汕頭大學
SHANTOU UNIVERSITY

硕 士 学 位 论 文

题　　目：多目标进化算法及其在约束优化中的研究

英文题目：**Research on Multi-objective Evolutionary Computation and Its Constrained Optimization**

姓　　名　王诏君　学　　号　151709063

所在学院　工学院　导师姓名　范　衡

专　　业　电子与通信工程

入学日期　2017 年 9 月　答辩日期　2020 年 6 月

学位论文原创性声明

本论文是我个人在导师指导下进行的工作研究及取得的研究成果。论文中除了特别加以标注和致谢的地方外，不包含其他人或其他机构已经发表或撰写过的研究成果。对本文的研究做出贡献的个人和集体，均已在论文中以明确方式标明。本人完全意识到本声明的法律责任由本人承担。

作者签名：_____ 日期：_____年____月____日

学位论文使用授权声明

本人授权汕头大学保存本学位论文的电子和纸质文档，允许论文被查阅和借阅；学校可将本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、缩印或其他复制手段保存和汇编论文；学校可以向国家有关部门或机构送交论文并授权其保存、借阅或上网公布本学位论文的全部或部分内容。对于保密的论文，按照保密的有关规定和程序处理。

作者签名：_____ 导师签名：_____

日期：_____年____月____日 日期：_____年_____月日

摘要

在实际中，很多实际工程应用问题都可以描述成多目标优化问题。这些目标之间往往相互制约，对其中一个目标优化必须以其他目标作为代价。进化计算开始应用于该问题，并行成了最近的一个热门领域。近年来，随着信息技术、通信技术、大数据的迅猛发展，产生了许多带约束条件的多目标(节能、环保、经济等目标)优化问题。这些问题有着超多变量、超多约束、超多目标的特点，经典优化方法在解决这些问题时性能较差，无法应用于工程实践中。为此，如何有效求解约束多目标优化问题也受到相关学者的关注。本文对多目标进化算法及其在约束优化中的设计进行了讨论与研究。具体如下：

1. 针对多目标优化问题，本文从群体聚合形态形成问题出发，提出了一种基于多目标基因编程的群体聚合形态自动生成的方法。该方法不用依赖设计者的先验知识设计群体机器人群体聚合模式，能根据动态环境和目标信息，自动生成群体聚合模式。综合仿真结果表明，该方法能有效地演化出一些结构简单、对复杂动态环境具有较强鲁棒性的群体聚合形态模型。

2. 针对约束多目标优化问题，设计了一种基于 multi-objective to multi-objective (M2M)处理机制的 Push 和 Pull 的搜索(PPS)框架。该框架将一个多目标约束优化问题分解为一组简单的多目标约束优化问题。每个简单的多目标约束优化问题对应一个子种群，它们之间以协作的方式进行求解。在处理约束时，每个子种群遵循 PPS 搜索过程，有助于每个工作种群跨越不可行的区域。实验结果表明，在 LIR-CMOP 测试问题集上，所提出的 PPS-M2M 算法显著优于其他 7 种经典约束多目标进化算法。在 CIMOP 测试问题集上，所提出的 PPS-M2M 算法显著优于同类算法 PPS-MOEA/D。

关键词：进化算法；多目标优化；约束处理机制；群体聚合形态；

Abstract

In the real world, many practical engineering optimization problems are multi-objective optimization problems. These objectives often restrict each other, and the optimization of one objective must be at the expense of the other objectives. Evolutionary computation has been applied to these engineering optimization problems and become a hot area recently. In recent years, with the rapid development of information technology, communication technology and big data, many constrained multi-objective (energy saving, environmental protection, economic and other objectives) optimization problems have emerged. These problems have the characteristics of a huge number of variables, constraints and objectives. The classical optimization method has poor performance in solving these problems and cannot be applied to engineering practice. For this reason, how to effectively solve constrained multi-objective optimization problems has also attracted the attention of relevant scholars. This paper discusses and studies the multi-objective evolutionary algorithm and its design in constrained optimization. The details are as follows:

1) For multi-objective optimization problems, the paper proposes an automated design framework for GRNs which are used to generate swarm robot patterns. This method does not rely on the a priori knowledge of the designer to design the swarm robot pattern. It can automatically generate the swarm robot pattern based on the dynamic environment and target information. Simulation results show that this method can effectively evolve some swarm robot pattern models with simple structures and strong robustness to complex dynamic environments.

2) In dealing with constrained multi-objective optimization problems (CMOPs), This paper proposes a method which combines a multi-objective to multi-objective (M2M) decomposition approach with the push and pull search (PPS) framework, namely PPS-M2M. To be more specific, the proposed algorithm decomposes a CMOP into a set of simple CMOPs. Each simple CMOP corresponds to a sub-population and is solved in a collaborative manner. When dealing with constraints, each sub-population follows a procedure of “ignoring the constraints in the push stage and considering the constraints in the pull stage”, which helps each working sub-population get across in-feasible regions.

The experimental results show that the proposed PPS-M2M is significantly better than the other seven algorithms and show that the proposed PPS-M2M outperforms PPS-MOEA/D on CIMOPs.

Keywords: Evolutionary Algorithm, Multi-objective Optimization, Constraint-handling Technique, Swarm Robot Pattern;

目　录

摘要..... I
Abstract.....II
目　录.....IV
第 1 章　绪论.....1
1.1 课题研究背景与意义.....1
1.2 进化算法发展简述.....3
1.3 多目标进化算法基础.....5
1.3.1 基于 Pareto 规则的多目标最优解集.....5
1.3.2 MOEA/D 分解方法.....6
1.3.3 多目标进化算法性能评估指标.....8
1.4 主要研究内容及结构安排.....8
第 2 章　基于多目标基因编程的群体聚合形态自动生成框架.....10
2.1 基因编程.....10
2.2 复合基因编程.....11
2.3 实验研究.....13
2.3.1 群体聚合形态当前研究背景.....13
2.3.2 基于群体聚合形态生成的基本元库定义.....14
2.3.3 基于群体聚合形态生成的适应度函数定义.....16
2.3.4 基于复合多目标基因编程的群体聚合形态自动设计框架.....17
2.3.5 实验与分析.....19
2.4 群体聚合形态自动形成的自动优化设计系统.....30
2.5 本章小结.....31
第 3 章　约束处理机制国内外研究现状.....32
3.1 带约束的多目标优化问题.....32
3.2 基于进化算法的约束处理机制研究现状.....33
第 4 章　基于 M2M 处理机制的 Push 和 Pull 的搜索框架..... 35
4.1 约束多目标进化算法 PPS-M2M.....35
4.1.1 M2M 分解方法.....35
4.1.2 PPS 框架.....37

4.1.3 PPS-M2M 算法流程.....39
4.1.4 PPS-M2M 与 PPS-MOEA/D 的差异.....43
4.2 多目标约束优化测试问题集.....43
4.3 实验研究.....47
4.3.1 几种经典的约束多目标进化算法.....47
4.3.2 实验参数设置.....47
4.4 实验分析及结论.....48
4.4.1 LIR-CMOP 测试问题实验结果.....48
4.4.2 CIMOP 测试问题实验结果.....56
4.5 本章小结.....61
第 5 章 总结.....62
参考文献.....64
攻读硕士学位期间主要的工作成果.....74
致谢.....75
附录.....76

第 1 章　绪论

英国生物学家查尔斯·达尔文通过 20 多年对生物研究在《物种起源》里提出“物竞天择，适者生存”[1]。自然界每时每秒都在变化的，根据《物种起源》，由于地球运动，动植物生存的环境都是时刻变化的，这导致各种生物(如动物、植物、微生物等)的形态或者机能也是总是变化的。值得注意的是，这些变化会使物种产生相应的基因(这些基因，有对生物的生存起到积极作用的；也有对生物的生存起到消极作用的)。生物为了生存，只能通过外在条件(如捕食者、自然变化、自身筛选)选择，把适应性强的个体保留下来，从而保证优良的变异基因可以延续下来。这就造成了各种生物(如动物、植物、微生物等)随着时间推移总是在进化的。进过的过程总是朝着更有利生物生存方向进行。

随着社会和科技的发展，数值优化问题变得更加复杂。为了更加快速的和高效的求解计算人员无法很容易求解的优化问题(如超多目标优化问题、超多变量的优化问题等)，世界各地的研究人员利用进化论的思想，经过多年的研究提出了一类启发式、随机式优化算法——进化算法(Evolutionary Algorithms, EAs)[2]。该算法继承了进化论的思想(“自然选择，优胜略汰”)。

随着进化计算技术发展(如一大批求解高效的算法被提出)，各类实际工程优化问题被解决。但是，由于此类优化问题通常含有多个目标/超多目标，这些目标之间往往相互制约，对其中一个目标优化必须以其他目标作为代价。此外，此类问题有时会带有约束[3]，所以进化算法方向的研究必然引申到对多目标共同优化和相关约束处理机制的研究。

1.1 课题研究背景与意义

一般情况下，优化问题是都是求解优化目标(如经济学问题中效益最大和成本最小)最小化/最大化问题。为了描写方便，本文利用一些数学变换求解优化目标最小。

一个最优化问题可以描述为以下形式：

$$\text{minimize } \vec{F}(\vec{x}) = (f_1(\vec{x}),f_2(\vec{x}),...,f_m(\vec{x})) \tag{1-1}$$

其中，$\vec{F}(\vec{x})$ 为目标函数，其中 $(f_1(\vec{x}),f_2(\vec{x}),...,f_m(\vec{x}))$ 是 m 个子目标，$\vec{x} =$

$(x_1,x_2,...,x_D \in R)$ 是 D 维决策变量。当 $m=1$ 时，最优化问题是一个单目标优化问题；当 $2 \leq m \leq 3$ 时，最优化问题是多目标优化问题；当 $m>3$ 时，最优化问题是高维目标优化问题。$R=\prod_{k=1}^{D}[L_k, U_k]$代表 D 维决策空间，U_i、L_i分别代表决策向量上第 i 位的取值上下界。

为了高效的解优化问题，传统的优化算法(梯度下降法[4]、牛顿法[5]、禁忌搜索(Tabu Search, TS)[6]、模拟退火(Simulate Annealing, SA)[7])被前人陆续设计出来了。

梯度下降法[4]的基本思想可以类比为一个下山的过程。为了保证梯度下降法准确求解，需要满足以下要求：1). 需要准确知道所求的目标函数；2). 目标函数可导；3). 目标函数是凸函数。虽然梯度下降法是一种最简单的求解最优化问题方法，但是它有以下不足之处：1). 靠近极小值(最小值)时收敛速度减慢；2). 可能会“之字形”地下降，从而不能准确找到极小值(最小值)。牛顿法[5]是一种在实数域和复数域上近似求解方程的方法。牛顿法在收敛速度上比梯度下降法(梯度下降法是一阶收敛)要快。但是此类方法有一定局限性，如计算比较复杂。禁忌搜索方法[6]是一种用来跳出局部最优的搜索方法。此类方法能够实现让目标函数值持续减少，也能指导下一步搜索方法。模拟退火方法[7]模拟金属退火原理。它的不足之处是其参数不容易设计，需通过大量试错实验来确定。

以上四种算法都有一定的局限性，主要表现在以下三个方面：1).不能高效的和快速的找到最优解；2).需要严格的设置可解决问题条件；3).对现实中常见的黑箱问题无能为力。

进化算法与传统的求解最优化问题算法相比较，具有独特的并行计算的特点，提高了运算速度。与此同时，进化算法不仅是在求解单峰最优化问题还是求解多峰优化问题上，都可以保证求解过程快速跳出局部最优解，从而达到全局最优解。进化算法与以上四种算法(梯度下降法、牛顿法、禁忌搜索方法和模拟退火方法)相比最显著的特点是进化算法不需要额外的梯度信息和一些其他的先验信息(如目标函数是否具有凸凹性、目标函数是否可导和目标函数是否具有连续性和离散型等)，所以进化算法可以被广泛应用到超多变量、超多约束、超多目标的最优化问题、复杂、多变的黑箱问题和非线性问题求解。

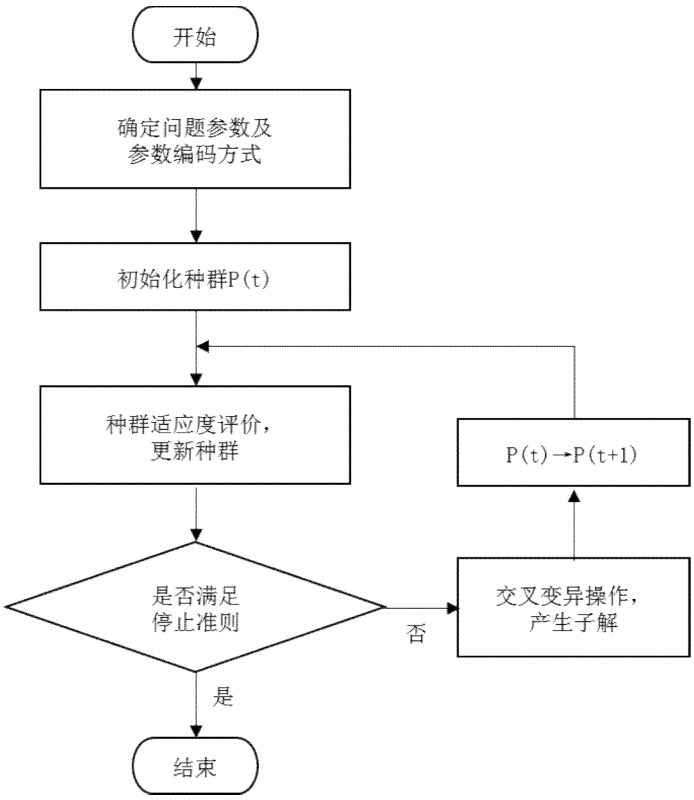


图 1-1　进化算法基本流程框图

在原理上，进化算法与经典的优化搜索算法相比，进化算法不用从单个解搜索到全局最优解，而是通过多解并行搜索方式向全局最优解靠近，如图 1-1 所示。大致过程如下，首先根据设计人员的需求设置一些个体(个体数一般大于 100)和确定个体的编码方式，再通过遗传因子，产生一些子个体。这些个体作用是为了向种群中增加新的基因。接着，通过设计人员设计的筛选机制选择符合的个体。反复进行多代，直到满足最后终止条件即可。

1.2 进化算法发展简述

进化计算通过很多代研究者共同努力多目标进化算法在这些年越来越被人所关注。

虽然在 1772 年, Franklin 提出了如何协调多目标冲突的问题，但是学术界一般认为多目标优化问题(MOPs)是 1896 年法国经济学家 V .Pareto 提出来的[8]。V .Pareto 从经济学中人们常遇见经济问题出发，把无法结合比较的多目标归结为多目标优化问题。1951 年，T .C .Koopmans[9]利用自己在生产分配活动中的研究提出来一个影响到现在的概念——Pareto 最优解(非支配解)。随着 1965 年 Rechenberg 和 Fogel 提出进

化策略(Evolutionary Strategy, ES)[10]和进化规划(Evolutionary Programming, EP)[11] 等概念，进化计算的思想渐渐复现在人们视野里面。1968 年，Z .Johnsen 通过前人研究系统地提出了多目标决策问题的研究报告。和同年 Holland[12]首次正式地、科学地提出遗传算法(Genetics Algorithm, GA)概念。Z .Johnsen 和 Holland 让 1968 年成为多目标优化发展的转折点，为后人对多目标优化研究敲开了“开门砖”。虽然进化计算的思想在 20 世纪 70 年代左右相继提出和完善，但是仍旧无法解决一些关键问题。David Goldberg[13] 和他的团队经过多年的努力，在 1989 年时，设计出系统化 MOEA 的理论体系。随着科技的发展和运算能力的提升(计算机的出现)，20 世纪 90 年代的 MOEA 研究如雨后春笋一般，得到快速发展。众多优秀的 MOEA 被相继提出，最具代表性的算法包括多目标进化算法(MOGA)[14]，非支配遗传算法(NSGA)[15]，和小生境遗传算法(NPGA)[16]。1999 年，Eckart Zitzler[17]等人提出了 Strength Pareto Evolutionary Algorithm(SPEA)，第一次提出了精英保留策略，直接将当前种群中优秀的个体保留到下一代中。后面的研究更是将非支配排序的机制和精英保留策略结合起来，提出了 Pareto Envelope-Based Selection Algorithm(PESA)[18] 、 Pareto Archived Evolution Strategy(PAES)[19]、SPEA2[20]、NSGA-II[21]以及 NPGA2[22]等。20 世纪 90 年代，美国的 Koza 等人[23]提出了遗传程序设计(Genetics Programming, GP)。至此，ES、EP、GA 和 GP 正式成为进化算法的四大经典分支[24]，并在随后 20 多年里得到了快速发展。

20 世纪以后，由于研究多目标进化算法的人越来越多，造就了多目标进化算法研究的角度丰富多样。一大批机制被研究出来，并应用到多目标进化计算领域。例如，Eckart Zitzler[25]提出了基于指标的 Indicator-based Evolutionary Algorithm(IBEA)。2005 年，Coello Coello 等人[26]通过大量研究提出了多目标粒子群算法(MOPSO)。2007 年，张青富教授[27]运用运筹学的方法，将多目标优化问题分解为多个单目标优化子问题，然后同时求解各个子问题，提出了一种基于分解的多目标进化算法(MOEA/D)。至此，基于 Pareto 的多目标进化算法、基于分解的多目标进化算法和基于指标的多目标进化算法正是成为多目标进化算法的三大研究方向[28]。

大量研究者在这三大研究方向上深入研究，提出来各种各样的改进版本。例如：对于基于 Pareto 的多目标进化算法的改进版本有：NSGA-III[29]，U-NSGA-III[30]，i-NSGA-II[31]以及 NSGA-Net[32]等。对于基于分解的多目标进化算法的改进版本有：MOEA/D-DE[33]，MOEA/D-ACDP[34]，EAG-MOEA/D[35]，MOEA/D-SAS[36]等。对于基于指标的多目标进化算法的改进版本有：IBEA-SVM[37]，IBEA[38]，mIBEA[39]，HypE[40]等。

1.3 多目标进化算法基础

一般情况下，根据目标函数个数，优化问题可以划分为单目标优化问题，多目标优化问题和高维目标优化问题，其中用于解决两/三目标优化问题(Multi-objective Optimization Problems, MOPs)的进化算法，被称为多目标进化算法(Multi-objective Evolutionary Algorithms, MOEAs)。当优化问题的优化目标大于三个时，此时优化问题被称为高维目标优化问题(Many-objective Optimization Problems)。用于解决此问题的进化算法，被称为高维目标进化算法(Many-objective Evolutionary Algorithms)。

1.3.1 基于 Pareto 规则的多目标最优解集

不像单目标优化问题求解出来的目标函数呈现标量形式那样，多目标优化问题求解出来的目标函数往往呈现一个向量形式。于此同时，在多目标优化问题中各目标之间通过决策变量相互制约，对其中一个目标优化必须以其他目标劣化作为代价，也就是说，要同时使多个子目标一起达到最优值是不可能的，而只能是在它们中间进行折衷(Tradeoffs)考虑，使各个子目标函数都尽可能地达到最优。与单目标优化问题的本质区别在于，多目标优化问题的解不是唯一的，而是存在一个最优解集合。因此在求解多目标优化问题时，我们通常会引入 Pareto 规则，从而得到了一个折衷的多目标最优解集。

定义 1.1：Pareto 支配

对于决策空间 R 内的任意两个解$\vec{x}_a$和$\vec{x}_b$，若满足以下 Pareto 支配规则，可认为$\vec{x}_a$优于$\vec{x}_b$，或称$\vec{x}_a$支配$\vec{x}_b$，记做$\vec{x}_a \prec \vec{x}_b$：

1)对于所有的目标函数，$\vec{x}_a$不比$\vec{x}_b$差。

2)至少存在一个目标函数，$\vec{x}_a$比$\vec{x}_b$好。

即

$$\forall i\, f_i(\vec{x}_a) \le f_i(\vec{x}_b) \wedge \exists j\, f_j(\vec{x}_a) < f_j(\vec{x}_b) \tag{1-2}$$

其中 $i,j \in \{1,2,\ldots,m\}$，m 为目标函数的数目。

定义 1.2：Pareto 解集

如果在对于决策空间 R 内存在一个解$\vec{x}^*$，不被 R 内其他任意一个解支配，那么$\vec{x}^*$被称为一个 Pareto 最优解。R 内的所有 Pareto 最优解组成的集合，称为 Pareto 解集(Pareto Set, PS)。

定义 1.3：Pareto 前沿

PS 在目标空间上的映射向量集合称为 Pareto 前沿(Pareto Front, PF)。可定义为以

下形式：$PF = \{\vec{F}(\vec{x})|\vec{x} \in PS\}$。

1.3.2 MOEA/D 分解方法

MOEA/D 作为一种经典的解决多目标优化问题的算法，其核心思想是将多目标优化问题被转化为一系列单目标优化子问题[27]，然后利用一定数量相邻问题的信息，采用进化算法对这些子问题同时进行优化。MOEA/D 的一般算法流程如图 1-2 所示。

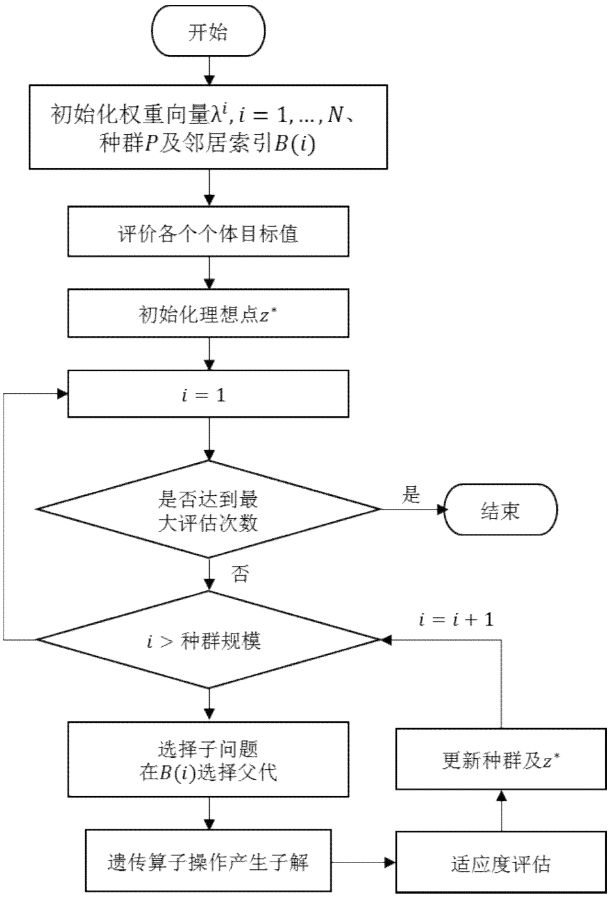


图 1-2 MOEA/D 算法流程图

对于 MOEA/D，首先需要考虑如何把一个 m 目标优化问题分解出 N 个子问题。为此，在 MOEA/D 中，需要把目标搜索区域均匀分成 N 个权重向量($\lambda = (\vec{\lambda}^1,\ldots,\vec{\lambda}^N)$是 N 个权重向量集合)，其中的任意一个权重向量$\vec{\lambda}^i = (\lambda_1^i,\ldots,\lambda_m^i)^T, i = 1,2,\ldots,N$ 需满足以下条件：

$$\sum_{k=1}^{m} \lambda_k^i = 1 \text{ 且 } \lambda_k^i \geq 0, k \in \{1,2,\ldots,m\} \tag{1-3}$$

通过它们将一个 m 目标优化问题分解出 N 个子问题。

在 MOEA/D 中，张青富教授等人提出了三种常用的分解子问题的方法：加权求和法(Weighted Sum Approach)[49]、切比雪夫法(Tchebycheff Approach)[49]、边界交叉法(Boundary Intersection Approach)[50]，以下对这三种方法作详细介绍：

