. Journal of Multi-Criteria Decision Analysis, 1998, 7(1):34–47.
- [46] Helsgaun K. Using POPMUSIC for Candidate Set Generation in the Lin-Kernighan-Helsgaun TSP Solver[J]. Roskilde Universitet, 2018, 7.
- [47] Deb K, Miettinen K, Chaudhuri S. Toward an estimation of nadir objective vector using a hybrid of evolutionary and local search approaches[J]. IEEE Transactions on Evolutionary Computation, 2010, 14(6):821–841.
- [48] 郑金华. 多目标进化算法及其应用 [M], volume 2. 科学出版社, 2007.
- [49] Johnson D S, Optimization L. the Traveling Salesman Problem,"[C]. Proceedings of Proc. 17th Colloquium on Automata, Languages and Programming, Springer-Verlag, 1990. 446–461.
- [50] Baraglia R, Hidalgo J I, Perego R. A hybrid heuristic for the traveling salesman problem[J]. IEEE Transactions on evolutionary computation, 2001, 5(6):613–622.
- [51] Johnson D S, McGeoch L A. Experimental analysis of heuristics for the STSP[M]. . Proceedings of The traveling salesman problem and its variations. Springer, 2007: 369–443.
- [52] Lust T, Jaskiewicz A. Speed-up techniques for solving large-scale biobjective TSP[J]. Computers & Operations Research, 2010, 37(3):521–533.
- [53] Lust T, Teghem J. Two-phase Pareto local search for the biobjective traveling salesman problem[J]. Journal of Heuristics, 2010, 16(3):475–510.

- [54] Helsgaun K. An effective implementation of the Lin–Kernighan traveling salesman heuristic[J]. European journal of operational research, 2000, 126(1):106–130.
- [55] Held M, Karp R M. The traveling-salesman problem and minimum spanning trees[J]. Operations Research, 1970, 18(6):1138–1162.
- [56] Held M, Karp R M. The traveling-salesman problem and minimum spanning trees: Part II[J]. Mathematical programming, 1971, 1(1):6–25.

致 谢

在此感谢对本论文作成有所帮助的人。

在学期间的研究成果及学术论文情况

攻读硕士学位期间发表（录用）论文情况

1. 以后可能会在这里也用上 biber
2. 不过目前还需要手写论文全称

研究生期间参与的科研项目

1. 国家自然科学基金 (No.12345678)