1）加权求和法

对于一个多目标优化问题，可通过加权求和的方式，把这个多目标优化问题转化为多个单目标优化子问题。假设一个多目标优化问题有 m 个目标，通过一个权重向量$\vec{\lambda}^i$对应分解得到的第 i 个子问题，其数学表达形式如下所示：

$$\begin{cases}\text{minimize } g^{ws}(\vec{x}|\vec{\lambda}^i)=\sum_{k=1}^{m}\lambda_k{}^i f_k(x)\\ \text{subject to } \vec{x}\in R\end{cases} \tag{1-4}$$

式中$g^{ws}(\vec{x}|\vec{\lambda}^i)$为第 i 个子问题的聚合函数，$\vec{x}$为决策变量，R 为决策变量搜索空间。加权求和法只适用于求解凸的优化问题，因为这种方法不能获得每一个最优 Pareto 最优向量，所以有一定的局限性。

2）切比雪夫法

在 MOEA/D 中，第二种将一个多目标优化问题转化为多个单目标优化子问题的方法是切比雪夫方法，其数学表达形式如下所示：

$$\begin{cases}\text{minimize } g^{te}(\vec{x}|\vec{\lambda}^i,\vec{z}^*)=\max\limits_{1\le k\le m}\{(1/\lambda_k^i)\cdot|f_k(\vec{x})-z_k^*|\}\\ \text{subject to } \vec{x}\in R\end{cases} \tag{1-5}$$

式中$g^{te}(\vec{x}|\vec{\lambda}^i)$为第 i 个子问题的聚合函数，$\vec{x}$为决策变量，R 为决策变量搜索空间。其中$\vec{z}^*=(z_1^*,z_2^*,\ldots,z_m^*)^T$为理想点，且$z_i^*=min\{f_i(\vec{x})|\vec{x}\in R\}$，$i=1,2,\ldots,m$。此方法的不足之处是在处理连续多目标问题时聚合方法曲线不平滑。

3）边界交叉法

边界交叉方法是 MOEA/D 中最后一种分解方法，也是一种经典的分解方法。边界交叉方法的数学表达形式如下所示：

$$\begin{cases}\text{minimize } g^{bip}(\vec{x}|\vec{\lambda}^i,\vec{z}^*)=d_1+\theta d_2\\ \text{subject to } \vec{F}(\vec{x})-\vec{z}^*=d_1\vec{\lambda}^i\\ d_2=\|\vec{F}(\vec{x})-\vec{z}^*-d_1\vec{\lambda}^i\|\\ \vec{x}\in R\end{cases} \tag{1-6}$$

式中$g^{bip}(\vec{x}|\vec{\lambda}^i)$为第 i 个子问题的聚合函数，$\vec{x}$为决策变量，R 为决策变量搜索空

间。其中$\vec{z}^* = (z_1^*,z_2^*,...,z_m^*)^T$为理想点，且$z_i^* = min\{f_i(\vec{x})|\vec{x} \in R\}$，$i = 1,2,...,m$。

在 MOEA/D 算法中，每个权重向量$\vec{\lambda}^i$关联一个解$\vec{x}_i$，MOEA/D 通过随机选择$\vec{\lambda}^i$的两个邻居向量关联的解作为父解，然后通过遗传操作产生子解$\vec{y}_i$，从而根据聚合函数值更新$\vec{\lambda}^i$上及其邻居向量上关联的解。一般地，以切比雪夫方法为例，对于一个个体解$\vec{x}^i$，互为邻居的两个权重向量$\vec{\lambda}^j$和$\vec{\lambda}^k$所构成的聚合函数的值$g^{te}(\vec{x}_i|\vec{\lambda}^j,\vec{z}^*)$和$g^{te}(\vec{x}_i|\vec{\lambda}^k,\vec{z}^*)$是比较接近的。

1.3.3 多目标进化算法性能评估指标

一般而言，为了评价一个设计出来的多目标进化算法的性能，需要一些评价标准将多目标进化算法的性能可数值化的和可直观化的表现出来。目前，国内外研究学者针对 MOEA 提出了许多评价工具，如：Generational Distance (GD)[51]、Inverted Generational Distance (IGD)[52]、Hypervolume (HV)[53]、Convergence Metric γ[54]、Diversity Metric Δ[55]、C-metric[56]、Knee-driven Dissimilarity[57]等。主要评估 MOEA 的两类性能：收敛性和分布性(或称多样性)。在对 MOEA 进行评估时，其实在比较已得的 Pareto 前沿 PF_{known}与实际的 Pareto 前沿 PF_{true}之间的差距。对于 MOEA 的收敛性是指：1) PF_{known}尽可能趋近于 PF_{true}；2)PF_{known}尽可能完整布满整个 PF_{true}。对于 MOEA 的多样性是指：1)PF_{known}尽可能地均匀分布；2) PF_{known}尽可能拥有较大的分布广度。

1.4 主要研究内容及结构安排

本文主要对多目标进化算法及其约束中的优化进行研究。针对无约束的多目标优化问题和带约束的多目标优化问题，分别设计相应的进化算法。值得注意的是，针对无约束的多目标优化问题设计出来的进化算法应用到了群体机器人围捕任务，即利用复合多目标基因编程的算法自动生成群体聚合形态(包围圈形态)。根据“没有免费午餐”理论 [58]，本文针对多目标约束优化问题提出一种基于 multi-objective to multi-objective (M2M)处理机制的 Push 和 Pull 的搜索框架(PPS)搜索框架。该框架将一个 CMOPs 分解为一组简单的 CMOPs。每个简单的 CMOPs 对应一个子种群，它们之间以协作的方式进行求解。在处理约束时，每个子种群遵循 PPS 搜索过程，有助于每个工作种群跨越不可行的区域。实验结果表明，在 LIR-CMOP 测试问题集上，所提出的 PPS-M2M 算法显著优于其他 9 种经典 CMOGAs。在 CIMOP 测试问题集上，所提出的 PPS-M2M 算法显著优于同类算法 PPS-MOEA/D。

主要的结构如下：

第一章通过调研相关文献，简述了进化算法发展史。此外，还简单介绍常用的进化算法和一些传统的优化算法，以及它们的特点。最后简单列举一些常用的评价进化算法的指标。

第二章提出一种应用于群体聚合形态自动生成的复合多目标基因编程。本章首先对群体聚合形态和多目标基因编程进行介绍，然后介绍自动生成群体聚合形态的假设条件。最后，根据在基于基因调控网络的群体聚合形态生成方面，现有的群体聚合形态生成大多需要人类设计人员根据经验设计网络结构，难以适应复杂多变环境的局限。因此，本文提出了一种适用于群体模式自动生成的基因调控网络(Gene Regulatory Network，GRN)进化设计框架。通过仿真验证，所提出的自动进化设计框架设计出来的 GRN 有很好的性能。

第三章指出在现实世界中优化问题都含有一些困难源。除了巨大的搜索空间，目标函数中的噪声以及建模过程的复杂性之外，还指出了约束的存在。约束可能会导致搜索将焦点从优化转移到寻找一个可行的(即有效的)解上。另一方面，在自然启发式算法刚出来的时候，它旨在处理不受约束的搜索空间，需向自然启发式算法添加约束处理技术，旨在通过可行的解将搜索引导到那些区域。

第四章提出一种基于 multi-objective to multi-objective (M2M)处理机制的 Push 和 Pull 的搜索框架(PPS)搜索框架。所提出来的算法首先与其他十一种算法在 LIR-CMOPs 测试问题上进行对比，实验所提出来的算法明显优于其他算法。然后，在 CIMOP 测试问题集上，所提出的 PPS-M2M 算法与同类算法 PPS-MOEA/D 进行对比，来解释所提的算法与 PPS-MOEA/D 的区别以及适用问题。

最后，本文进行总结。总结了这篇论文主要内容：1) 为了解决群体聚合形态形成，提出了一个基于多目标基因编程的群体聚合形态自动生成框架；2) 针对约束多目标优化问题的特点，提出了一种新的带约束的进化算法——PPS-M2M，该算法能很好解决目标函数不平衡问题和具有较大不可行区域问题。

第 2 章　基于多目标基因编程的群体聚合形态自动生成框架

大多数现有的用于群体聚合形态生成的基因调控网络(Genetic Regulatory Networks，GRN)的设计取决于设计人员的先验知识。因此，为了适应复杂和动态的环境，GRN 的设计通常是反复试验的过程。此外，通过对此类问题进行分析，发现是一个多目标优化问题。基于此，在这项工作中，我们提出了用于 GRN 的自动化设计框架，该框架用于生成群体聚合形态。所提出的框架应用了一种复合多目标基因编程算法(Composite Multi-objective Genetic Programming，CMOGP)，该方法将多目标基因编程(Multi-objective Genetic Programming，MOGP)[59]、非支配排序遗传算法 (Nondominated Sorting Genetic Algorithm II， NSGA-II)[21]、差分进化算法(Differential Evolution Algorithm， DE)[43]和多目标决策(Multi-Objective Decision Approach)[60]相结合。此外，所提出的框架应用了一些基本的网络元件来自动设计 GRN，而无需任何先验知识。在 GRNs 的设计过程中，MOGP 和 DE 用于并行优化 GRNs 的结构和参数。 NSGA-II 用于平衡 GRN 模型的复杂性和性能，从而获得满足当前约束环境的 GRN 模型。采用多目标决策来选择帕累托最优解中的所需 GRN 模型。全面的仿真结果表明，提出的框架可以有效地开发一些新颖的 GRN，其结构要比人类专家的设计更简单，并且对复杂和动态的环境更加稳定。

2.1 基因编程

基因编程(Genetic Programming，GP) [23][59]是遗传算法(Genetic Algorithm，GA)的一个分支，它只用对计算机下达指令(我需要什么)，而不用告诉计算机怎么去实现，最终，通过适应度函数来评估在预定义任务上的工作情况。值得说明的是，在这个过程中，计算机程序被编码成一个基因组，它可以是一个语法树(a syntax tree)[61]、一个指令序列[62]、或其他线性和层次结构的数据结构[63]。

在进化过程中，GP 从基因组群开始，基因组群通常是随机产生的，并在一系列代数中逐步进化。在每一代中，GP 首先根据适应度选择当前种群的一部分，然后对选择的种群进行交叉和变异等遗传操作，产生新的基因组，形成下一个种群。值得注意的是，当对选择的种群进行遗传操作时，语义树的部分枝干与其他语义树的部分枝干进行交叉，同个语义树的操作符函数变异成其他操作符函数。

GP 中的种群个体是由操作符函数(Operator Function)和终端(Terminal Node)组成的层次结构树。不仅每个树节点都有一个操作符函数，而且每个终端节点都有一个操作数。树形结构使得数学表达式易于演化和计算。在本例中(如图 2-1 所示)，树表示如下模型：

$$y = \frac{x_1}{x_2} + x_2 - x_1 \tag{2-1}$$

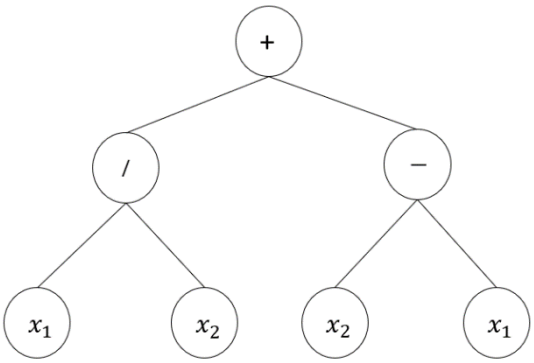


图 2-1 基因编程中的层次结构树。数学运算符{/，+，-}是运算符函数。x_1 和 x_2用作终端节点。

由于在运用 GP 时，具有以下优点：1)不需要预先知道求解目标和解决方案的相关知识，只需要定义好操作符函数之间的连接关系，再通过适应度函数进行选择即可；2)通过 GP 得到最终解，通过操作符函数之间的连接关系可以很好解释，从而方便研究者挖掘出来大量的知识。因此，近几年 GP 广泛被应用到各个领域上。

2.2 复合基因编程

在现实世界中，很多实际工程优化问题都是多目标问题。这些目标之间往往相互制约，对其中一个目标优化必须以其他目标作为代价。为了处理这样目标，我们必须对传统的 GP 算法进行一定程度的改进，使它能优化具有多个相互制约关系的目标。接着，由于传统的 GP 在优化参数时，没有一定的导向性，即在参数的取值范围内随机选择，这样导致了不能高效的优化参数。此外，GP 是 GA 的一种。当达到终止条件时，往往是由多个个体组成的最优解集。虽然这些个体都是最优解集，但是不方便研究人员从中选择。

为了克服这些缺点，我们提出一种复合多目标基因编程算法(a composite multi-objective genetic programming， CMOGP)，该方法将多目标基因编程(a multi-objective genetic programming，MOGP)[59]、非支配排序遗传算法(a nondominated sorting genetic algorithm II， NSGA-II)[21]、差分进化算法(a differential evolution

algorithm，DE)[43]和多目标决策(a multi-objective decision approach)[60]相结合。在CMOGP 中，NSGA-II 被用来平衡多个冲突的目标值，从而进化搜索出来最优的解集。为了更好指导 MOGP 的参数优化，DE 被应用到 CMOGP 里。最后，利用多目标决策方法，在帕累托最优解中选择出最有期望的解。CMOGP 的流程如图 2-2 所示。在图 2-2 中，首先，随机初始化 N 个个体，这些个体是通过 MOGP 初始化生成的。然后，利用 DE 去优化这 N 个个体里的参数，使这些参数成为当前个体中的较优参数。紧接着，通过使用 MOGP 去优化这 N 个个体的结构，从而产生新的 N 个结构个体。最后，利用终止条件来判断是否达到终止条件，如果没有达到终止条件，这些新的 N 个个体将进入下一次循环当中；如果达到终止条件，利用多目标决策机制输出最优的非支配个体。

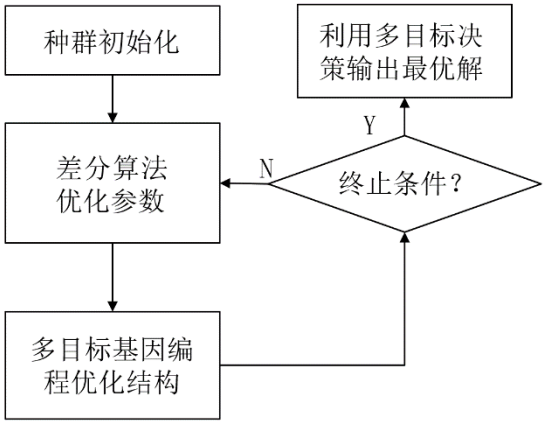


图 2-2 复合多目标基因编程的流程图

此外，CMOGP 的伪代码如算法 2-1 所示。在算法 2-1 中，1~3 行是算法的初始化阶段。即在第 1 行，使用 Ramped half-and-half 方法[64]创建初始化种群P_0。在第 2 行，利用 DE 去优化初始种群P_0中每个个体的参数。在第 3 行，适应度函数被用于评估初始种群中每个个体的适应度值。5~14 行是算法的主体部分。即通过交叉、变异和重组等遗传算子产生新的子代群体G_g。在第 7-10 行里，利用 DE 对子代群体中每个个体 q 的进行参数优化，然后通过适应度函数的评价得到每个个体 q 的适应度值。在第 12 行里，新的种群P_{g+1}被 NSGA-II 选择出来，并进入下一次循环里。在第 13 行里，更新计数器。第 15 行是算法输出阶段，即在第 15 行，通过多目标决策选择一个期望的非支配可行解$\vec{x}_{best}$。

算法 2-1：用于群体聚合形态生成的 CMOGP 算法设计

输入：

1)G：特点任务(场景)下输入因素；

2)gen_{max}：最大代数。

输出：期望的非支配可行解$\vec{x}_{best}$。

1 初始化：利用 Ramped half-and-half 方法随机生成 N 个个体，且存放在初始种群P_0里；

2 优化参数：利用 DE 去优化种群P_0里 N 个个体的参数；

3 评价：通过适应度函数评价种群P_0里 N 个个体；

4 设置 $gen = 0$；

5 当 $gen \leq gen_{max}$时

6 　　利用遗传操作生成子种群G_g，且G_g里包含 N 个个体；

7 　　循环种群G_g里每个个体 q，即 $q \in G_g$

8 　　　　利用 DE 去优化个体 q 的参数；

9 　　　　通过适应度函数评价个体 q；

10 　　循环结束

11 　　$R_g \leftarrow P_g \cup Q_g$；//$P_g$为第 g 代父代种群；R_g为第 g 代父代种群与第 g 代子种群融合的种群；

12 　　利用 NSGA-II 选择机制，从R_g中选择新的种群P_{g+1}，且种群P_{g+1}含有 N 个个体；

13 　　$gen = gen + 1$；

14 判断结束

15 输出：利用多目标决策算法从 Pareto 解集里选择一个期望的非支配可行解$\vec{x}_{best}$。

2.3 实验研究

在本节中，为了说明本章所提出的 CMOGP 的性能，CMOGP 将应用到群体聚合形态生成中。本节将从群体聚合形态当前研究现状出发指出其不足之处。然后，为了弥补此不足之处，本章提出基于复合多目标基因编程的群体聚合形态自动设计框架，并且设置出三个场景来体现该框架的有效性。这三个场景分别为：1)目标从两个圆形障碍物中穿过；2)目标从具有挑战性障碍物的通道中穿过；3)将前两个场景优化得到的个体迁移到圆形障碍物随机分布的环境中进行测试。

2.3.1 群体聚合形态当前研究背景

一般来说，群机器人系统(SRSs)是由许多最小的、简单的和低成本的机器人组成，每一个机器人都有有限的功能和性能[65]。然而，这些简单的机器人可以以协作

的方式一起工作，完成单个机器人无法完成的复杂和具有挑战性的任务。因为 SRSs 具有并行性、可扩展性、稳定性、低成本、鲁棒性强、适应性强等特点，广泛应用于协同目标搜索救援[66][67]、卫星通信[68][69]、编队变换[70][71]、源定位[72][73]等领域。

许多研究人员试图从动物集体行为和细胞行为中获得灵感的原则，从而实现 SRSs。这两种行为的共同点是群体机器人的群体聚合形态生成，具体体现在形态生成和形态保持两个方面。更具体地说，多机器人与邻域机器人相互合作或者多机器人与周围环境信息交互生成某种群体形态。这形态的模式可以是预定义的或自适应形成的[65]。

此外，近年来，群体模式的生成得到了广泛的探索，根据控制机制可分为四类[65]，即形态发生(Morphogenesis)[74][75]、反应扩散模型(Reaction-Diffusion Model)[76][77][78]、趋化性(Chemotaxis)[79]和基因调控网络(Gene Regulatory Network，GRN)[80]。在本章中，我们将 GRN 应用于群体模式生成。基于 GRN 的群体模式生成是受生物进化的启发，将基因进化的过程映射到群体模式生成中。群机器人可以对静态和动态目标进行自组织的包围。

近十年来，许多研究者在基于 GRN 的群体模式生成领域做了大量的研究。例如，Jin 等人[81]提出了一种用于多机器人模式生成的分级基因调控网络(H-GRN)。在这项工作中，该模型由两层 GRN 组成。第一层被设计成一个预定义的群体聚合形态生成。第二层被用来控制群体机器人运动到第一层生成的群体聚合形态上。为了躲避障碍物，Peng 等人[82]提出了一种改进的 GRN 来包围多个目标。在这项工作中，他们只是对 H-GRN 的底层进行了改进。Oh 等人[83]提出了一种用于群体聚合形态生成的进化层次基因调控网络(EH-GRN)，以便在受限环境中生成自适应模式来包围目标。在这项工作中，仅仅是对 GRN 模型的参数进行优，而 GRN 模型的结构是由专家预先定义的。这些现有的 GRN 模型要么由人工专家预先定义其结构和参数，要么由人工专家预先确定其固定结构再进行参数优化，使其适应变化的、复杂的、动态的环境的能力受到限制。

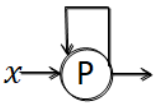
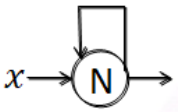
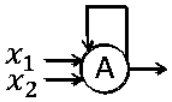
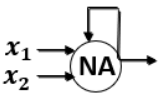
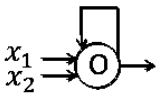
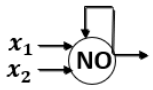
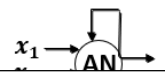
为了解决现有 GRN 模型的局限性，我们提出了一种基于复合多目标遗传编程的群体聚合形态生成的自动化设计框架。该框架不需要提前定义 GRN 模型的结构。相反,采用一组基本的网络元件自动生成 GRN 模型。

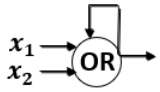
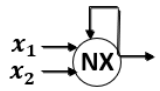
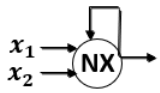
2.3.2 基于群体聚合形态生成的基本元库定义

最近的生物学研究结果表明，存在一定功能的网络模块往往出现在真实的复杂网络中[84]。Peter M. Bower[85] 通过逻辑分析发现三联体蛋白质之间的逻辑关系。此外，多细胞生物的基因调控网络也经常使用到这些逻辑关系。受这些研究的启发，

十种逻辑关系(positive、negative、AND、NAND、OR、NOR、ANDN、ORN、XOR和 XNOR)被用作基本的网络模块，用来构造基因调控网络。基因调控网络基本元件库：

表 2-1 基本元件库

元件名称	描述	关系方程	图例
Positive	基因x激活基因y。	$\frac{dy}{dx}=-y+sig(x,\theta,k)$ $sig(x,\theta,k)=\frac{1}{1+e^{-k(x-\theta)}}$ 其中，θ,k 为元件参数。	
Negative	基因x抑制基因y。	$\frac{dy}{dx}=-y+[1-sig(x,\theta,k)]$ $sig(x,\theta,k)=\frac{1}{1+e^{-k(x-\theta)}}$ 其中，θ,k 为元件参数。	
AND	当基因x_1和x_2同时表达时，则基因y表达。	$\frac{dy}{dx}=-y+[1-sig(x_1*x_2,\theta,k)]$ $sig(x,\theta,k)=\frac{1}{1+e^{-k(x-\theta)}}$ 其中，θ,k 为元件参数。	
NAND	当基因x_1不表达或基因x_2不表达时，则基因 y 表达	$\frac{dy}{dt}=-y+sig(x_1*x_2,\theta,k)$ $sig(x,\theta,k)=\frac{1}{1+e^{-k(x-\theta)}}$ 其中，θ,k 为元件参数。	
OR	当基因x_1或者基因x_2表达时，则基因y表达。	$\frac{dy}{dx}=-y+[1-sig(x_1+x_2,\theta,k)]$ $sig(x,\theta,k)=\frac{1}{1+e^{-k(x-\theta)}}$ 其中，θ,k 为元件参数。	
NOR	当基因x_1不表达并且基因x_2不表达时，则基因 y 表达	$\frac{dy}{dt}=-y+[1-sig(x_1+x_2,\theta,k)]$ $sig(x,\theta,k)=\frac{1}{1+e^{-k(x-\theta)}}$ 其中，θ,k 为元件参数。	
ANDN	当基因x_1表达并且基因	$\frac{dy}{dt}=-y+sig(x_1*(1-x_2),\theta,k)$	

	x_2不表达时，则基因 y 表达	$sig(x,\theta,k)=\frac{1}{1+e^{-k(x-\theta)}}$ 其中，θ,k 为元件参数。	
	当基因x_1不表达并且基因x_2表达时，则基因 y 表达	$\frac{dy}{dt}=-y+sig((1-x_1)*x_2,\theta,k)$ $sig(x,\theta,k)=\frac{1}{1+e^{-k(x-\theta)}}$ 其中，θ,k 为元件参数。	
ORN	当基因x_1表达或基因x_2不表达时，则基因 y 表达	$\frac{dy}{dt}=-y+sig(x_1+(1-x_2),\theta,k)$ $sig(x,\theta,k)=\frac{1}{1+e^{-k(x-\theta)}}$ 其中，θ,k 为元件参数。	
	当基因x_1不表达或基因x_2表达时，则基因 y 表达	$\frac{dy}{dt}=-y+sig((1-x_1)+x_2,\theta,k)$ $sig(x,\theta,k)=\frac{1}{1+e^{-k(x-\theta)}}$ 其中，θ,k 为元件参数。	
XOR	当基因x_1和基因x_2都表达，或基因x_1和基因x_2都不表达时，则基因 y 表达	$\frac{dy}{dt}=-y+[1-sig(x_1*(1-x_2),\theta,k)-sig((1-x_1)*x_2,\theta,k)]$ $sig(x,\theta,k)=\frac{1}{1+e^{-k(x-\theta)}}$ 其中，θ,k 为元件参数。	
XNOR	当基因x_1和基因x_2一个表达，但基因x_1和基因x_2不同时表达时，则基因 y 表达	$\frac{dy}{dt}=-y+[1-sig(x_1*(1-x_2),\theta,k)-sig((1-x_1)*x_2,\theta,k)]$ $sig(x,\theta,k)=\frac{1}{1+e^{-k(x-\theta)}}$ 其中，θ,k 为元件参数。	

2.3.3 基于群体聚合形态生成的适应度函数定义

适应度函数引导着种群的进化方向，即适应度函数影响群体聚合形态的形成。例如，对于围捕任务，需要在不碰撞目标的情况下对其进行围捕。为了满足这一任务，设定了群机器人距离目标最近和最远的距离。即d_{min}和d_{max}分别为群体聚合形态到目标的最小和最大距离。此外，另一个重要的任务是群体机器人在受限环境中躲避障碍物。为了满足这一任务，设定了群机器人与障碍物之间的最近距离，即是

d_{min}^{obs}群体聚合形态到障碍物的最小距离。因此，适应度函数设置如下:

$$f_1 = \sum_{i=1}^{N_p}\sum_{j=1}^{N_t}\frac{sig\left(d_{pt}^{ij},d_{max},k_1\right)+sig\left(d_{min},d_{pt}^{ij},k_2\right)}{N_pN_t} + \sum_{i=1}^{N_p}\sum_{k=1}^{N_o}\frac{sig(d_{min}^{obs},d_{po}^{ik},k_3)}{N_pN_o} \quad (2\text{-}2)$$

其中，N_p、 N_t 和N_o 分别为围捕机器人个数、目标个数和障碍物个数。d_{pt}^{ij}为第 i 个围捕机器人到第 j 个目标的距离； d_{po}^{ik}为第 i 个围捕机器人到第 k 个障碍物的距离； k_1、 k_2 和k_3为参数。

为了描述基因调控网络的复杂度，一个重要的指标是基因调控网络中节点的个数。这个指标可以看作是另外一个优化目标，定义如下:

$$f_2 = node(m) \quad (2\text{-}3)$$

其中，m 为 GRN 模型，$node()$为计算 GRN 模型的节点个数。

2.3.4 基于复合多目标基因编程的群体聚合形态自动设计框架

针对现有的群体聚合形态生成大多需要人类设计人员根据经验设计网络结构，难以适应复杂多变环境的局限，提出了一种基于多目标基因编程的群体聚合形态自动设计框架，研究内容的整体框架如图 2-3 所示。该框架分为三部分：a) 群体聚合形态表达；b) 基于多目标基因编程的群体聚合形态优化设计框架；c) 群体聚合形态转换。

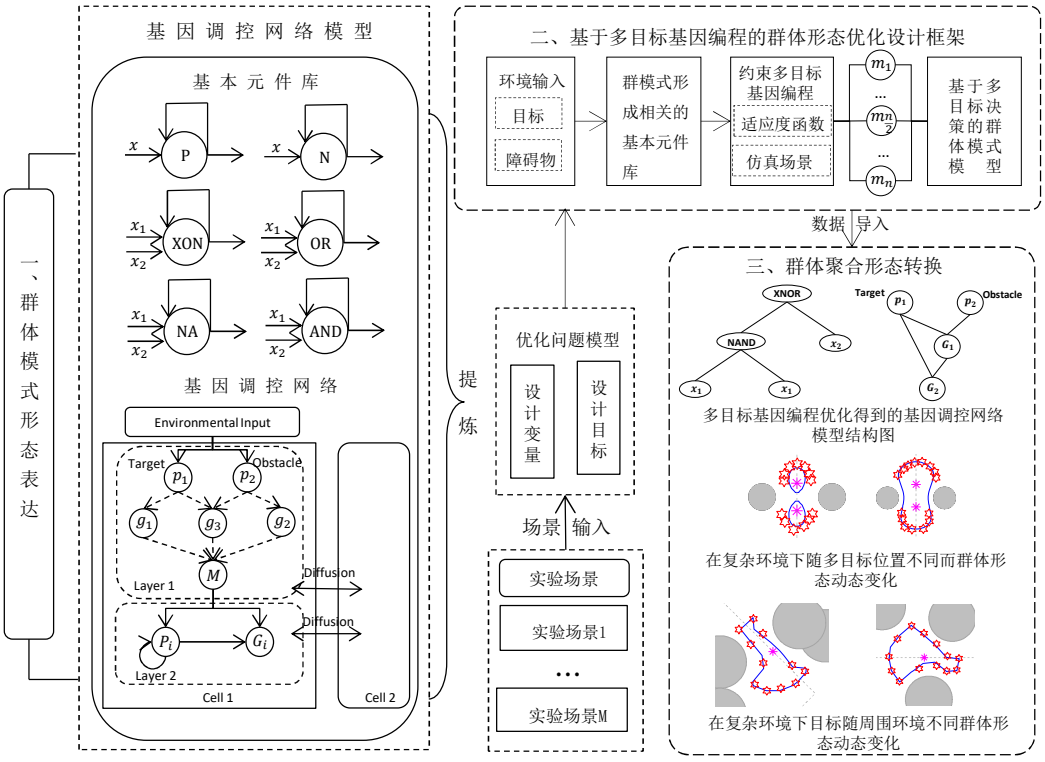


图 2-3 基于多目标基因编程的群体聚合形态自动设计框架

a) 群体聚合形态表达：在现实世界，基因调控网络的设计通常是一个繁琐的试错过程，人为设计的基因调控网络难以适应复杂和动态的环境。需要针对不同的实际问题进行修改和重新测试,直到满足实际实现的要求，这就需要大量的专业知识，试错时间，编码工作。为了解决这个问题，我们采用了模块化自动设计的思想来生成所需的基因调控网络。首先构建基因调控网络模型的基本元件库，然后利用模块化自动设计的思想来自动构建基因调控网络，从而实现基因调控网络对群体聚合形态的统一表达。为此，我们提出了基于多目标基因编程的分层次基因调控网络，框架如图 2-4 所示。

首先，MOGP-GRN 的上层用于群体模式的生成，基因调控网络模型通过基本元件库和环境输入相结合而自动演化出来的。在图 2-4 中，p_1和p_2表示环境输入(即目标和障碍)产生的蛋白浓度，并在空间上扩散。它们将作为基因调控网络上层的输入，激活基因g_1、g_2和g_3。这些激活基因根据环境输入自动组装成所需要的基因调控网络模型。特别是 M 的浓度起着形态发生的作用，在目标周围形成所需的形状，并传递到下层。值得注意的是只有组织机器人探测到目标时上层基因调控网络才被激活，而非有组织的机器人只是跟随邻近的有组织机器人的运动。通过将障碍物作为环境输入，群体聚合形态模式本身随着障碍物的接近而改变，所有机器人都可以同时执行捕获和躲避两种任务。

一旦上层生成群体聚合形态模式，它将作为下层的输入来触发其行为。下层负责控制群体机器人形成该群体聚合形态，并始终保持机器人之间的最小距离。群体机器人的运动控制是通过下层实现，其中下层里的蛋白质 G 和 P 用来表示机器人当前的位置和内部状态。通过p_1、p_2和 G 的扩散，机器人可以共享目标和障碍物的信息，保持机器人之间的最小距离。

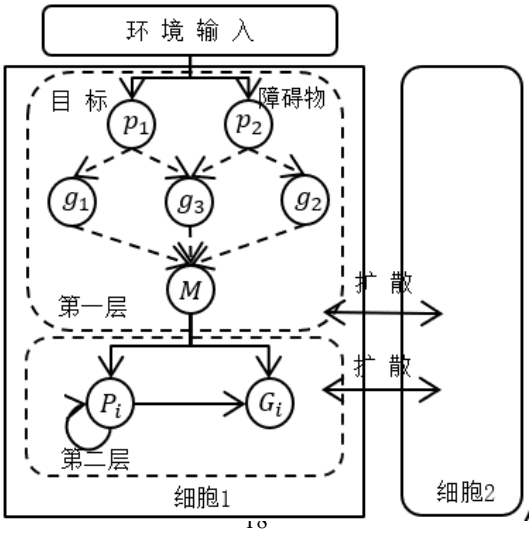


图 2-4 基于多目标基因编程的分层次基因调控网络

b) 基于多目标基因编程的群体聚合形态优化设计框架：在基因调控网络对群体聚合形态的行为进行统一表达的基础上，采用多目标基因编程的方法对基因调控网络进行拓扑结构和参数的优化。由于现有的基因调控网络主要是依赖人的经验进行设计，难以表达复杂动态的群体机器人系统的行为。而采用多目标基因编程的方法对基因调控网络进行优化，可以打破人类对基因调控网络模型拓扑结构的认知局限，实现对基因调控网络的创新设计。为此，我们提出了基于多目标基因编程的群体聚合形态优化设计框架，框架如图 2-5 所示。通过此框架对自动构建基因调控网络的拓扑结构和参数进行同时优化。

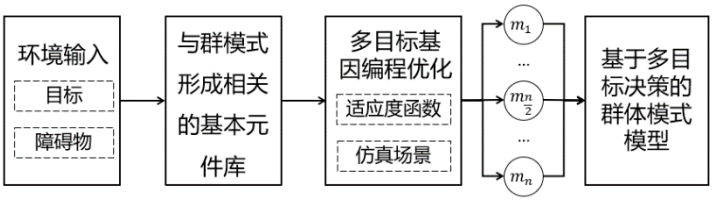


图 2-5 基于多目标基因编程的群体聚合形态优化设计框架

c) 群体聚合形态转换：通过多目标基因编程对基本元件库的模型进行连接形成新的基因调控网络模型，并在网络模型空间与内部参数进行寻优，以获得在多个方面性能更加优秀的、具有创新优化结构的基因调控网络。该基因调控网络可以根据不同任务、环境特征，如背景、障碍、友邻、本体等多种环境因素，进行动态的自适应模式转化。如在受限环境中围捕目标时，根据目标的位置和目标周围的环境信息，群体机器人会动态的转换成最适应当前环境的群体聚合形态。

2.3.5 实验与分析

进行实验以验证所提出的设计自动化框架的有效性。 首先，本章提出的框架生成了一个 GRN 模型(GRN-1)，以包围穿过两个圆形障碍物(OBS-1)的两个目标。然后，本章提出的框架重新获得新的 GRN 模型(GRN-2)，以包围穿过具有挑战性含有障碍的通道(OBS-2)的两个目标。最后，把 GRN-1 和 GRN-2 直接迁移到具有圆形障碍物随机分布的环境(OBS-3)，进一步说明它们的效果。

2.3.5.1 针对 OBS-1 的群体聚合形态生成

为了证明所提出的自动框架的性能，首先在 OBS-1 中比较了所提出的框架和基

于进化的分层基因调控网络(EH-GRN)[83]。值得注意的是，EH-GRN 的参数可以通过进化策略(Evolutionary Strategy, ES)进行优化，其结构是人为定义的，而所提出的框架可以通过遗传编程(GP)生成具有参数和拓扑结构的 GRN 模型。 特别是，群体聚合形态有望完成以下任务：

1). 当两个目标穿过障碍物时，这两个目标必须由群体聚合形态包围。

2). 群体聚合形态应避免障碍物。

3). 群体聚合形态应该能够引导群机器人包围两个目标。

在进化过程中，详细参数列出如下：

1). 对于所提出的框架，将两种基本元件 Positive 和 Negative 的调节参数θ的取值范围介于 0 到 1 之间的实数[81]。其余的基本元件的调节参数θ的取值范围是介于 0 到 2 之间的实数[83]。每个基本元件的θ都被 DE 优化。

2). 对于 EH-GRN，通过协方差矩阵适应进化策略(CMA-ES)[86]优化了每个基本元件的调节参数θ_i $(i = 1,2,\ldots,13)$。根据[83]，将调节参数θ_7，θ_8，θ_9 和 θ_{13}的取值范围介于 0 到 2 的实数，其余参数的取值范围介于 0 到 1 的实数。

3). 对于所提出的框架，用于拓扑结构优化的种群大小为 40，用于参数优化的种群大小为 10。对于 EH-GRN，种群大小设置为 40。

4). 对于所提出的框架中使用的 GP，其 GP 树的最大深度为 4，最小深度为 1。

5). CMOGP 和 DE 的交叉率分别为 1.0 和 0.9。CMOGP 和 DE 的变异率分别为 0.1 和 0.5。

6). 所提出的框架和 EH-GRN 的评估次数均为 4000。

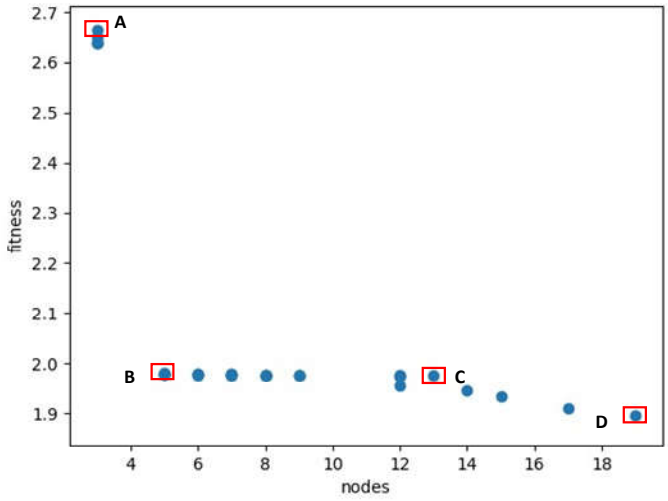
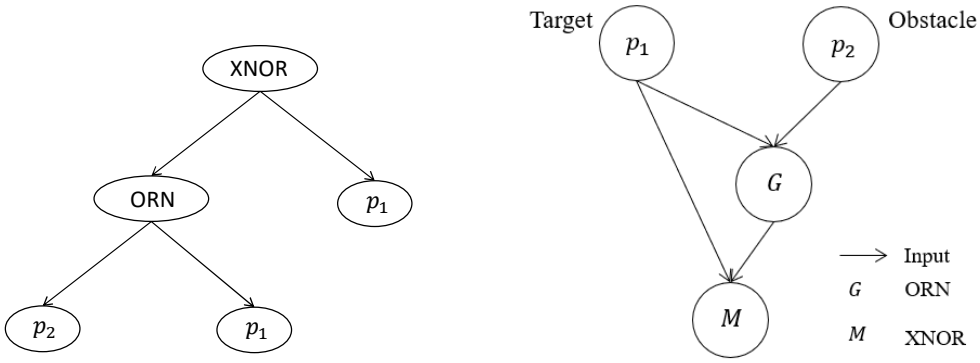


图 2-6 当评价次数达到 4000 时，通过多目标基因编程得到非支配解集。 点 A，点 B，点 C 和点 D 是四个特别选择的。特别地，点 A 和点 D 是两个极限点。点 B 和点 C 是两个拐点。

在图 2-6 中，非支配解集是通过 CMOGP 优化得到的。图 2-6 显示了在点 A 处

的节点数为 3，这在所有非支配解中均节点个数是最少，但其适应度值最大。尽管在所有非支配解中 D 点的适应度值最小，但 D 点的结构最复杂。选择点 B 和 C，因为它们被称为拐点。此外，B 点和 C 点的适应度非常接近，B 点的结构比 C 点简单得多。此外，即使 B 点的结构比 A 点稍微复杂， B 点的性能要比 A 点好得多。因此，在任务中选择了 B 点的 GRN 模型，称为 GRN-1。图 2-7(a)显示了基因型，即点 B 的 GP 树，图 2-7(b)显示了表现型，即点 B 的 GRN 模型，该模型用于群体模式生成层建议的框架。



(a)GRN-1 的基因型(GP 树)　　(b)GRN-1 的表现型(GRN 模型)

图 2-7 GRN-1 的基因型(遗传编程树)和表现型(GRN 模型)。p_1和p_2分别代表目标和障碍物产生的形态梯度。在(a)中，ORN 和 XNOR 是两个基本的网络元件。在(b)中，G 是激活基因[83]，M 代表群体形态形成的梯度空间。

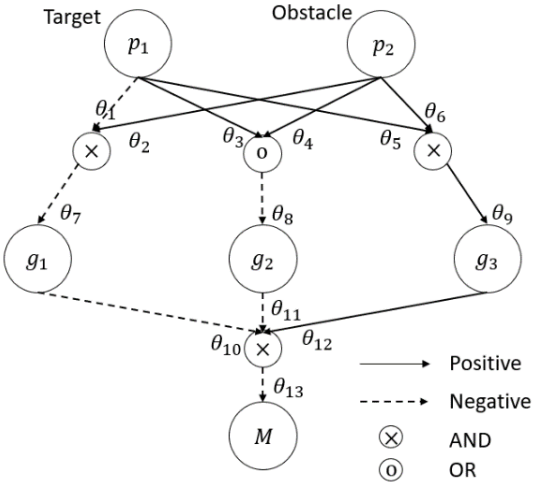


图 2-8 用于群体聚合形态生成的 EH-GRN-1 结构的图。该模型的结构是预先定义的，并且使用 CMA-ES 来优化调节参数θ_i $(i = 1,2,\ldots,13)$。g_1，g_2和g_3是激活基因[83]。M 代表群体形态形成的梯度空间。p_1和p_2分别代表目标和障碍物产生的形态梯度。

对于 GRN-1，每个群机器人均遵循以下动力学方程，以生成定义群机器人模式的形态发生梯度空间：

$$\frac{dG}{dt} = -G + [1 - sig(x_2 + (1 - x_1), \theta_1, k)] \tag{2-4}$$

$$\frac{dM}{dt} = -M + 1 - sig(x_1 * (1 - G), \theta_2, k) - sig((1 - x_1) * G, \theta_2, k) \tag{2-5}$$

其中，θ_1和θ_2分别是 0.5050 和 1.3021。公式(2-4)是元件 ORN 的数学描述，公式(2-5)是元件 XNOR 的数学描述。G 和 M 是形态梯度空间，M 被用于生成群体聚合形态。

为了验证所提出框架的性能，使用了一种最新方法，即 EH-GRN 作为比较。 图 2-8 示出了特定的 EH-GRN 模型，即 EH-GRN-1。在 EH-GRN-1 中，所有调节参数θ_i $(i = 1,2,...,13)$通过 CMA-ES 进行了优化。另外，p_1和p_2可以通过环境输入来调节。g_1，g_2和g_3是激活基因[83]。M 的浓度在目标周围形成所需的群体模式。

对于 EH-GRN-1，每个群机器人都遵循以下动力学方程来生成定义群体聚合形态发生梯度空间：

$$\frac{dy_1}{dt} = -y_1 + sig(p_1, \theta_1, k) \tag{2-6}$$

$$\frac{dy_2}{dt} = -y_2 + 1 - sig(p_2, \theta_2, k) \tag{2-7}$$

$$\frac{dy_3}{dt} = -y_3 + 1 - sig(p_1, \theta_3, k) \tag{2-8}$$

$$\frac{dy_4}{dt} = -y_4 + 1 - sig(p_2, \theta_4, k) \tag{2-9}$$

$$\frac{dy_5}{dt} = -y_5 + 1 - sig(p_1, \theta_5, k) \tag{2-10}$$

$$\frac{dy_6}{dt} = -y_6 + 1 - sig(p_2, \theta_6, k) \tag{2-11}$$

$$\frac{dg_1}{dt} = -g_1 + sig(y_1 * y_2, \theta_7, k) \tag{2-12}$$

$$\frac{dg_2}{dt} = -g_2 + sig(y_3 + y_4, \theta_8, k) \tag{2-13}$$

$$\frac{dg_3}{dt} = -g_3 + sig(y_5 * y_6, \theta_9, k) \tag{2-14}$$

$$\frac{dy_{10}}{dt} = -y_{10} + sig(g_1, \theta_{10}, k) \tag{2-15}$$

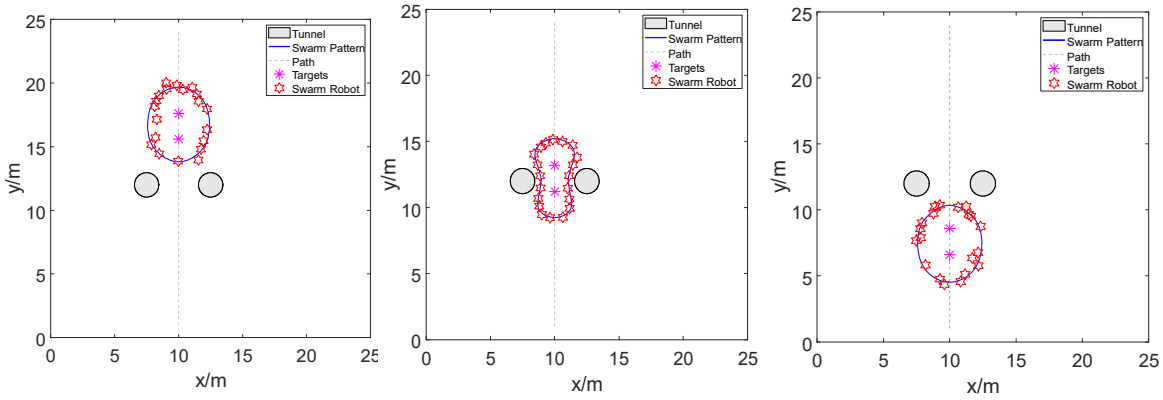
$$\frac{dy}{dt} = -y_{11} + sig(g_2, \theta_{11}, k) \tag{2-16}$$

$$\frac{dy_{12}}{dt} = -y_{12} + 1 - sig(g_3, \theta_{12}, k) \tag{2-17}$$

$$\frac{dM}{dt} = -M + sig(y_{10} * y_{11} * y_{12}, \theta_{13}, k) \tag{2-18}$$

其中，参数值如下：$\theta_1 = 0.7531$、$\theta_2 = 1.0000$、$\theta_3 = 0.1611$、$\theta_4 = 0.8001$、$\theta_5 = 0.0646$、$\theta_6 = 0$、$\theta_7 = 1.6172$、$\theta_8 = 2$、$\theta_9 = 1.4784$、$\theta_{10} = 0.6821$、$\theta_{11} = 0$、$\theta_{12} = 1.0000$ 和$\theta_{13} = 1.1359$。

图 2-9 显示，当两个目标移动通过两个圆形障碍物时，GRN-1 会生成群机聚合形态以包围两个目标。例如，当两个目标靠近两个圆形障碍物时，GRN-1 会生成椭圆形态，如图 2-9(a)中的蓝线所示，以引导群机器人围绕两个目标。当两个目标位于两个圆形障碍物的中间时，如图 2-9(b)所示，群机聚合形态从椭圆形变为近似矩形，这导致群机器人包围两个目标且没有与障碍碰撞。群机聚合形态改变的可能原因是由两个圆形障碍物形成的形态发生剂梯度空间压缩了由两个目标形成的形态发生剂梯度空间。当两个目标都远离两个圆形障碍物时，如图 2-9(c)所示，群机聚合形态从近似矩形模式变回椭圆形模式。



(a) (b) (c)

图 2-9 当两个目标穿过两个障碍物时，由 GRN-1 生成的群机形态包围两个目标。障碍物由两个阴影圆圈表示。蓝色轮廓代表群机形态。六边形代表群体机器人。(a)~(c)显示了两个目标位于不同位置时的几种典型情况，该群机形态引导群机器人包围两个目标。可以看出，群机器人可以成功地跟随并包围两个目标。

图 2-10 显示 EH-GRN-1 在两个目标移动通过两个圆形障碍物时生成群体聚合形态来包围两个目标。例如，当两个目标靠近两个圆形障碍物时，由 EH-GRN-1 生成的群机器人模式不仅包围两个目标，而且还包围两个圆形障碍物，如图 2-10(a)所示。

产生这种群体形态的原因是，由两个圆形障碍物形成的形态梯度空间与由两个目标形成的形态梯度空间融合了。尽管由模式引导的群机器人围绕着两个目标，但是它们并没有均匀地分布在模式上。当两个目标位于两个圆形障碍物的中间时，将生成一个矩形图案，如图 2-10(b)所示。但是，由于无法在模式中很好地分布群机器人，因此它们现在聚集在模式的一侧。当两个目标远离两个圆形障碍物时，生成的图案仍然围绕目标和障碍物。但是，如图 2-10(c)所示，成群的机器人集中在障碍物附近，没有跟随目标。换句话说，群体形态无法引导群体机器人包围目标。

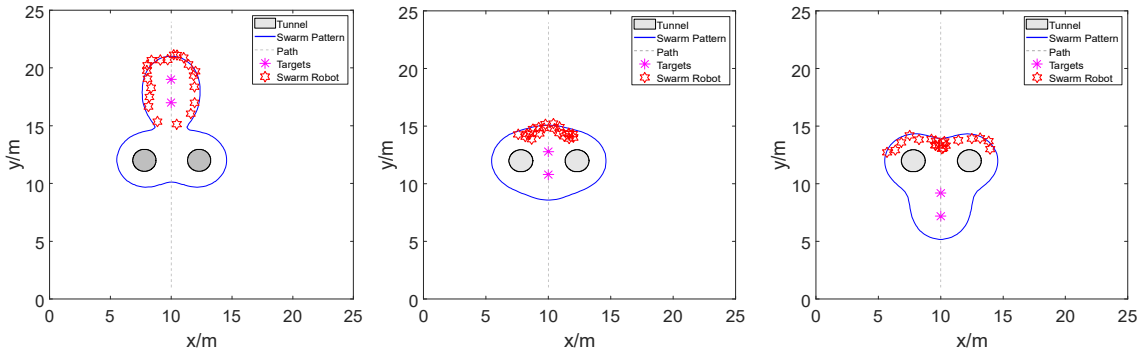


图 2-10 当两个目标移动通过两个障碍物时，由 EH-GRN-1 生成的群机形态将两个目标包围。障碍物由两个阴影圆圈表示。蓝色轮廓代表群体模式。六边形代表群体机器人。(a)~(c)显示了两个目标位于不同位置时的几种典型情况，该群机器人模式导致群机器人环绕两个目标。可以看出，群机器人不能成功地跟随目标并且不能包围它们。

2.3.5.2 针对 OBS-2 的群体聚合形态生成

该框架还经过了 OBS-2(即包含复杂障碍物的通道)的测试。当两个目标通过通道移动时，GRN 模型需要生成动态变化的模式以适应具有挑战性的环境。在 OBS-2 中关于实验结果的描述中，期望使用群体模式与 OBS-1 几乎相同。

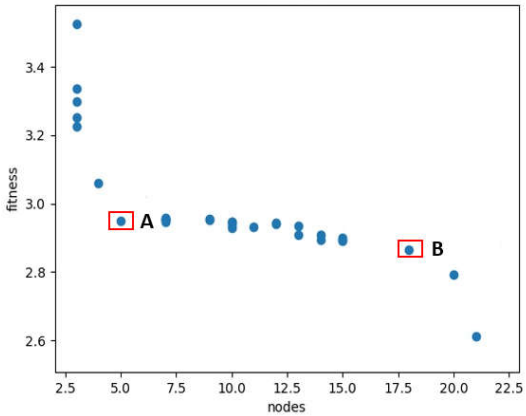
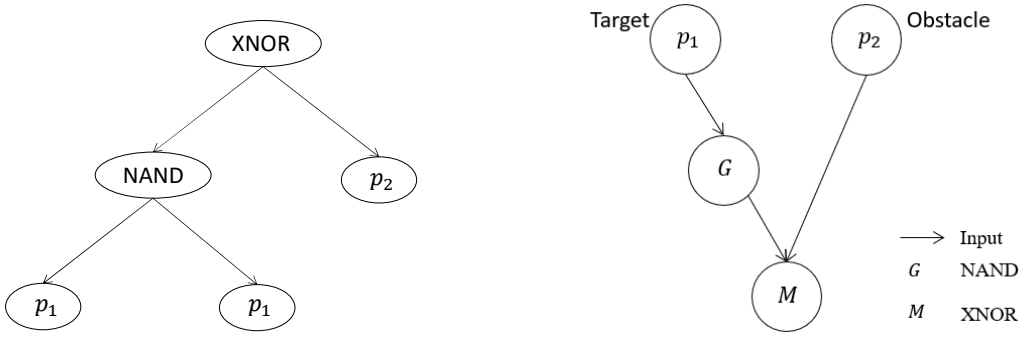


图 2-11 当评价次数达到 4000 时，通过多目标基因编程得到非支配解集。点 A 和点 B 是两个选定的拐点。

在进化过程中，参数与之前的模拟实验相同。在图 2-12 中，非支配解通过

CMOGP 获得的。图 2-12 示出了点 A 和 B 是拐点。尽管 A 点的适应度值与 B 点的适应度没有显着差异，但 A 点的 GRN 模型结构比 B 点更简单。因此，在任务中选择了 A 点的 GRN 模型，称为 GRN -2。图 2-12(a)显示了基因型，即点 A 的遗传编程树，图 2-12(b)显示了表现型，即点 A 的 GRN 模型，该表现型用于群体模式生成层建议的框架。



(a)GRN-2 的基因型(GP 树)　　(b)GRN-2 的表现型(GRN 模型)

图 2-12 GRN-2 的基因型(遗传编程树)和表现型(GRN 模型)。p_1和p_2分别代表目标和障碍物产生的形态梯度。在(a)中， NAND 和 XNOR 是两个基本的网络元件。在(b)中，G 是激活基因[83]，M 代表群体形态形成的梯度空间。

对于 GRN-2，每个群机器人均遵循以下动力学方程，以生成群体形态：

$$\frac{dy_1}{dt} = -y_1 + [1 - sig(x_1 * x_1, \theta_1, k)] \tag{2-19}$$

$$\frac{dM}{dt} = -M + 1 - sig(y_1 * (1 - x_2), \theta_2, k) - sig((1 - y_1) * x_2, \theta_2, k) \tag{2-20}$$

其中，θ_1和θ_2分别是 0.8393 和 0.9256。公式(2-19)是元件 NAND 的数学描述，公式(2-20)是元件 XNOR 的数学描述。y_1和 M 是形态梯度空间，M 被用于生成群体聚合形态。

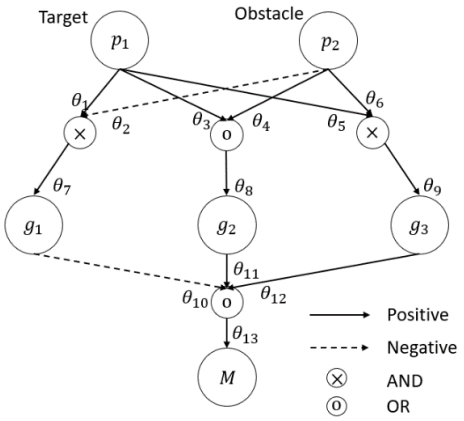


图 2-13 用于群体聚合形态生成的 EH-GRN-2 结构的图。该模型的结构是预先定义的，并且使用 CMA-ES 来优化调节参数θ_i $(i = 1,2,...,13)$。 g_1，g_2和g_3是激活基因[83]。M 代表群体形态形成的梯度空间。p_1和p_2分别代表目标和障碍物产生的形态梯度。

为了验证所提出框架的性能，将 EH-GRN 用作约束环境的比较。 图 2-13 示出了特定的 EH-GRN 模型，即 EH-GRN-2。在 EH-GEN-2 中，所有调节参数θ_i $(i = 1,2,...,13)$通过 CMA-ES 进行了优化。另外，p_1和p_2可以通过环境输入来调节。g_1，g_2和g_3是激活基因。M 的浓度在目标周围形成所需的群体模式。

对于 EH-GRN-2，每个群机器人都遵循以下动力学方程来生成定义群体聚合形态发生梯度空间：

$$\frac{dy_1}{dt} = -y_1 + 1 - sig(p_1, \theta_1, k) \tag{2-21}$$

$$\frac{dy_2}{dt} = -y_2 + 1 - sig(p_2, \theta_2, k) \tag{2-22}$$

$$\frac{dy_3}{dt} = -y_3 + sig(p_1, \theta_3, k) \tag{2-23}$$

$$\frac{dy_4}{dt} = -y_4 + sig(p_2, \theta_4, k) \tag{2-24}$$

$$\frac{dy_5}{dt} = -y_5 + sig(p_1, \theta_5, k) \tag{2-25}$$

$$\frac{dy_6}{dt} = -y_6 + sig(p_2, \theta_6, k) \tag{2-26}$$

$$\frac{dg_1}{dt} = -g_1 + sig(y_1 * y_2, \theta_7, k) \tag{2-27}$$

$$\frac{dg_2}{dt} = -g_2 + sig(y_3 + y_4, \theta_8, k) \tag{2-28}$$

$$\frac{dg_3}{dt} = -g_3 + sig(y_5 * y_6, \theta_9, k) \tag{2-29}$$

$$\frac{dy_{10}}{dt} = -y_{10} + 1 - sig(g_1, \theta_{10}, k) \tag{2-30}$$

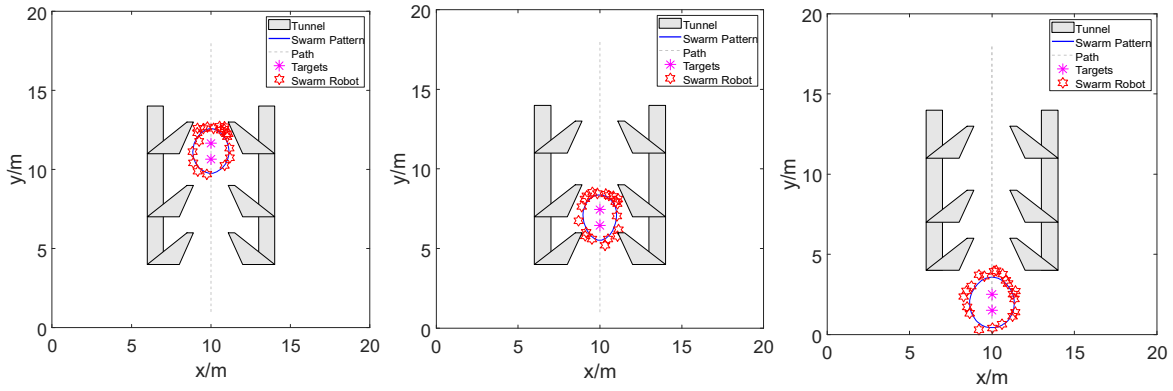
$$\frac{dy}{dt} = -y_{11} + sig(g_2, \theta_{11}, k) \tag{2-31}$$

$$\frac{dy_{12}}{dt} = -y_{12} + sig(g_3, \theta_{12}, k) \tag{2-32}$$

$$\frac{dM}{dt} = -M + sig(y_{10} + y_{11} + y_{12}, \theta_{13}, k) \tag{2-33}$$

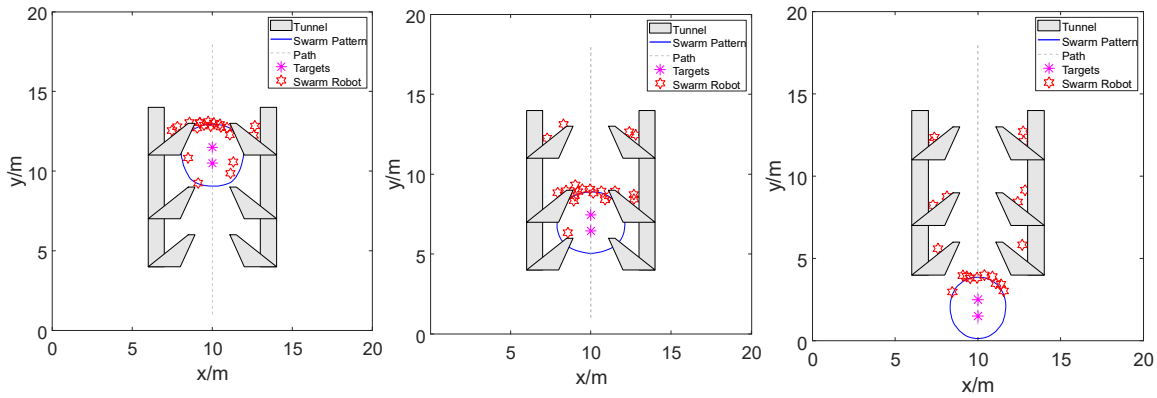
其中，参数值如下：$\theta_1 = 0.5553$、$\theta_2 = 0$、$\theta_3 = 1$、$\theta_4 = 1$、$\theta_5 = 0.996$、$\theta_6 = 1$、$\theta_7 = 1.2097$、$\theta_8 = 0.0662$、$\theta_9 = 2$、$\theta_{10} = 0$、$\theta_{11} = 1$、$\theta_{12} = 1$ 和$\theta_{13} = 0.2339$。

图 2-14 和图 2-15 显示，当目标从 OBS-2(即包含障碍物的通道)移动时，GRN-2 和 EH-GRN-2 会生成群机器人模式以包围两个目标。从图 2-14(a)-(c)中我们可以看到，当目标移动通过通道时，GRN-2 生成的模式可以很好地环绕两个目标。此外，受该形态引导的群机器人非常均匀地分布在模式周围，从而始终将目标紧密围绕。相反，由 EH-GRN-2 生成的图案与障碍物一起包围两个目标，即障碍物帮助围绕目标作为图案的一部分，如图 2-15(a)-(c)所示。结果，与 GRN-2 生成的群体聚合形态相比，该形态给群体机器人带来了更大的困难。可以看出，群机器人在该形态下周围分布不均，有些机器人甚至被突出部分阻塞。



(a) (b) (c)

图 2-14 当两个目标穿过 OBS-2 时，由 GRN-2 生成的群机器人模式将两个目标包围。障碍物由两个阴影圆圈表示。蓝色轮廓代表群体模式。六边形代表群体机器人。(a)~(c)显示了两个目标位于不同位置时的几种典型情况，该群机器人模式导致群机器人环绕两个目标。可以看出，群机器人可以成功地跟随并包围两个目标。

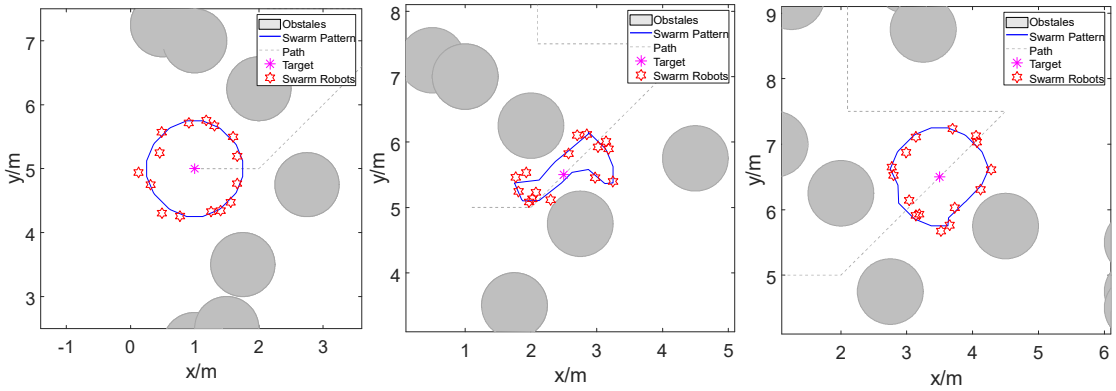


(a) (b) (c)

图 2-15 当两个目标穿过两个障碍物时，由 EH-GRN-2 生成的群机器人模式将两个目标包围。障碍物由两个阴影圆圈表示。蓝色轮廓代表群体模式。 六边形代表群体机器人。(a)~(c)显示了两个目标位于不同位置时的几种典型情况，该群机器人模式导致群机器人环绕两个目标。可以看出，群机器人不能成功地跟随目标并且不能包围它们。

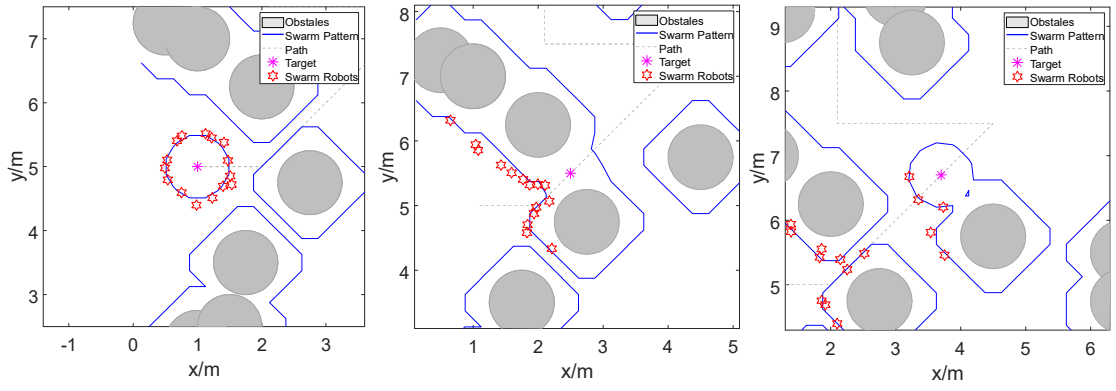
2.3.5.3 将为 OBS-1 和 OBS-2 获得的模型迁移到 OBS-3

为了证明所提出的设计自动化框架生成 GRN 模型具有可迁移性，将 GRN-1 和 GRN-2 迁移到障碍物随机分布的新应用场景中。在 OBS-3 中关于实验结果的描述中，期望使用群体模式与 OBS-1 几乎相同。



(a) (b) (c)

图 2-16 当目标移动经过随机分布有大量圆形障碍物的 OBS-3 时，由 GRN-1 生成的群机器人模式将目标包围。障碍物由大量阴影圆圈表示。蓝色轮廓代表群体模式。六边形代表群体机器人。(a)~(c)显示了三种典型的场景：当目标在 OBS-3 的不同位置移动时，群体模式引导群体机器人环绕目标。可以看出，在所有情况下，成群的机器人都紧紧围绕目标。

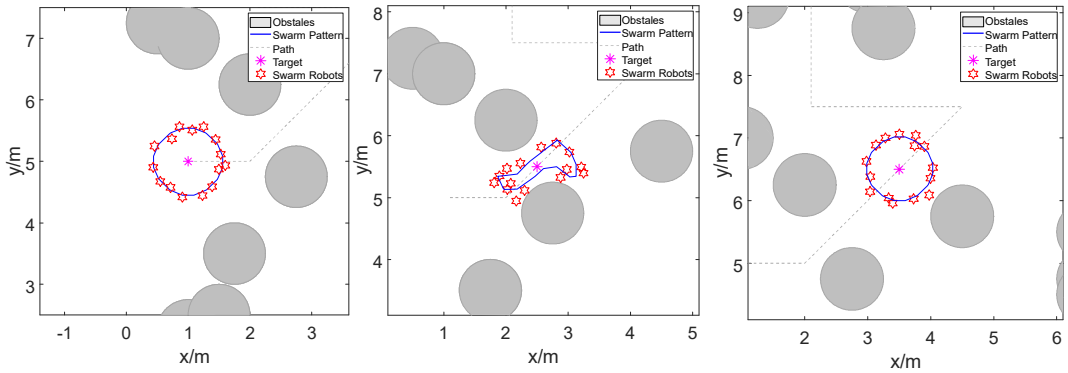


(a) (b) (c)

图 2-17 当目标移动通过随机分布有大量圆形障碍物的 OBS-3 时，由 EH-GRN-1 生成的群机模式将目标包围。障碍物由大量阴影圆圈表示。蓝色轮廓代表群体模式。六边形代表群体机器人。(a)~(c)显示了三种典型场景，它们说明了群体模式如何导致群体机器人环绕 OBS-3 中的目标。

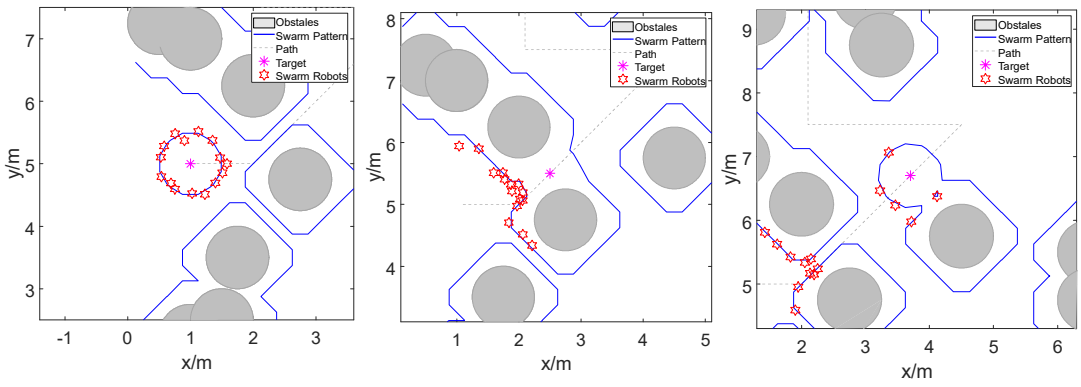
可以看出，图案与障碍物的边界合并了，并且群体机器人在很大的图案轮廓中分布不均匀。结果，当目标移动到(c)的位置时，大量的机器人被该模式遗忘，无法包围目标。

作为比较，还在 OBS-3 中测试了 EH-GRN-1 和 EH-GRN-2。图 2-16 和图 2-18 分别显示了由 GRN-1 和 GRN-2 生成的群体模式围绕环境中的目标。我们可以清楚地观察到，由 GRN-1 和 GRN-2 生成的群体模式不仅可以不断调整模式以适应目标周围的约束环境，而且可以引导群机器人围绕目标。图 2-17 和图 2-19 分别显示了由 EH-GRN-1 和 EH-GRN-2 生成的群体模式围绕环境中的目标。我们可以清楚地观察到，EH-GRN-1 和 EH-GRN-2 标记了约束环境中所有障碍物的位置，然后由 EH-GRN-1 和 EH-GRN-2 生成的群体模式与障碍物合并。目标包围目标。这些形式的群机器人模式不能导致所有群机器人围绕目标逃脱的目标。一个可能的原因是，当群体模式与目标周围的障碍物合并时，会生成一个新的群体模式，该模式同时包围目标和障碍物，如图 2-17(b)和图 2-19(b)所示。新的群体模式不能导致所有群机器人包围目标。



(a) (b) (c)

图 2-18 当目标移动经过随机分布有大量圆形障碍物的 OBS-3 时，由 GRN-2 生成的群机器人模式将目标包围。障碍物由大量阴影圆圈表示。蓝色轮廓代表群体模式。六边形代表群体机器人。(a)~(c)显示了三种典型的场景：当目标在 OBS-3 的不同位置移动时，群体模式引导群体机器人环绕目标。可以看出，在所有情况下，成群的机器人都紧紧围绕目标。



(a) (b) (c)

图 2-19 当目标移动通过随机分布有大量圆形障碍物的 OBS-3 时，由 EH-GRN-1 生成的群机模式将目标包围。障碍物由大量阴影圆圈表示。蓝色轮廓代表群体模式。六边形代表群体机器人。(a)~(c)显示了三种典型场景，它们说明了群体模式如何导致群体机器人环绕 OBS-3 中的目标。可以看出，图案与障碍物的边界合并了，并且群体机器人在很大的图案轮廓中分布不均匀。结果，当目标移动到(c)的位置时，大量的机器人被该模式遗忘，无法包围目标。

2.4 群体聚合形态自动形成的自动优化设计系统

为了方便研究人员自主得到群体聚合形态模型，我们开展了群体聚合形态形成系统的探索工作。该系统是一款群体聚合形态自动形成的自动优化设计系统。该系统结合了基因调控网络、智能优化算法和数据挖掘，可实现群体聚合形态自动优化设计。借助该系统，可以省去传统设计过程中的基因调控网络草图设计、程序编写、参数优化等一系列重复性的工作，从而大大减少设计时间，有效提升设计效率，快速得到满足用户需求的优化设计结果。

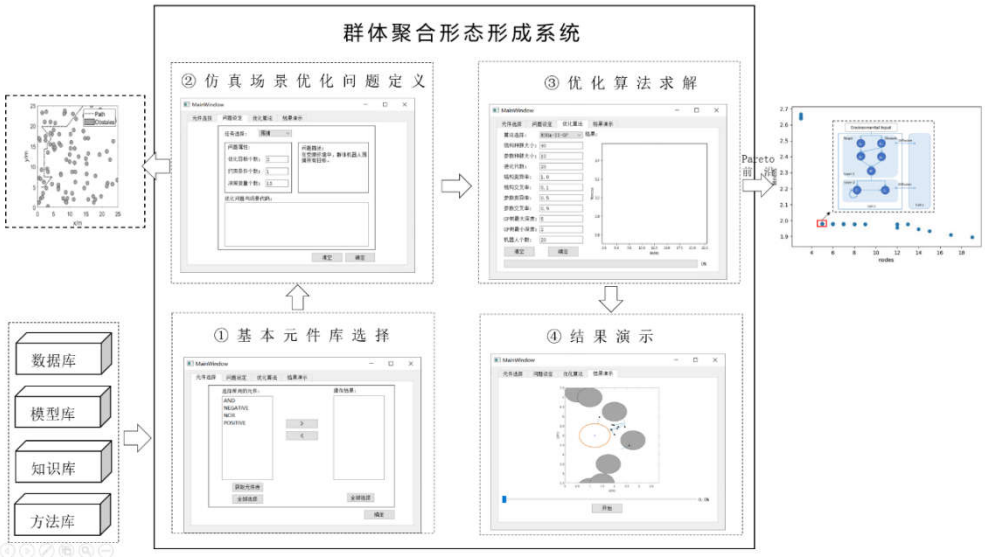


图 2-20 群体聚合形态形成系统的示意图

图 2-20 示该系统界面示意图。该系统分为 4 部分：(1)选择基本元件库：根据用户的偏好，用户可以从基本元件库中选择所需的基本元件。然后，借助基因编程方法，可以自动设计基因调控网络模型。(2)仿真场景优化问题定义：用户可以根据实际的需求，通过脚本的形式自定义仿真场景、优化目标(包括多个冲突的目标)，设计变量和设计约束。(3)优化算法求解：根据需求用户可以选择相应的算法。通过设置相关参数，该平台可以自动进行运算。(4)结果演示：将优化得到的基因调控网络模型用于仿真场景。

该设计平台框架其实可以看作成一大类设计框架。各种主流方法采用的手段都是用一种模块化的建模语言来描述一个预先设定的模块库，然后用基因编程(或者类似于基因编程的可以自动在开放式设计空间中进行拓扑结构和参数并行搜索优化)的方法进行自动寻优。

2.5 本章小结

GRN 模型的设计通常是一个繁琐且反复试验的过程，主要取决于人类专家的经验和直觉。而且，专门为特定应用场景设计的 GRN 模型几乎不能保证适用于新场景。为了解决这些问题，本文提出了一种自动设计框架，该模型可以使用 CMOGP 生成群机器人模式，CMOGP 是一种复合多目标遗传程序，可以通过重新配置一组精心定义的基本 GRN 网络主题来合成 GRN 模型。实验结果表明，使用提出的设计自动化框架演化的 GRN 模型可以生成模式，以指导群体机器人在各种挑战性场景中成功包围目标。更重要的是，当将演化后的模型转移到该模型之前从未进行过训练的新场景时，它们仍然可以有效地生成模式，该模式可以指导成群的机器人成功包围目标。

尽管通过仿真实验已经证明了所提出的设计自动化框架的优势，但仍需要进行更多的物理实验以进一步验证其有效性，这是我们研究计划的下一步。多目标围捕问题仍需进一步研究。例如，仅考虑一些典型且固定的环境约束，它们不能代表现实世界中的所有障碍。作为未来的工作，我们将在此提议的框架中添加一些机制，以在未知环境中实现群体聚合形态的生成，并在实际应用中将建议的 GRN 应用于 E-puck 教育机器人。

第 3 章　约束处理机制国内外研究现状

一般情况下，约束优化问题由于约束的种类不一样(如等式约束、不等式约束等)使得约束优化问题具有不同的特性。尤其是，当优化问题具有超多目标(两个目标、三个目标或者三个以上目标)和决策变量个数很多时，使得约束问题更加复杂和求解。因此，如何设计约束处理机制起到至关重要的目的。本章将会对约束处理机制作发展做简单描述。

3.1 带约束的多目标优化问题

尽管存在一些现实世界中最优化问题的复杂性，但如今它们很容易通过自然启发式元启发式方法来解决，例如进化算法[87]和群体智能算法(Swarm Intelligence Algorithms，SIA)[88]。在本章中，这两种类型的方法将统称为自然启发式算法。

Michalewicz[89]描述了在现实世界中优化问题都含有一些困难源。 除了巨大的搜索空间，目标函数中的噪声以及建模过程的复杂性之外，还指出了约束的存在。约束可能会导致搜索将焦点从优化转移到寻找一个可行的(即有效的)解上。另一方面，在刚开始设计 NIA 时，NIA 旨在处理不受约束的搜索空间[90][91]。为了使 NIA 考虑约束，需向 NIA 添加约束处理技术，目的是引导搜索到具有可行解的区域。

一个 CMOPs 可以描述为以下形式：

$$\begin{aligned} &\text{minimize } \vec{F}(\vec{x}) = (f_1(\vec{x}),f_2(\vec{x}),...,f_m(\vec{x})) \\ &\text{subject to } g_i(\vec{x}) \le 0,\ i = 1,...,p; \\ &h_j(\vec{x}) = 0,\ j = 1,...,q. \end{aligned} \tag{3-1}$$

其中，$\vec{F}(\vec{x})$即为目标函数，而$\vec{x} = (x_1,x_2,...,x_D \in R)$就是决策变量。$R = \prod_{k=1}^{D}[L_k, U_k]$代表 D 维决策空间，U_i、L_i分别代表决策向量上第 i 位的取值上下界。最优化问题的约束一般由等式约束和不等式约束组成，公式(3-1)中$g_i(\vec{x}) \le 0$是第 i 个不等式约束，而$h_j(\vec{x}) = 0$是第 j 个等式约束。p 和 q 分别是不等式约束和等式约束个数。

个体$\vec{x}$的总体约束违反程度为$\phi(\vec{x})$，定义如下：

$$\phi(\vec{x}) = \sum_{i=1}^{p} G_i(\vec{x}) + \sum_{j=1}^{q} H_j(\vec{x})$$

(3-2)

对于个体$\vec{x}$，如果$\phi(\vec{x})=0$，则$\vec{x}$被称为可行解。如果$\phi(\vec{x})>0$，则$\vec{x}$被称为不可行解。

3.2 基于进化算法的约束处理机制研究现状

在约束多目标进化算法中的一个关键问题是如何在最小化目标和满足约束条件之间保持平衡。事实上，进化计算中的大多数约束处理机制都试图实现这一平衡。例如，惩罚函数方法[92]采用惩罚因子λ，以保持最小化目标和满足约束条件之间的平衡。此方法是通过将每个个体总体约束违反程度乘以预定义的惩罚因子λ，从把CMOPs 转换为 MOPs。根据λ值的变化可以将惩罚函数法分为以下几类：1)当$\lambda=\infty$时，此方法被称为死亡惩罚法(Death Penalty Approach)[93]，这意味着不可行解是完全不可接受的。2)如果λ是搜索过程中的固定值，则称为静态惩罚方法(Static Penalty Approach)[94]。3)如果λ在搜索过程中发生变化，则称为动态惩罚方法(Dynamic Penalty Approach)[95]。4)在λ根据搜索过程中收集到的信息变化的情况下，它被称为自适应惩罚方法(Adaptive Penalty Approach)[96]-[99]。

为了避免调整惩罚因子，一些研究者通过比较目标函数和约束条件之间关系，建立另一类约束处理机制，如约束支配原则(Constraint Dominance Principle， CDP)[21]、Epsilon 约束处理方法(Epsilon Constraint-handling，EC) [100]、随机排序方法(Stochastic Ranking，SR) [101]等。虽然 CDP 是一种常用的约束处理方法(因为它简单而且没有额外的参数)，但它不适用于求解可行域非常小且狭窄的带约束多目标优化问题[34]。这是因为在使用 CDP 进化很多代后，工作种群中的大多数解都是不可行的。此外，工作种群的多样性很难得到很好的维持，因为选择解时只是根据 CDP 的第二条规则(基于违反约束的情况)。

为了解决可行域非常小且狭窄的带约束多目标优化问题，Epsilon 约束处理(EC)[100]方法被提出来。除了在放缩约束上不一样外，它的基本思想类似于 CDP。在 EC 中，约束的松弛水平是由ε的值来控制，这有助于在大多数解是不可行时，保持工作种群的多样性。更具体地说，如果解的总体约束违反程度小于ε时，则此解被认为是可行的。在 EC 中，约束的松弛水平ε是一个关键参数。在ε=0 的情况下，EC 是和 CDP 一样。虽然 EC 可以用来解决具有小可行区域的 CMOP，但适当地控制ε的值并不是一件小事。

CDP[21]和 EC[100]都首先比较了约束条件，然后比较了目标函数值。在比较顺序上，SR[101]不同于 CDP 和 EC。对于任意两种解，如果随机数小于p_f，则说明选择

非支配的解更好，即比较目标函数值。反之，如果随机数大于p_f，则优先比较解的约束违反程度，然后是比较目标函数值的大小。值得注意的是，如果$p_f = 0$时，SR相当于 CDP。

为了在最小化目标和满足约束条件之间保持良好的平衡，一些研究人员将几种约束机制结合起来，可以进一步分为两种理论，包括在不同的进化阶段或不同的子问题中采用不同的约束处理机制。例如，自适应权衡模型(Adaptive Trade-off Model，ATM)[102]在不同的进化阶段使用了两个不同的约束机制，包括多目标方法和自适应惩罚函数。约束处理方法(Ensemble of Constraint-handling Methods，ECHM)[103]集合三种不同的约束处理技术，包括 Epsilon 约束处理(Epsilon Constraint-handling，EC)[100]、自适应惩罚函数(Self-adaptive Penalty Functions，SP)[99]和可行解(Feasible Solutions，SF)[104]的优越性。在 ECHM 中，三种约束处理机制分别生成三种群，每个子种群使用不同的约束处理方法。

第 4 章　基于 M2M 处理机制的 Push 和 Pull 的搜索框架

在处理约束多目标优化问题(Constrained Multi-objective Optimization Problems，CMOPs)时，多目标进化算法(Multi-objective Evolutionary Algorithms，MOEAs)的一个关键问题是平衡工作种群的收敛性和多样性。然而，大多数最先进的 MOEAs 在平衡它们方面表现不佳，并可能导致工作种群集中在部分 Pareto 前沿，导致在保持多样性和实现收敛之间严重的不平衡搜索。本章提出了一种将多目标对多目标(Multi-objective to Multi-objective，M2M)分解方法[105]与推拉搜索(Push and Pull Search，PPS)框架[106]相结合的方法，即 PPS-M2M。更具体地说，该算法将一个 CMOP 分解成一组简单的 CMOPs。每个简单的 CMOP 对应于一个工作子种群，并以协作的方式解决。在处理约束时，每个子种群都遵循 PPS 搜索准则，使每个工作子种群跨越不可行区。为了评价该算法的性能，将其与 CM2M[107]、MOEA/D-Epsilon[100]、MOEA/D-SR[108]、MOEA/D-CDP[108]、C-MOEA/D[109]、NSGA-II-CDP[21]、CM2M2[110]等 9 种算法在一组 CMOPs 测试集上进行了比较。实验结果表明，该算法明显优于其他 9 种算法。此外，本章还提出了一组约束不平衡多目标优化问题(Constrained and Imbalanced Multi-objective Optimization Problems，CIMOPs)来比较 PPS-M2M 和 PPS-MOEA/D[]。实验结果表明，所提出的 PPS-M2M 算法在 CIMOPs 上的性能优于 PPS-MOEA/D 算法。

4.1 约束多目标进化算法 PPS-M2M

本节将具体描述 PPS-M2M 算法。PPS-M2M 将 M2M 分解方法与 PPS 框架结合在一个非支配的排序框架中来解决 CMOPs。

4.1.1 M2M 分解方法

在初始阶段，PPS-M2M 使用分解策略将 CMOP 分解为一组子问题，这些问题以协作的方式与对应于子种群的每个子问题一起解决。为了简单起见，我们假设 CMOP 的所有的目标函数$f_1(x)$，$f_2(x)$，...，$f_m(x)$是非负的。否则，目标函数$f_i(x)$被$f_i(x)-\overline{f}_i$替换，其中$\overline{f}_i$是到目前为止找到的目标函数$f_i(x)$的最小值[105]。

假设将目标函数空间划分为 K 个子区域，则 K 个方向向量$\boldsymbol{v_1}$，$\boldsymbol{v_2}$，…，$\boldsymbol{v_K}$均匀分布在单位球的第一象限上，其中$\boldsymbol{v_K}$是第 K 个子区域的中心向量。然后将目标函数空间划分为 K 个不相邻的子区域Ω_1，Ω_2，…，Ω_K，其中第 k 个(k= 1，…，K)子区域可以通过公式(4-1)获得。通过这样的分解方法，多目标优化问题被分解为 K 个简单的 CMOP。该过程在算法 4-1 中详细被描述。作为一个例子，当目标数 $m = 2$ 和子区数 $K = 6$ 时，种群分解如图 4-1 所示，其中$\boldsymbol{v_1}$，$\boldsymbol{v_2}$，…，$\boldsymbol{v_6}$是六个均匀分布的方向向量。

$$\begin{cases} minimize \quad F(x) = (f_1(x),\ f_2(x),\ \ldots,\ f_m(x)) \\ subject\ to \quad x \in \prod_{i=1}^{n} [a_i,b_i] \\ F(x) \in \Omega_k = \{\boldsymbol{u} \in \boldsymbol{R}_+^m | \langle \boldsymbol{u},\boldsymbol{v^k} \rangle \le \langle \boldsymbol{u},\boldsymbol{v^j} \rangle \ for\ any\ j = 1,\ldots,K\} \end{cases} \tag{4-1}$$

其中，$\langle \boldsymbol{u},\boldsymbol{v^j} \rangle$为 $\boldsymbol{u}$ 与$\boldsymbol{v^j}$之间的锐角。

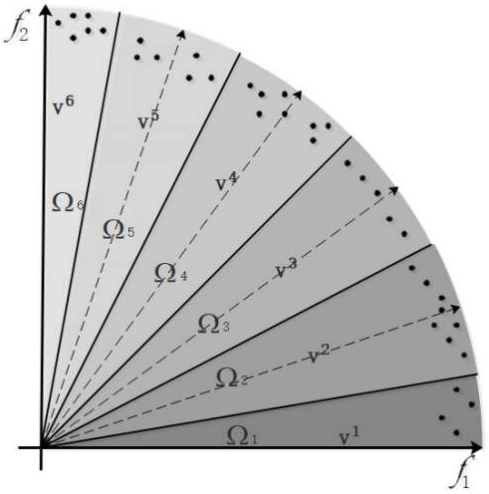


图 4-1 种群分解方式

算法 4-1：Allocation of Individuals to Sub-populations

1 **Function** $result = AllocationSubPop(Q,K)$

2 **for** $i \leftarrow 1$ **to** K **do**

3 Initialize M_k as the solutions in Q whose F-values are in Ω_k according to Eq. (4-1); // Q: a set of individual solutions and their F-values.

4 **end**

5 return $M_1,\ldots,M_k$;

6 **end**

4.1.2 PPS 框架

将 PPS 框架[106]的搜索过程分为两个主要的搜索阶段：推送(Push)搜索阶段和拉索(Pull)搜索阶段。在推送搜索阶段，每个子种群使用一个无约束 NSGA-II 在不考虑任何约束的情况下寻找非支配的解。利用无约束 NSGA-II 求解简单的多目标优化问题时，用两个属性[21]，包括非支配排序i_{rank}和拥挤距离$i_{distance}$。因为个体有两个属性，当一个个体 i 与另一个个体 j 相比较时，如果满足以下两个要求，则个体 i 可以进入后代种群：

1. 如果一个个体 i 支配另一个个体 j，即$i_{rank} < j_{rank}$；
2. 如果个体 i 与另一个个体 j 彼此之间互不支配，那么选择拥有大的拥挤距离的个体，即$i_{rank} = j_{rank}$，$i_{distance} > j_{distance}$。

换句话说，当两个个体属于不同的非支配等级时，等级较低(较优)的个体被选择;

当两个个体具有相同的非支配等级时，选择拥挤距离较大的个体。

在推送搜索阶段，在不考虑任何约束的情况下对约束优化问题进行优化。算法 4-2 给出了推送搜索的伪代码。在第 2 行中，对子种群P_k进行非优势排序，在第 4-7 行中，一些解被选择到$P_k^{'}$中，直到$P_k^{'}$中的解的数量等于S_k。在第 8-10 行中，选择$S_k - |P_k^{'}|$的解到F_i中。在第 11 行，通过设置$P_k = P_k^{'}$来更新子种群P_k。

算法 4-2：Push Subproblems

1 **Function** $result = PushSubproblems(P_k, S_k)$

2 F = nondominated $-$ sort(P_k), $F = (F_1, F_2, ...)$;

3 $P_k^{'} = \emptyset$ and $i = 1$;

4 **While** ($|P_k^{'}| + |F_i| \leq S_k$)

5 $P_k^{'} = P_k^{'} \cup F_i$;

6 $i = i + 1$;

7 **end**

8 calculate crowding-distance in F_i;

9 sort solutions in F_i by crowding-distance in a descending order;

10 $P_k^{'} = P_k^{'} \cup F_i[1:(S_k - |P_k^{'}|)]$;

11 $P_k = P_k^{'}$;

12 **end**

等式(4-2)中描述了从推搜索阶段切换到拉搜索阶段的控制搜索过程机制。在进化过程刚开始时，r_k的值初始化为 1.0，以确保在推阶段进行彻底搜索。r_k的值由公式(4-2)更新。当r_k的值小于或等于预设阈值ε时，搜索行为将更改为拉搜索阶段。

$$r_k \equiv \max\{r_{zk}, r_{nk}\} \leq \varepsilon \tag{4-2}$$

其中，ε为用户定义的预设阈值。在本章中，我们设置ε为10^{-3}。在过去的 l 代中，根据公式(4-3)可以计算理想点的变化率r_{zk}。此外，根据公式(4-4) 可以计算最差点的变化率r_{nk}。

$$r_{zk} = \max_{i=1,\ldots,m}\left\{\frac{|z_i^k - z_i^{k-l}|}{\max\{|z_i^{k-l}|, \Delta\}}\right\} \tag{4-3}$$

$$r_{nk} = \max_{i=1,\ldots,m}\left\{\frac{|n_i^k - n_i^{k-l}|}{\max\{|n_i^{k-l}|, \Delta\}}\right\} \tag{4-4}$$

其中，$z^k = (z^1,\ldots,z_m^k)$和$n^k = (n^1,\ldots,n_m^k)$分别为第 k 代中的理想点和最差点。$z_i^{k-l} = (z_1^{k-l},\ldots,z_m^{k-l})$和$n_i^{k-l} = (n_1^{k-l},\ldots,n_m^{k-l})$分别为第 $k-l$ 代中的理想点和最差点。r_{zk}和r_{nk}取值范围都为[0,1]。Δ为一个非常小的正整数，其作用是保证公式(4-3)和(4-4)的分母不等于 0。在本章中，Δ被设置为10^{-6}。当r_k小于ε时，推搜索阶段完成，拉搜索阶段准备开始。

在拉搜索阶段，我们需要防止种群陷入局部最优，平衡可行区域和不可行区域之间的进化搜索。为了实现这些目标，我们使用改进的 Epsilon 约束处理机制[104]来平衡拉搜索阶段的目标最小化和约束，具体定义如下：

$$\epsilon(k) = \begin{cases} \phi_\theta & if\ k = 0 \\ (1-\tau)\epsilon(k-1) & if\ rf_k < \alpha \\ \epsilon(0)(1-\frac{k}{T_c})^{cp} & if\ rf_k \geq \alpha \\ 0 & otherwise \end{cases} \tag{4-5}$$

其中，$\epsilon(k)$为ϵ函数的值。ϕ_θ为初始总体中排名靠前的θ个个体的总体约束违反程度。rf_k为是可行解在第 k 代的比例。在$rf_k < \alpha$的情况下，当约束的松弛度减小时，$\tau \in [0,1]$控制速度。α控制可行区域和不可行区域之间的搜索偏好。在$rf_k \geq \alpha$的情况下，cp 被用来控制约束松弛的减小间隔。当计数器 k 达到T_c时，$\epsilon(k)$停止更新。$\epsilon(0)$设置为推搜索阶段结束时的最大约束违反程度。在$rf_k < \alpha$的情况下，公式(4-5)设置

$\epsilon(k)$的以指数速度的递减，与[100]中的ε设定相比，它有可能更有效地找到可行的解。在$rf_k \geq \alpha$的情况下，公式(4-5)与[100]中采用的相同的ε设置。在拉搜索阶段，使用算法 4-3 来描述该约束处理机制如何选择一个新的个体。

算法 4-3：Pull Subproblems

1 **Function** $result = PullSubproblems(M_k, gen, T_{max})$
2 // T_{max}: the maximum generation.
3 $result = false$;
4 **if** $gen \leq 0.9T_{max}$ **then**
5 **for** $i \leftarrow 1$ **to** K **do**
6 An improved epsilon constraint-handing mechanism is used to search for non-dominated and feasible solutions in M_k;
7 **end**
8 **else**
9 Merge $M_1,...,M_k$ into a single population M_s;
10 An improved epsilon constraint-handing mechanism is employed on M_s to search non-dominated and feasible solutions;
11 **end**
12 **return** $result$
13 **end**

4.1.3 PPS-M2M 算法流程

算法 4-4 中介绍了 PPS-M2M 的伪代码。在算法 4-4 中，第 1~3 行是算法初始化阶段。具体如下，在第 2 行里，一个 CMOP 的种群被划分为 K 子种群，每个子种群的个体数等于 S。在第 3 行，理想点和最低点之间的最大变化率r_k被初始化为 1.0，搜索阶段的标志被设置为“$PushStage = true$”。该算法从第 4 行反复运行到第 38 行，直到满足终止条件。第 5~13 行描述了每个子种群生成新解的过程。在第 6~10 行是具体产生新解的过程。在第 12~13 行，解集 Q 根据方程(4-1)分配给每个子种群。通过第 15 行计算了在 l 代期间r_k的理想点和最低点之间的最大变化率。参数$\epsilon(k)$

在第 17-26 行更新。第 27~37 行说明了每个分组如何更新的。如果子种群个数M_k小于 S，则 $S-|M_k|$个解从 Q 中随机选择并添加到M_k中。如果子种群个数M_k大于 S，则使用 PPS 框架选择 S 个解。更具体地说，第 32 行说明了在推送搜索阶段，S 个体是通过使用非支配排序方法且不考虑任何约束来选择的。第 34 行说明了在拉搜索阶段，使用改进的 Epsilon 约束处理方法选择 S 个解。在第 35 行，生成计数器进行更新。在第 40 行，选择了一组非支配且可行的解。

算法 4-4：PPS-M2M

Input:

K: the number of subproblems;

K unit direction vectors: $\boldsymbol{v_1}$,$\boldsymbol{v_2}$,...,$\boldsymbol{v_K}$;

S: the size of each subpopulation;

Q: a set of individual solutions and their F-values;

T_c: the control generation for $\epsilon(k)$;

T_{max}: the maximum generation.

Output:

a set of non-dominated and feasible solutions.

1 **Initialization:**

2 Decompose a population into K sub-populations (M_1,...,M_k), each sub-population contains S individuals according to $AllocationSubPop(Q, S, K)$;

3 Set $r_k = 1.0$, $PushStage = true$;

4 **while** $gen \leq T_{max}$ **do**

5 **for** $i \leftarrow 1$ **to** K **do**

6 **foreach** $x \in M_k$ do

7 Randomly choose y from M_k;

8 Apply genetic operators on x and y to generate a new solution z;

9 $R: = R \cup \{z\}$;

10 **end**

11 $Q: = R \cup \{\cup_{k=1}^{K} M_k\}$;

12 Use Q to set M_1,...,M_k according to Eq. (4-1);

13 **end**

if $gen \geq l$ **then**
    Calculate r_k according to Eq. (4-2);
end
if $gen < T_c$ **then**
    if $r_k \leq \epsilon$ **and** $PushStage == true$ **then**
        $PushStage == false$;
        $\epsilon(gen) = \epsilon(0) = maxViolation$;
    else
        Update $\epsilon(gen)$ according to Eq. (4-5);
    end
else
    $\epsilon(k) = 0$;
end
for $i \leftarrow 1$ **to** K **do**
    if $|M_k| \leq S$ **then**
        randomly select $S - |M_k|$ solutions from Q and add them to M_k.
    else
        if $PushStage == true$ **then**
            $result = PushSubproblems(M_k, S)$;
        else
            $result = PullSubproblems(M_k, gen, T_{max})$;
        end
    end
end
$gen = gen + 1$;
end
Output the non-dominated and feasible solutions.

作为一个例子，PPS-M2M 的搜索过程如图 4-2 所示。首先，在目标空间中对五个方向向量$v^1,...,v^5$进行均匀采样，其中构造了 5 个不相邻的子区域$M_1,...M_5$。图 4-2(a)~(c)展示了工作子种群在不考虑任何约束的情况下搜索到无约束 PF。值得注意的是，在这种特殊情况下，一些位于无约束 PF 上的解是可行的。在拉搜索阶段，

不可行的解被拉到可行和非显性区域，如图 4-2(d)~(f)所示。

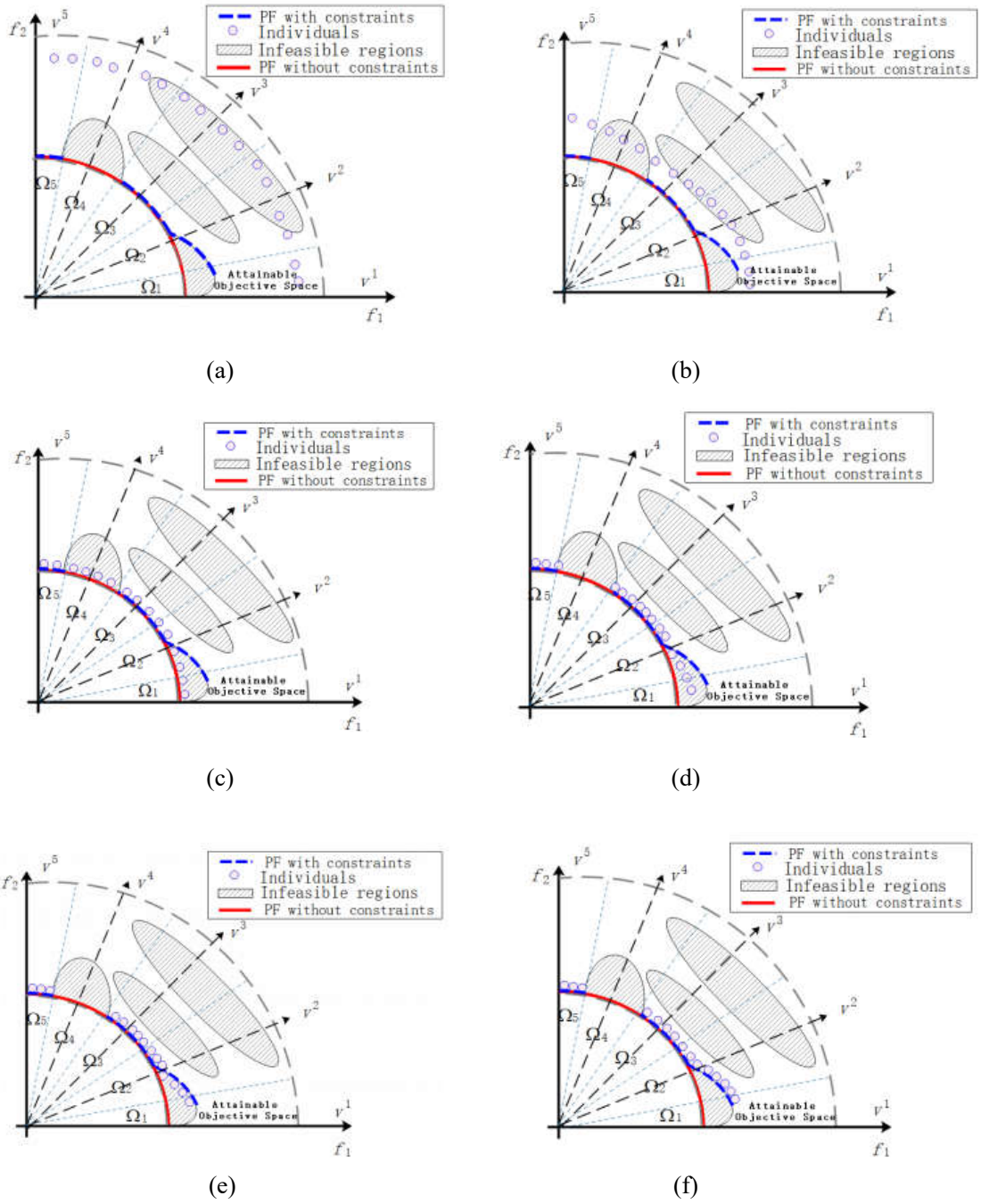


图 4-2 不可行区域使得原始无约束 PF 部分可行。目标空间分为 5 个子区域，5 个方向向量$v^1,...,v^5$均匀分布在单位球体的第一象限上。目标空间分为 5 个不相邻的子区域$M_1,...M_5$。(a)~(c)显示了

推搜索过程中每个子区域的工作种群跨越不可行区域，没有被阻挡。(d)~(f) 显示了拉搜索过程中每个子区域的工作种群中不可行的解被拉到可行和非支配区域。

4.1.4 PPS-M2M 与 PPS-MOEA/D 的差异

所提出的 PPS-M2M 和 PPS-MOEA/D[106]是 PPS 框架的两个不同的实例化。PPS-M2M 和 PPS-MOEA/D 之间的主要区别概述如下：

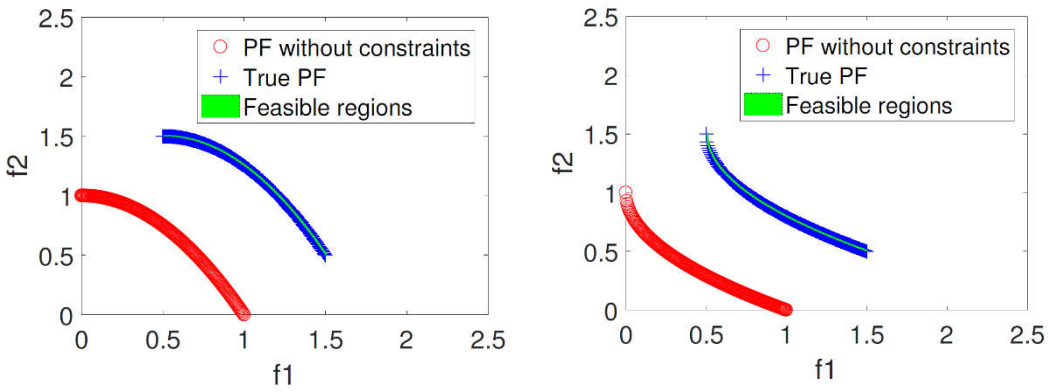
1. PPS-MOEA/D 和 PPS-M2M 使用不同分解方法。在 PPS-M2M 中，采用 M2M 分解方法将带约束的多目标(CMOPs)优化问题分解为一组简单的 CMOPs，而 PPS-MOEA/D 将 CMOP 分解为若干个单目标优化问题。

2. PPS-M2M 和 PPS-MOEA/D 使用不同的选择策略。在 PPS-M2M 中，将非支配排序方法和 IEpsilon 约束处理方法[104]相结合来选择下一代个体，PPS-MOEA/D 将分解方法和 IEpsilon 约束处理方法[104]相结合来选择下一代个体。

3. 尽管 PPS-M2M 和 PPS-MOEA/D 都是有效的约束多目标进化算法，但它们适用于求解具有不同特性的 CMOPs。PPS-M2M 更适合求解目标函数不平衡、约束函数具有多样性困难的 CMOPs，而 PPS-MOEA/D 更适合求解具有较大的不可行域的 CMOPs。详情见第 4.4.2 节

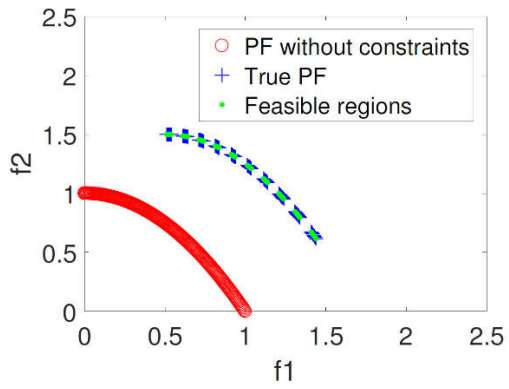
4.2 多目标约束优化测试问题集

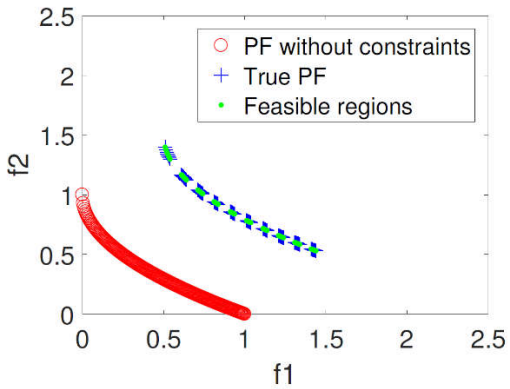
为了测试 CMOGAs 在处理 CMOPs 时的性能，本文中使用 LIR-CMOP[104][111] 测试问题集对算法性能进行测试，详情见附表 I 中。该问题的特点是具有不同特点的不行区域(如不可行区域在真实了 PF 下面、不可行区域在真实了 PF 上面和部分不可行区域覆盖了真实的 PF)，这导致种群很难收敛到真实的 PF。此外，目前大多数进化算法认为不可行区域对种群搜索都是同等重要，而 LIR-CMOPs 问题诞生说了这种观点是不对。只有离真实 PF 近的不可行区域对种群搜索才有较重的影响。



(a) LIR-CMOP1

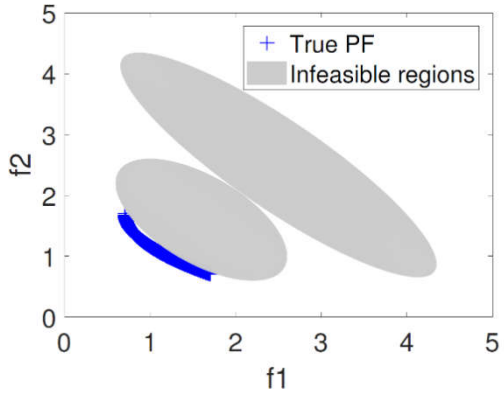
(b) LIR-CMOP2

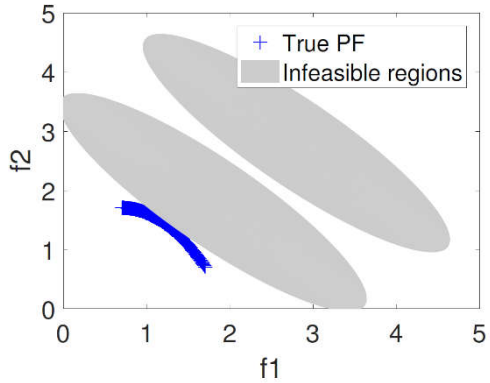




(c) LIR-CMOP3

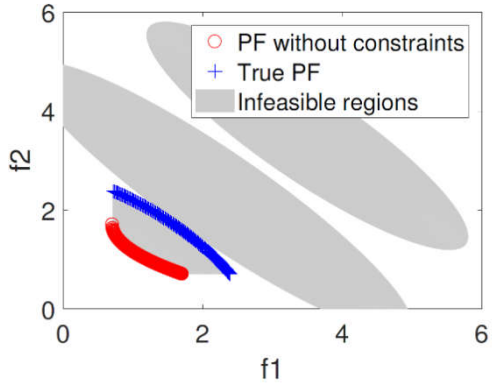
(d) LIR-CMOP4

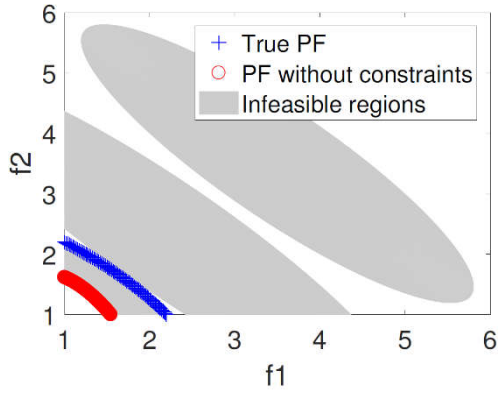




(e) LIR-CMOP5

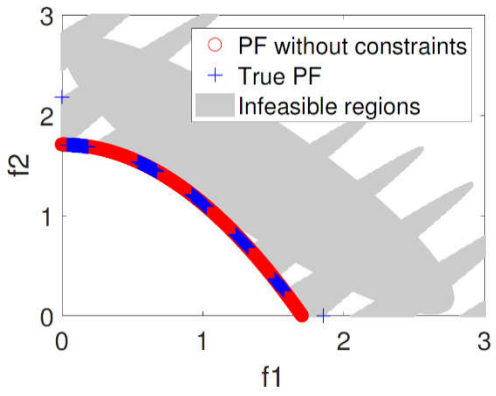
(f) LIR-CMOP6

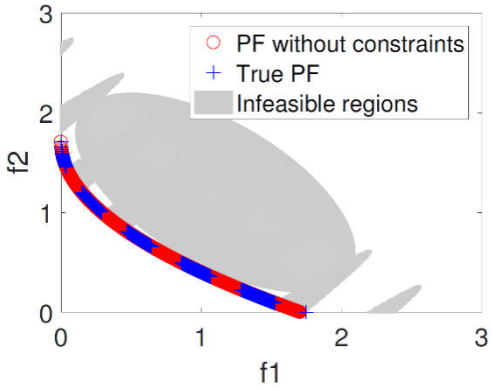




(h) LIR-CMOP7

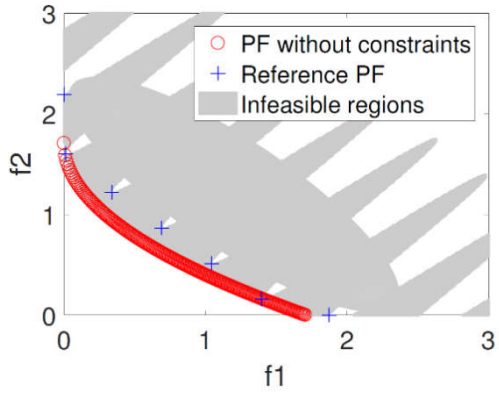
(i) LIR-CMOP8

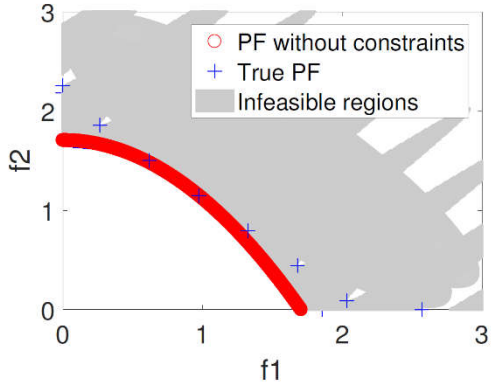




(j) LIR-CMOP9

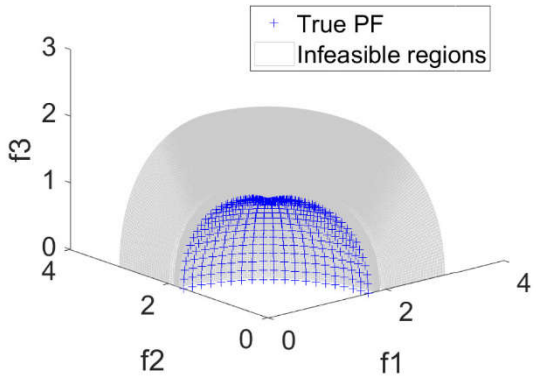
(k) LIR-CMOP10

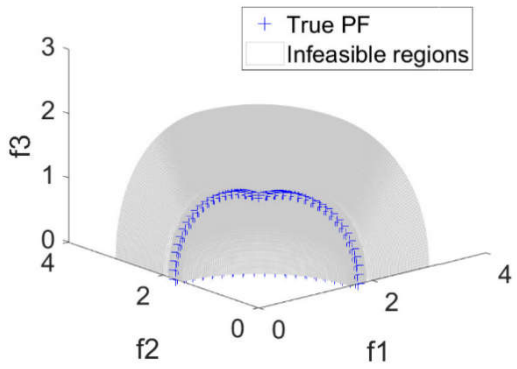




(l) LIR-CMOP11

(m) LIR-CMOP12



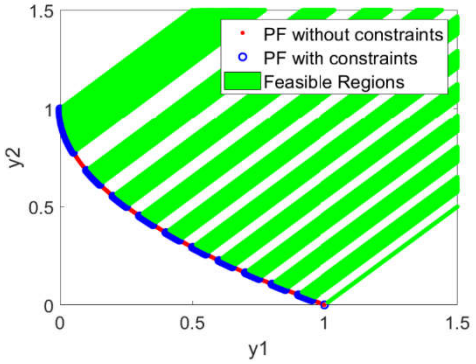


(o) LIR-CMOP13　　(p) LIR-CMOP14

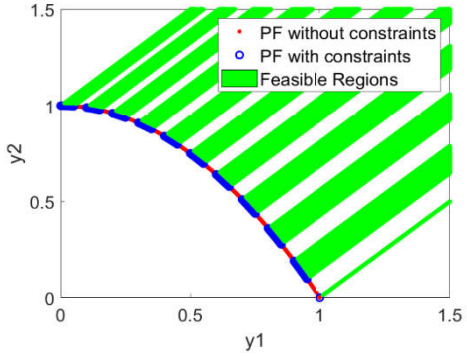
图 4-3　测试问题 LIR-CMOP1-14 在目标空间上可行区域及不可行区域的分布情况

如图 4-3 所示，LIR-CMOP1-14 具有不同特征的不可行区域(如不可行区域包含了真实了 PF、部分不可行区域覆盖了真实的 PF 等)，这使得种群搜索时候很难收敛到真实 PF 上。

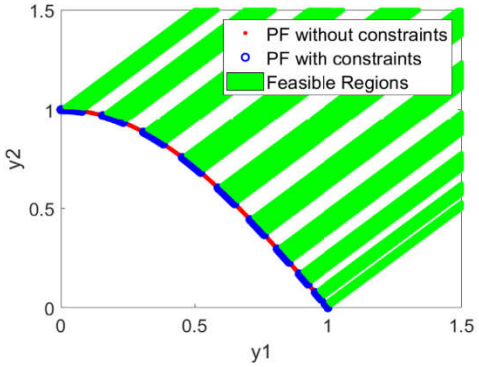
此外，本文根据 MOPs[105]和 DASCMOP[112]设计了一组新的约束和不平衡多目标优化问题(Constrained and Imbalanced Multi-objective Optimization Problems，CIMOPs)，详情见附表 II 中。这类测试问题的一般特性是它们的目标函数与 MOPs[105]是一样的，即它们都是不平衡的。由 DAS-CMOPs[112]启发，CIMOPs 的约束函数可以用多样性难度来设置的，如下图所示。此外，前 5 个问题的优化目标都是两维的，后 2 个问题的优化目标都是三维的。



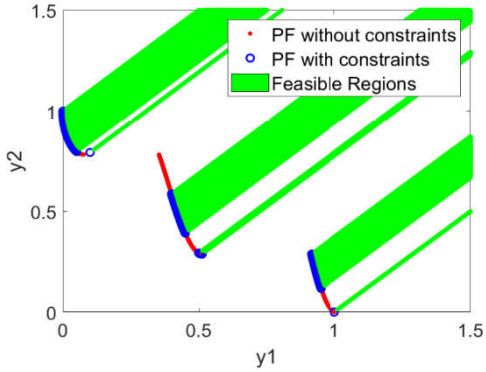
(a) CIMOP1



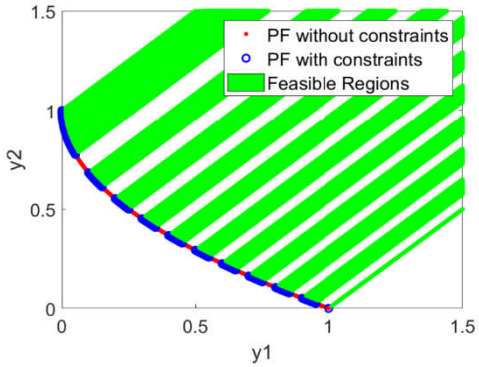
(b) CIMOP2

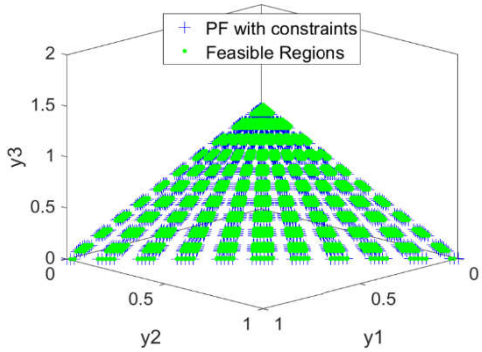


(c) CIMOP3

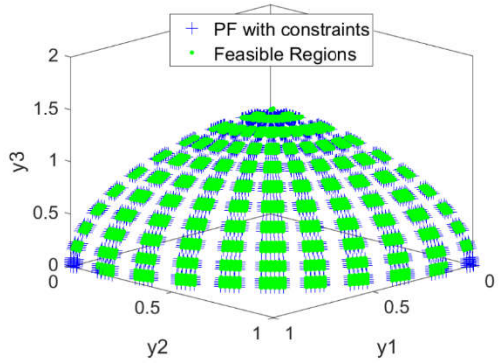


(d) CIMOP4





(e) CIMOP5　　(f) CIMOP6



(g) CIMOP7

图 4-4 测试问题 LIR-CMOP1-14 在目标空间上可行区域及不可行区域的分布情况

4.3 实验研究

为了评估本文提出的 PPS-M2M 算法的性能，我们通过多种算法在 LIR-CMOP 测试问题集的性能来验证提出的算法，这 7 类算法分别是 CM2M[107]，MOEA/D-Epsilon[100]，MOEA/D-SR[108]，MOEA / D-CDP[108]，C-MOEA/D[109]，NSGA-II-CDP[21]和 CM2M2 [110]。此外，为了评估本文提出的 PPS-M2M 与 PPS-MOEA/D 的性能，我们设置了一个新的测试问题集——CIMOPs

4.3.1 几种经典的约束多目标进化算法

在本节中，我们使用 PPS MOEA/D [106]，CM2M[107]，MOEA/D-Epsilon[100]，MOEA/D-SR[108]，MOEA/D-CDP[108]，C-MOEA / D[109]，NSGA-II-CDP[21]和 CM2M2 [110]与本文提出的 PPS-M2M 算法进行比较。

4.3.2 实验参数设置

为了保证实验比较的公平性，所有算法使用相同机制时设置的参数都是相同的，如下：

1)种群大小和终止条件：在所有算法中，种群大小 N 均设置为 300。 此外，当达到 300,000 次评价次数(Function Evaluations，FE)时，将终止这这些算法。

2)参数设置：这些算法应用 DE 遗传算子生成后代。这些算法的交叉和变异算子被设置为相同的控制参数，分别为 $F = 0.5$，$CR = 1.0$，其中 F 和 CR 分别为尺度因子(Scale Factor)和交叉率(Crossover Rate)的值。此外，我们使用标准工具包来实现对等算法，并遵循他们最初研究中的建议来设置算法参数。具体来说，PPS-M2M 和 PPS-MOEA/D 采用了一种改进的 Epsilon 约束处理方法，并将相关参数设置为[104]中的参数，详细如下：$\alpha = 0.95$，$\tau = 0.1$，$T_c = 800$，$\theta = 0.05N$。PPS-M2M、CM2M 和 CM2M2 将目标空间分解为 K 个子区域，因此它们的 K 个子区域是相同的，K 是根据[107]设置的($K = \lfloor N \rfloor$)。MOEA/D-Epsilon、MOEA/D-SR、MOEA/D-CDP 都是基于分解的算法，它们的邻域大小 T 设置为 $0.1N$。MOEA/D-Epsilon 采用了 Epsilon 约束处理方法，相关参数都遵循[100]设计的。根据[]，MOEA/D-SR 中的随机排序(Stochastic Ranking，S_r)被设置为 0.05，且S_r是用来直接平衡目标函数和惩罚函数。在 CM2M 中，按照[107]不可行权重N_1被设置为 90，可行权重N_2设置为 210。

4.4 实验分析及结论

为了评估所提出的算法(PPS-M2M)和 8 种最先进的 CMOEAs(PPS-MOEA/D[106]、CM2M[107]，MOEA/D-Epsilon[100]，MOEA/D-SR[108]，MOEA/D-CDP[108]，C-MOEA / D[109]，NSGA-II-CDP[21]和 CM2M2 [110])的性能，本文采用两种通用的性能指标，分别为反世代距离(Inverted Generation Distance，IGD)[52]和超体积(Hypervolume，HV)[53]。值得注意的是，IGD 和 HV 已在第 1.3.3 节详细介绍过。

4.4.1 LIR-CMOP 测试问题实验结果

对于测试问题集 LIR-CMOP1-14，表 4-1 和表 4-2 展示了 8 种约束多目标进化算法 PPS-M2M、CM2M[107]，MOEA/D-Epsilon[100]，MOEA/D-SR[108]，MOEA/D-CDP[108]，C-MOEA / D[109]，NSGA-II-CDP[21]和 CM2M2 [110]测试结果。黑体为针对每种测试问题最好的解。通过 IGD 和 HV 的比较，本文提出的算法在大多数情况都优于其他对比算法。

表 4-1　PPS-M2M 及其他 7 种比较的算法在测试问题 LIR-CMOP1-14 上的 IGD 结果

LIR-CMOP	PPS-M2M	C-MOEA/D	MOEA/D-CDP	MOEA/D-Epsilon	MOEA/D-SR	NSGA-II-CDP	CM2M	CM2M2

1	mean	**1.330E-02**	1.591E-01	1.348E-01	8.234E-02	4.406E-02	3.23E-01	3.106E-02	1.356E-02
	std	4.407E-03	3.534E-02	5.996E-02	5.321E-02	3.360E-02	7.33E-02	9.888E-03	2.993E-03
2	mean	1.485E-02	1.462E-01	1.549E-01	4.708E-02	2.057E-02	3.03E-01	2.742E-02	**1.167E-02**
	std	5.576E-03	4.141E-02	2.966E-02	1.339E-02	1.072E-02	7.24E-02	8.596E-03	3.021E-03
3	mean	1.987E-02	2.309E-01	2.268E-01	7.858E-02	1.529E-01	4.08E-01	4.383E-02	**1.395E-02**
	std	7.371E-03	4.135E-02	4.403E-02	2.978E-02	7.688E-02	1.15E-01	1.445E-02	5.867E-03
4	mean	2.504E-02	2.080E-01	2.188E-01	5.662E-02	2.038E-01	3.85E-01	4.187E-02	**1.196E-02**
	std	9.030E-03	4.197E-02	3.766E-02	3.366E-02	7.907E-02	1.35E-01	2.371E-02	5.604E-03
5	mean	**3.922E-03**	1.162E+00	1.207E+00	1.201E+00	1.123E+00	5.53E-01	2.941E-01	1.181E+00
	std	1.515E-03	2.180E-01	1.660E-02	1.963E-02	2.842E-01	6.88E-01	5.013E-01	1.826E-02
6	mean	**5.077E-03**	1.265E+00	1.303E+00	1.231E+00	1.175E+00	5.74E-01	5.356E-01	9.915E-01
	std	1.117E-02	3.067E-01	2.319E-01	3.602E-01	3.967E-01	4.21E-01	5.509E-01	5.510E-01
7	mean	**3.711E-03**	1.620E+00	1.623E+00	1.568E+00	1.136E+00	2.38E-01	5.237E-01	7.077E-02
	std	1.993E-03	3.036E-01	2.905E-01	4.101E-01	7.315E-01	4.06E-01	7.677E-01	5.172E-02
8	mean	**2.949E-03**	1.607E+00	1.631E+00	1.577E+00	1.369E+00	6.02E-01	7.924E-01	6.582E-02
	std	9.666E-05	2.680E-01	2.464E-01	3.767E-01	5.735E-01	7.39E-01	7.455E-01	5.937E-02
9	mean	**3.704E-01**	4.981E-01	4.868E-01	4.962E-01	4.813E-01	6.44E-01	4.599E-01	4.197E-01
	std	2.529E-02	6.991E-02	5.372E-02	6.987E-02	4.571E-02	1.60E-02	6.426E-02	8.113E-02
10	mean	**1.572E-02**	3.775E-01	3.774E-01	3.257E-01	2.821E-01	5.97E-01	2.291E-01	6.108E-01
	std	4.656E-02	7.446E-02	6.858E-02	9.833E-02	1.135E-01	3.20E-02	8.123E-02	1.375E-01

11	mean	**2.770E-02**	4.422E-01	4.662E-01	4.154E-01	3.489E-01	4.87E-01	4.400E-01	5.135E-01
	std	4.656E-02	1.759E-01	1.439E-01	1.508E-01	1.129E-01	1.05E-01	1.038E-01	1.131E-01
12	mean	**9.310E-02**	3.597E-01	3.236E-01	3.680E-01	3.012E-01	5.80E-01	1.488E-01	2.549E-01
	std	9.310E-02	1.074E-01	1.023E-01	8.664E-02	8.989E-02	1.17E-01	5.757E-02	1.592E-02
13	mean	**1.874E-01**	1.266E+00	1.289E+00	1.183E+00	1.093E+00	1.39E+01	1.57E+00	1.263E+00
	std	2.690E-02	2.173E-01	6.321E-02	3.456E-01	4.269E-01	2.26E-01	1.05E-02	2.254E-01
14	mean	**1.746E-01**	1.235E+00	1.103E+00	1.127E+00	1.143E+00	1.36E+01	2.39E+00	1.220E+00
	std	2.840E-02	1.209E-01	3.857E-01	3.329E-01	3.002E-01	2.17E+00	1.63E-02	2.192E-01
Friedman Test		**1.4286**	7	6.1786	5.8571	4.0714	7.4286	4.7857	4.6429

表 4-2　PPS-M2M 及其他 7 种比较的算法在测试问题 LIR-CMOP1-14 上的 HV 结果

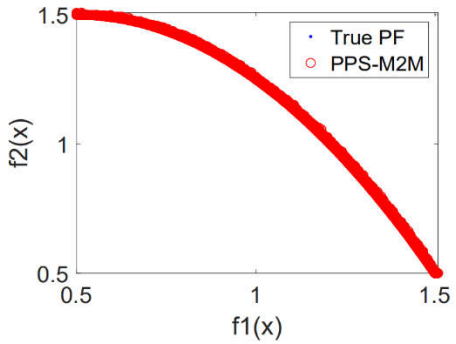
LIR-CMOP		**PPS-M2M**	**C-MOEA/D**	**MOEA/D-CDP**	**MOEA/D-Epsilon**	**MOEA/D-SR**	**NSGA-II-CDP**	**CM2M**	**CM2M2**
1	mean	**1.009E+00**	9.499E-01	7.540E-01	9.590E-01	9.960E-01	5.160E-01	9.897E-01	1.008E+00
	std	4.590E-03	7.038E-02	8.950E-02	3.280E-02	2.910E-02	5.570E-02	1.322E-02	3.563E-03
2	mean	1.337E+00	1.395E+01	1.374E+01	1.280E+00	1.340E+00	8.240E-01	1.321E+00	**1.345E+00**
	std	7.093E-03	8.154E-02	6.160E-02	2.880E-02	1.470E-02	1.150E-01	1.124E-02	4.409E-03
3	mean	**1.095E+00**	7.558E-01	7.600E-01	7.980E-01	5.910E-01	4.080E-01	8.407E-01	8.645E-01
	std	6.010E-03	5.730E-02	5.809E-02	3.930E-02	1.070E-01	2.880E-02	1.787E-02	5.795E-03
4	mean	**1.095E+00**	1.069E+00	1.051E+00	1.020E+00	8.150E-01	6.170E-01	1.051E+00	1.088E+00
	std	8.869E-03	6.952E-02	5.462E-02	4.190E-02	8.700E-02	1.060E-01	3.870E-02	3.633E-03

5	mean	1.455E+00	1.192E-01	5.805E-02	4.300E-02	1.820E-01	9.390E-01	1.084E+00	1.513E-01
	std	6.935E-03	3.352E-01	4.042E-04	2.350E-01	4.390E-01	3.210E-01	6.097E-01	4.184E-01
6	mean	**1.118E+00**	7.863E-02	4.312E-02	5.400E-02	3.020E-01	4.130E-01	5.337E-01	1.455E-01
	std	3.888E-02	3.011E-01	2.362E-01	2.210E-01	4.620E-01	1.890E-01	3.987E-01	3.034E-01
7	mean	**3.005E+00**	2.990E-01	2.886E-01	3.030E-01	9.880E-01	2.400E+00	2.021E+00	2.776E+00
	std	2.434E-01	6.927E-01	6.348E-01	9.070E-01	1.270E+00	6.520E-01	1.349E+00	1.706E-01
8	mean	**3.016E+00**	3.246E-01	2.695E-01	1.060E-01	1.100E+00	1.900E+00	1.498E+00	2.769E+00
	std	2.494E-02	5.878E-01	5.297E-01	5.490E-01	1.200E+00	7.560E-01	1.277E+00	2.149E-01
9	mean	**3.152E+00**	3.715E+00	3.755E+00	2.740E+00	2.750E+00	2.060E+00	2.826E+00	2.941E+00
	std	8.752E-02	2.079E-01	1.600E-01	1.480E-01	1.640E-01	1.080E-02	2.114E-01	1.698E-01
10	mean	**3.216E+00**	3.274E+00	3.268E+00	2.890E+00	2.930E+00	2.040E+00	2.845E+00	1.847E+00
	std	7.555E-02	1.623E-01	1.416E-01	1.020E-01	1.350E-01	4.450E-02	1.720E-01	4.934E-01
11	mean	**4.343E+00**	3.937E+00	3.842E+00	3.340E+00	3.380E+00	3.110E+00	3.115E+00	2.738E+00
	std	1.401E-01	6.479E-01	5.507E-01	2.570E-01	2.900E-01	1.540E-02	3.003E-01	5.532E-01
12	mean	**5.409E+00**	5.977E+00	6.134E+00	4.880E+00	5.030E+00	3.280E+00	5.221E+00	4.900E+00
	std	1.717E-01	3.855E-01	3.617E-01	3.170E-01	1.750E-01	3.610E-01	1.878E-01	1.115E-01
13	mean	**4.874E+00**	6.444E-01	4.728E-01	4.550E-01	1.890E+00	0.000E+00	4.566E-01	3.119E+00
	std	1.068E-01	1.317E+00	2.689E-01	1.300E+00	2.570E+00	0.000E+00	2.100E-02	2.746E+00
14	mean	**5.449E+00**	7.766E-01	1.627E+00	1.330E+00	1.270E+00	0.000E+00	4.142E-01	2.313E+00
	std	1.237E-01	6.140E-01	2.473E+00	2.450E+00	2.290E+00	0.000E+00	5.110E-03	2.835E+00

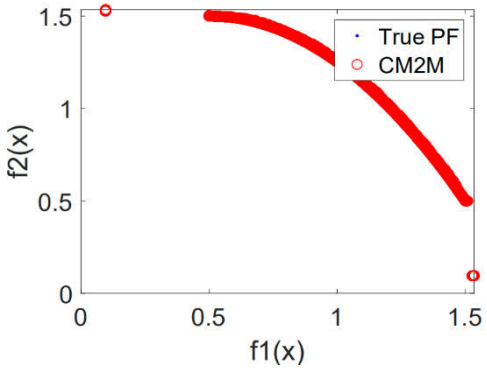
Friedman Test	**1.1786**	6.3571	5.5357	5.57	3.8929	6.6429	3.9286	3.5

为了进一步讨论所提出的 PPS-M2M 在求解 CMOPs 方面的优势，我们在 LIR-CMOP1、LIRC MOP7 和 LIR-CMOP11 上绘制了各种算法在 HV 中位数上的非支配解集。图 4-3 展示了 LIR-CMOP1、LIR-CMOP7 和 LIR-CMOP11 的可行和不可行区域，以及 PF。

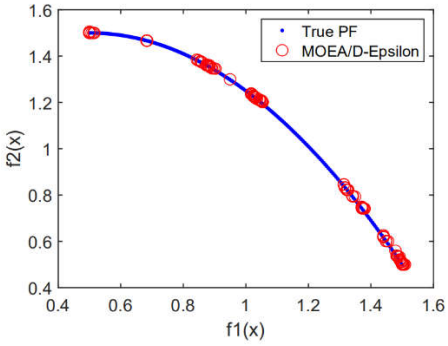
如图 4-3 (a)所示，LIR-CMOP1 的可行区域很小，这表明该问题是可行性困难问题。在图 4-5 中绘制了每种算法在 LIR-CMOP1 上实现的非支配解。我们可以观察到 PPS-M2M、CM2M、MOEA/D-SR 和 CM2M2 可以收敛到真正的 PF 并具有良好的多样性。然而，根据 HV 和 IGD 指标，所提出的 PPS-M2M 优于其他比较算法。图 4-5 展示了 CM2M 的收敛性比 PPS-M2M 的较差。一个可能的原因是一些子区域被分配到 CM2M 中不可行的区域，而这些子区域中的解不能收敛到真正的 PF。



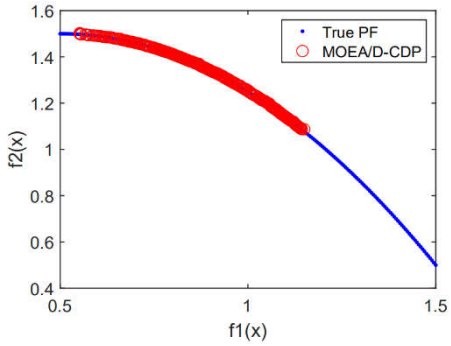
(a) PPS-M2M



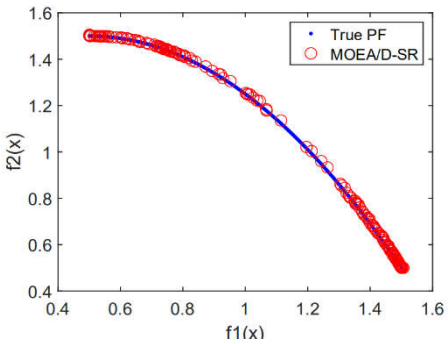
(b) CM2M



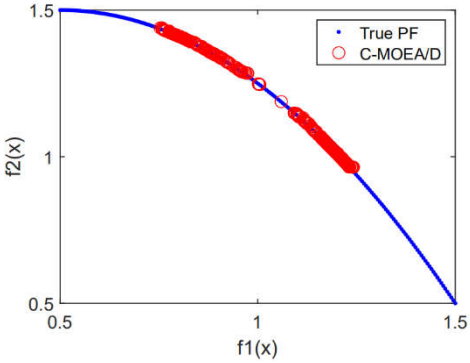
(c) MOEA/D-Epsilon



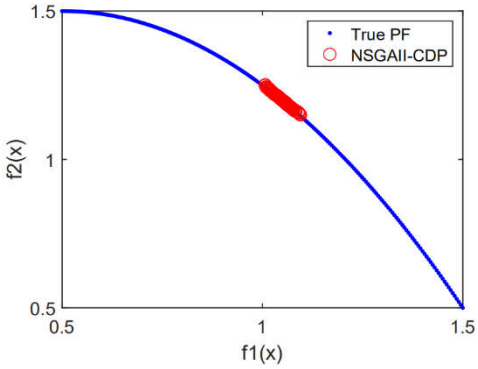
(d) MOEA/D-CDP



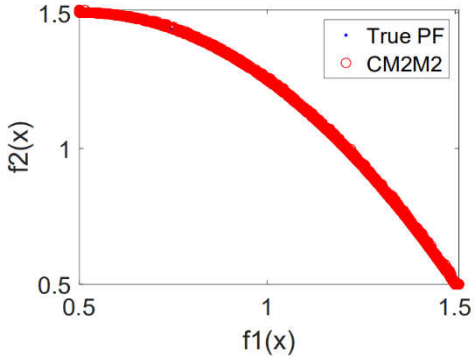
(e) MOEA/D-SR



(f) C-MOEA/D

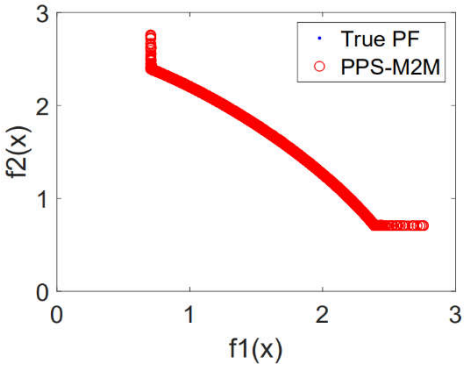


(g) NSGA-II-CDP

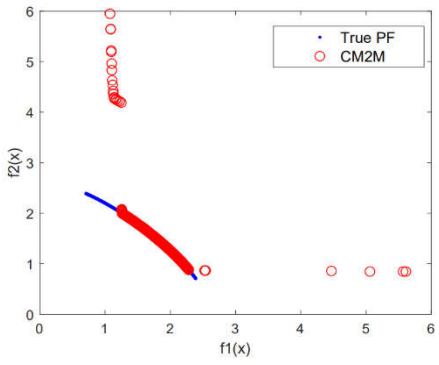


(f) CM2M2

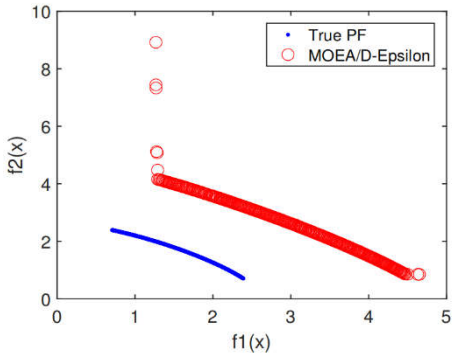
图 4-5 每种算法在 LIR-CMOP1 上获得的非支配解集



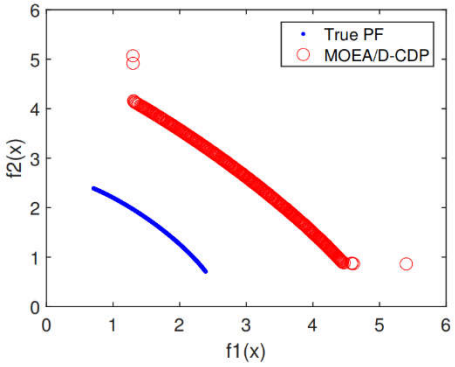
(a) PPS-M2M



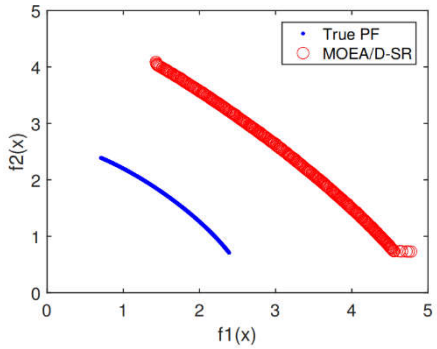
(b) CM2M



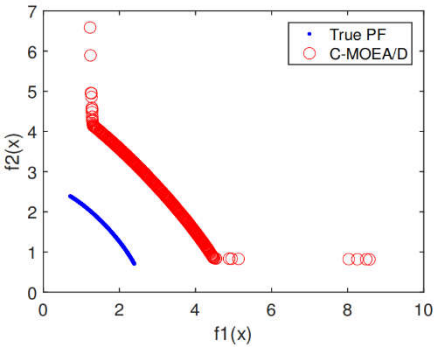
(c) MOEA/D-Epsilon



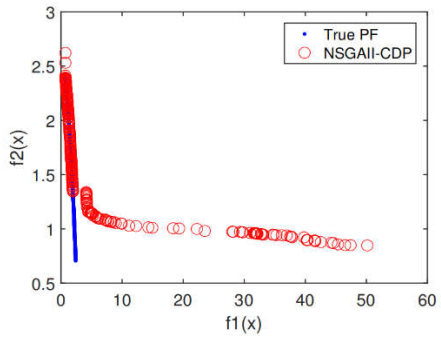
(d) MOEA/D-CDP



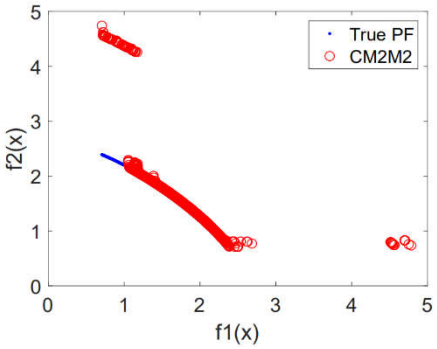
(e) MOEA/D-SR



(f) C-MOEA/D



(g) NSGA-II-CDP



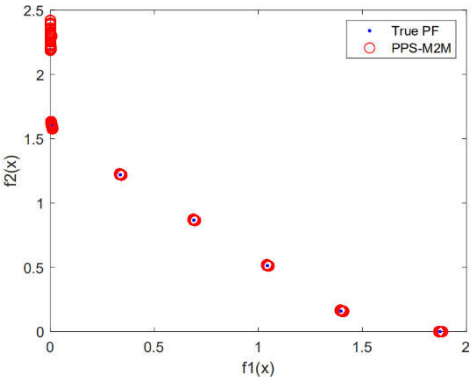
(f) CM2M2

图 4-6 每种算法在 LIR-CMOP7 上获得的非支配解集

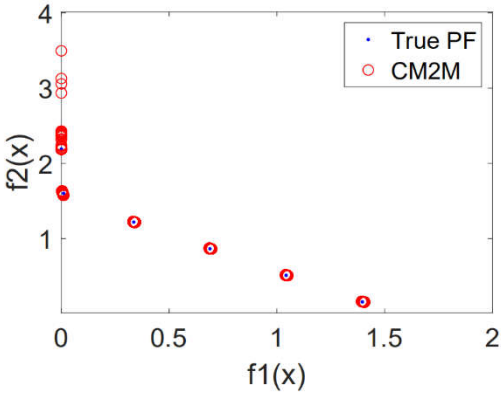
图 4-3 展示了在 LIR-CMOP7 上，在无约束 PF 前有三个大的不重叠不可行区域。此外，无约束 PF 被一个不可行区域所覆盖，这表明了一个收敛性困难的问题。图 4-6 展示出了 PPS-M2M 和其他 7 种算法(CM2M、MOEA/D-Epsilon、MOEA/D-SR、MOEA/DCDP、C-MOEA/D、NSGA-II-CDP 和 CM2M2)在 LIR-CMOP7 上的结果。我们可以看到，只有 PPS-M2M 才能跨越不可行的区域，达到整个真实 PF，其原因

是 PPS-M2M 在推搜索阶段忽略了的约束，可以方便地跨越不可行区域。然而，其他比较算法不能有效地跨越三个不可行区域，因为 PF 前面的两个大的不可行区域阻碍了种群的收敛到真实 PF。

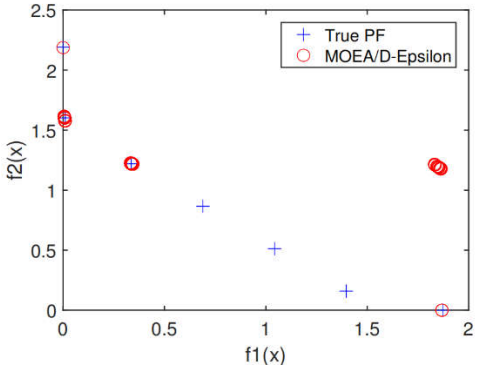
如图 4-3 所示，LIR-CMOP11 的 PF 是离散的，这表明该问题多样性困难问题。此外，LIR-CMOP11 有 7 个 Pareto 最优解，其两个解位于无约束 PF 上，五个解位于约束 PF 上。图 4-7 给出了这些算法在 LIR-CMOP-CMOP11 上的结果。在图 4-7 上，我们很容易发现只有 PPS-M2M 和 CM2M 才能找到所有的 Pareto 最优解。此外，与 CM2M 相比，PPS-M2M 具有良好的收敛性。一个可能的原因是，在最后 10%代上，PPS-M2M 将每个子群体被组合成一个整体，再使用改进的 Epsilon 约束处理方法进化而来的。而其余的算法(MOEA/D-Epsilon、MOEA/D-SR、MOEA/D-CDP、C-MOEA/D、NSGAII-CDP 和 CM2M2)只能找到一部分的 Pareto 最优解，因为不可行的区域阻碍了这些算法的种群收敛到真正的 PF 上。此外，在这些算法中很难正确地设置约束处理方法的参数，因为在搜索过程中没有很好地探索和估计约束的景观。



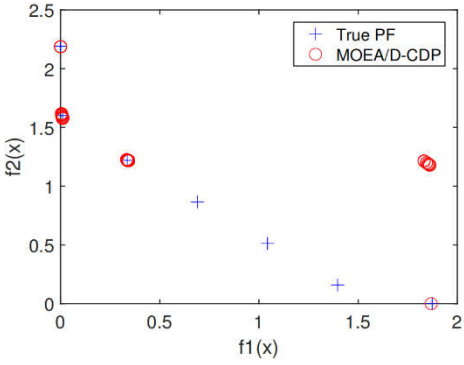
(a) PPS-M2M



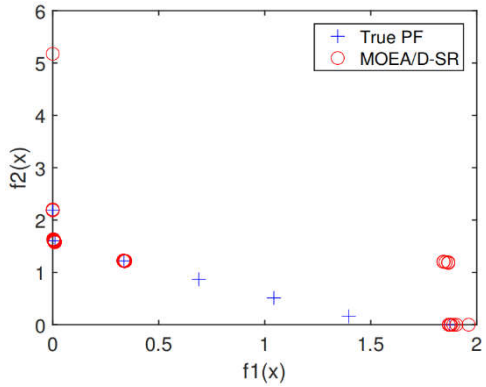
(b) CM2M



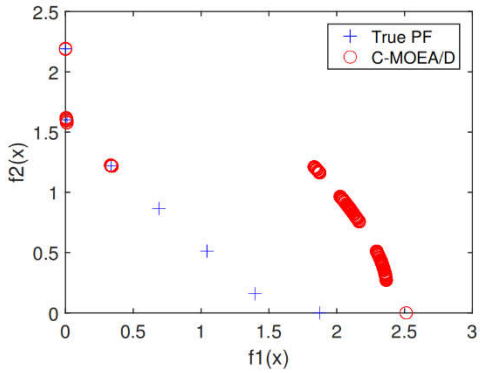
(c) MOEA/D-Epsilon



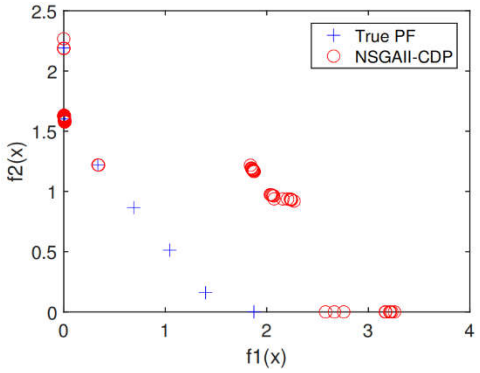
(d) MOEA/D-CDP



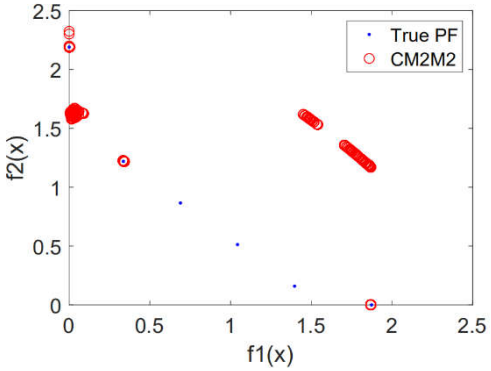
(e) MOEA/D-SR



(f) C-MOEA/D



(g) NSGA-II-CDP



(f) CM2M2

图 4-7 每种算法在 LIR-CMOP11 上获得的非支配解集

根据上述观察和分析，我们可以得出结论，所提出的 PPS-M2M 在大多数测试问题集上优于其他 7 种 CMOEAs(CM2M、MOEA/D-Epsilon、MOEA/D-SR、MOEA/D-CDP、C-MOEA/D、NSGA-II-CDP 和 CM2M2)。

4.4.2 CIMOP 测试问题实验结果

为了说明 PPS-M2M 和 PPS-MOEA/D 之间的差异，我们比较了 PPS-M2M 和 PPS-MOEA/D 在 LIR-CMOPs 和 CIMOPs 上的性能。附于附录页附表 II 展示了 CIMOPs 具体的目标函数及约束条件定义。这类测试问题的一般特性是 CIMOPs 的目标函数与 MOPs[105]是一样的，即它们都是不平衡的。由 DAS-CMOPs[112]启发，CIMOPs 的约束函数可以用多样性难度来设置的。换句话说，CIMOPs 具有不平衡的目标函数和多样性困难的约束函数，而 LIR-CMOPs 具有较大的不可行区域。

首先，在 LIR-CMOP1-14 上比较了所提出的 PPS-M2M 和 PPS-MOEA/D，如表 4-3 所示。此外，表 4-3 显示了 PPS-M2M 和 PPS-MOEA/D 在 LIR-CMOP1-14 上的

IGD 和 HV 值。从这些表中，我们可以观察到虽然 PPS-MOEA/D 在 LIR-CMOP1-14 上优于所提出的 PPS-M2M，但是他们之间差别是不大的。值得注意的是，PPS-M2M 可以比其他 7 种算法(CM2M、MOEA/D-Epsilon、MOEA/D-SR、MOEA/D-CDP、C-MOEA/D、NSGA-II-CDP、和 CM2M2)更好的结果，如表 4-1 和表 4-2 所示。

表 4-3　PPS-M2M 和 PPS-MOEA/D 在测试问题 LIR-CMOP1-14 上的 IGD 和 HV 结果

LIR-CMOP		**IGD**		**HV**	
		PPS-M2M	**PPS-MOEA/D**	**PPS-M2M**	**PPS-MOEA/D**
1	mean	1.330E-02	**6.413E-03**	1.009E+00	**1.016E+00**
	std	4.407E-03	1.938E-03	4.590E-03	1.580E-03
2	mean	1.485E-02	**4.673E-03**	1.337E+00	**1.349E+00**
	std	5.576E-03	7.844E-04	7.093E-03	1.010E-03
3	mean	1.987E-02	**8.545E-03**	1.095E+00	**8.703E-01**
	std	7.371E-03	5.184E-03	6.010E-03	2.650E-03
4	mean	2.504E-02	**4.677E-03**	**1.095E+00**	1.093E+00
	std	9.030E-03	1.116E-03	8.869E-03	2.467E-03
5	mean	3.922E-03	**1.837E-03**	1.455E+00	**1.462E+00**
	std	1.515E-03	9.263E-05	6.935E-03	2.919E-04
6	mean	5.077E-03	**2.490E-03**	1.118E+00	**1.129E+00**
	std	1.117E-02	3.399E-04	3.888E-02	1.771E-04
7	mean	3.711E-03	**2.797E-03**	3.005E+00	**3.015E+00**
	std	1.993E-03	9.854E-05	2.434E-01	2.663E-03
8	mean	2.949E-03	**2.778E-03**	3.016E+00	**3.017E+00**
	std	9.666E-05	7.558E-05	2.494E-02	1.139E-03
9	mean	3.704E-01	**9.940E-02**	3.152E+00	**3.570E+00**
	std	2.529E-02	1.519E-01	8.752E-02	2.242E-01
10	mean	1.572E-02	**2.108E-03**	3.216E+00	**3.241E+00**
	std	4.656E-02	7.754E-05	7.555E-02	3.077E-04
11	mean	2.770E-02	**2.832E-03**	4.343E+00	**4.390E+00**
	std	4.656E-02	1.359E-03	1.401E-01	2.217E-04
12	mean	9.310E-02	**2.704E-02**	5.409E+00	**5.614E+00**
	std	9.310E-02	5.002E-02	1.717E-01	1.525E-01
13	mean	1.874E-01	**6.455E-02**	4.874E+00	**5.710E+00**
	std	2.690E-02	2.177E-03	1.068E-01	1.275E-02
14	mean	1.746E-01	**6.419E-02**	5.449E+00	**6.193E+00**
	std	2.840E-02	1.690E-03	1.237E-01	1.310E-02
Friedman Test		2	1	1.9286	1.0714

然后，为了说明 PPS-M2M 与 PPS-MOEA/D 的区别，我们在 CIMOPs 上进行测试。表 4-4 和表 4-5 显示了四种算法(PPS-M2M、PPS-MOEA/D、CM2M 和 CM2M2)在 CIMOPs 上的 IGD 和 HV 值。值得说明的是，由于 CM2M 和 CM2M2 采用 M2M 分解方法，因此将它们包括在内以验证 PPS-M2M 的性能不仅依赖于 M2M 方法，而且还依赖于 PPS 框架中的约束处理机制。从表 4-4 和表 4-5，我们可以得出结论，PPS-M2M 明显优于其他三种算法(PPS-MOEA/D、CM2M 和 CM2M2)。

表 4-4　PPS-M2M 及其他 3 种比较的算法在测试问题 CIMOP1-7 上的 IGD 结果

CIMOPs		PPS-M2M	PPS-MOEA/D	CM2M	CM2M2
1	mean	**2.05E-02**	2.81E-01	7.96E-01	5.22E-02
	std	1.13E-03	8.44E-02	6.36E-03	1.79E-02
2	mean	**4.42E-02**	2.95E-01	6.38E-01	1.81E-01
	std	1.56E-02	1.29E-02	2.01E-02	3.18E-02
3	mean	**1.19E-01**	3.51E-01	5.58E-01	2.80E-01
	std	1.13E-02	3.06E-02	1.54E-02	3.66E-02
4	mean	**4.72E-03**	2.01E-01	6.47E-01	1.09E-02
	std	1.00E-03	2.92E-02	1.49E-02	3.44E-02
5	mean	**4.45E-02**	3.76E-01	6.60E-02	8.26E-02
	std	5.28E-03	4.59E-02	1.25E-02	2.23E-02
6	mean	**9.17E-02**	2.13E-01	6.98E-01	2.34E-01
	std	8.67E-03	4.46E-02	2.42E-02	2.46E-02
7	mean	**1.64E-01**	3.65E-01	8.00E-01	3.67E-01
	std	1.87E-02	1.64E-02	5.11E-02	4.51E-02
Friedman Test		1	2.8571	3.7143	2.4286

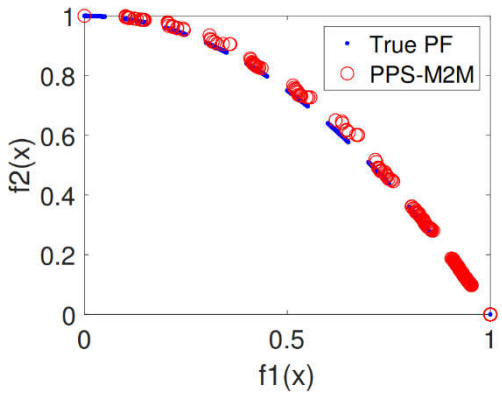
表 4-5　PPS-M2M 及其他 3 种比较的算法在测试问题 CIMOP1-7 上的 HV 结果

CIMOPs		PPS-M2M	PPS-MOEA/D	CM2M	CM2M2
1	mean	**1.065E+00**	6.623E-01	3.833E-01	1.032E+00
	std	1.359E-03	1.388E-01	5.250E-02	2.243E-02
2	mean	**7.251E-01**	4.620E-01	7.109E-01	5.237E-01
	std	2.415E-02	1.478E-02	9.110E-03	2.600E-02
3	mean	5.834E-01	2.470E-01	**6.630E-01**	3.384E-01
	std	2.507E-02	3.612E-02	2.260E-02	4.312E-01
4	mean	**9.288E-01**	5.999E-01	7.856E-01	8.019E-01
	std	2.686E-03	1.747E-02	1.080E-03	7.203E-02
5	mean	**1.028E+00**	6.550E-01	8.355E-01	9.617E-01
	std	7.116E-03	2.281E-02	6.810E-03	3.263E-02

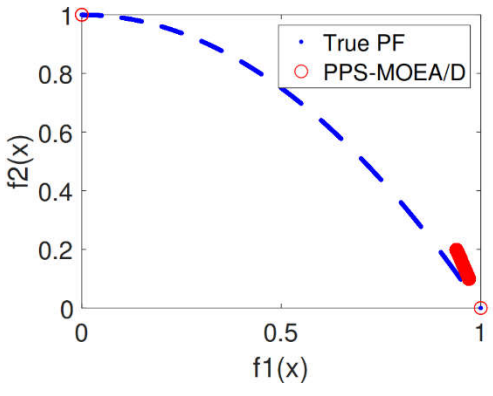
6	mean	**1.437E+00**	1.326E+00	8.748E-01	9.505E-01
	std	1.227E-02	5.988E-02	1.440E-02	4.994E-02
7	mean	**1.041E+00**	9.348E-01	8.319E-01	7.827E-01
	std	3.658E-02	6.182E-03	1.850E-02	3.010E-02
Friedman Test		1.1429	3.2857	2.8571	2.7143

为了说明所提出的 PPSM2M 在求解 CIMOPs 方面的优势，我们绘制了每种算法在 CIMOP2 和 CIMOP4 上实现的非支配解集，如图 4-8 和 4-9 所示。我们可以观察到 PPS-M2M 可以找到大多数 Pareto 最优解。然而，其他三种算法只能找到小部分的真实 PFs。一个可能的原因是 CIMOP2 和 CIMOP4 在约束函数中具有多样性困难。更具体地说，CIMOP2 和 CIMOP4 的目标函数是不平衡的。因此，MOEA/D 的分解方法只能找到 PFS 的几个部分。此外，通过观察图 4-8 和图 4-9，我们可以发现 CM2M 和 CM2M2 的收敛性能比 PPS-M2M 差，即 CM2M 和 CM2M2 得到的许多解不能收敛到真正的 PFs。一种可能的原因是 CIMOP2 和 CIMOP4 的约束 PF 是它们的无约束 PF 的子集。由于 PPS-M2M 首先收敛到无约束 PF，而不考虑任何约束，因此它可以很好地利用 PPS 机制处理这种约束的问题。但是，CM2M 和 CM2M2 采用了不同的约束处理机制，始终考虑可行的种群个体，这对于处理 CIMOP2 和 CIMOP4 的约束并不是那么有效。

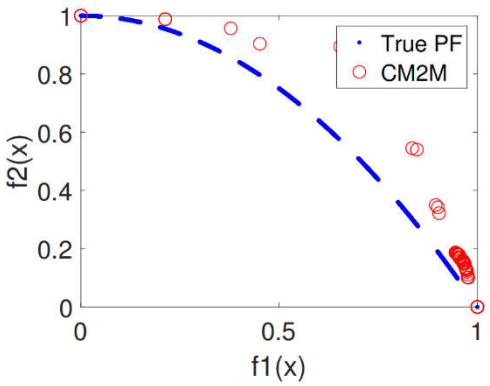
基于上述观察和分析，我们得出结论，PPS-M2M 是一种强大的算法，可以很好地处理具有大不可行区域或不平衡目标和约束呈现多样性困难的 CMOPs。与 PPS-MOEA/D 相比，PPS 框架的另一个实例化，PPS-M2M 更适合于求解目标不平衡和多样性困难的约束函数的 CMOPs，而 PPS-MOEA/D 更适合于求解具有较大不可行区域的 LIR-CMOPs。



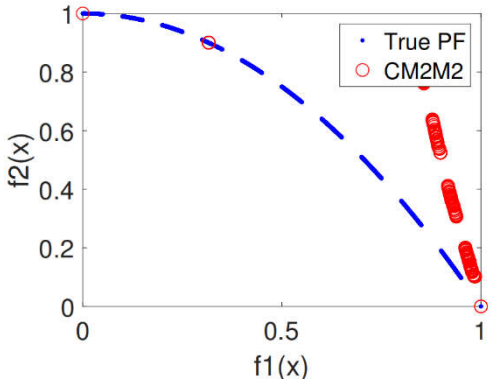
(a) PPS-M2M



(b) PPS-MOEA/D

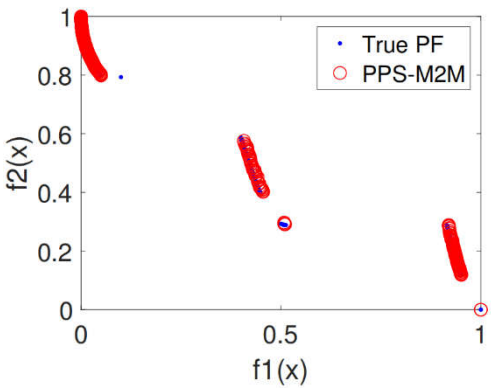


(c) CM2M

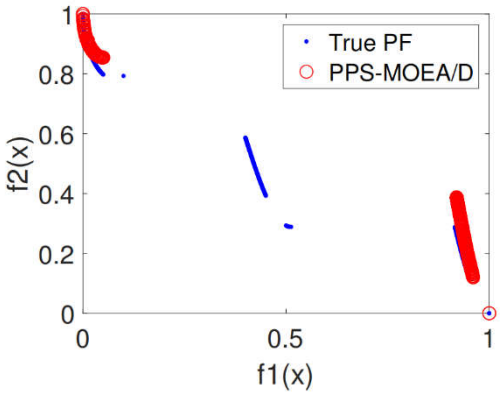


(d) CM2M2

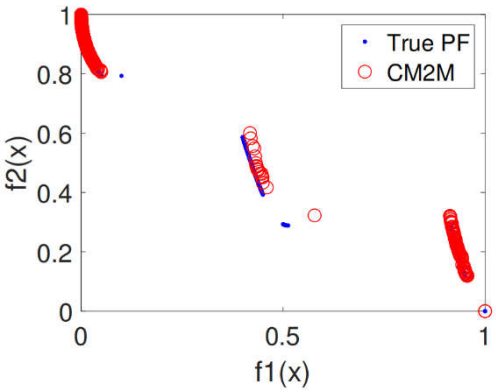
图 4-8 每种算法在 CIMOP2 上获得的非支配解集



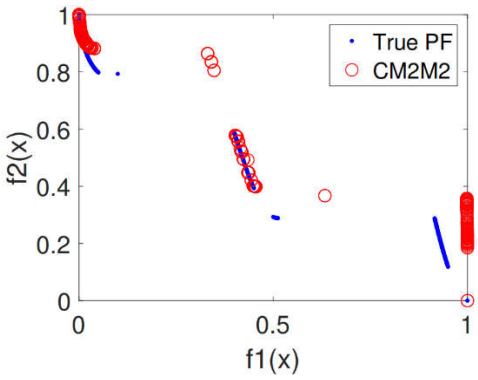
(a) PPS-M2M



(b) PPS-MOEA/D



(c) CM2M



(d) CM2M2

图 4-8 每种算法在 CIMOP4 上获得的非支配解集

4.5 本章小结

本文提出了一种新的算法，即 PPS-M2M，它将多目标到多目标(M2M)分解方法与推拉搜索(PPS)框架相结合，用以处理 CMOPs。更具体地说，PPS-M2M 的搜索过程分为两个阶段——即推拉搜索过程。在推送搜索阶段，PPS-M2M 使用 M2M 分解方法分解将一个 CMOP 分成一组简单的 CMOP，这些 CMOP 对应于一组子种群。每个简单的 CMOP 都是以协同的方式被解决的，由于不考虑任何约束，这可以帮助子群体毫不费力地跨越了不可行的区域。

此外，在推搜索阶段，算法还可以估计一些约束景观信息，如可行解与不可行解的比率和最大总体约束违反，它可以进一步用于指导拉搜索阶段约束处理机制的参数设置。当理想点和最低点之间的最大变化率小于或等于预定义的阈值时，搜索过程被切换到拉搜索阶段。在拉式搜索阶段开始时，通过采用改进的 Epsilon 约束处理方法，将在推式阶段获得的每个子种群的不可行解拉到可行和非支配区域。在最后 10%代中，所有子种群被合并成一个整体，再通过使用改进的 Epsilon 约束处理方法进一步搜索。综合实验结果表明，在大多数 LIR-CMOP1-14 上，所提出的 PPS-M2M 优于其他 7 种 CMOEAs(CM2M、MOEA/D-Epsilon、MOEA/D-SR、MOEA/D-CDP、C-MOEA/D、NSGAII-CDP 和 CM2M2)。

为了说明 PPS-M2M 和 PPS-MOEA/D 之间的差异，我们比较了 PPS-M2M 和 PPS-MOEA/D 在 LIRCMOP 和 CIMOP 上的性能。实验结果表明，所提出的 PPS-M2M 在 CIMOPs 上的效果明显优于 PPS-MOEA/D，而 PPS-MOEA/D 在 LIR-CMOPs 上的性能优于 PPS-M2M。但是 PPS-M2M 比其他 7 种算法(CM2M、MOEA/D-Epsilon、MOEA/D-SR、MOEA/D-CDP、C-MOEA/D、NSGA-II-CDP、和 CM2M2)性能优越显著。

从综合实验分析可以看出，PPS-M2M 是一种有效地处理大不可行区域或不平衡目标和约束函数有着多样性困难的 CMOP 的强大算法。与 PPS-MOEA/D 相比，PPS-M2M 更适合于求解目标不平衡和多样性困难的约束函数的 CMOP，而 PPS-MOEA/D 适用于解决有着不可行区域的 LIR-CMOPs。

有许多方法可以提高 PPS-M2M 的性能，包括增强拉式搜索阶段的约束处理机制，以及集成机器学习方法以将计算资源动态分配到 PPS-M2M 方法的子种群等。值得注意的是，由于 PPS 是一个非常强大的解决 CMOP 的框架，因此可以生成和定制 PPS 框架的许多新实例，以解决具有不同专长的 CMOP。这也是我们未来的研究方向之一。 此外，还将利用所提出的 PPS-M2M 来研究现实世界的优化问题。

第 5 章 总结

在现实生活中，这些多目标优化问题往往都是伴随大量的约束条件的，这使得本来就求解困难的多目标优化问题变得难上加难。约束多目标优化问题的求解从被定义出来，就广泛受到大量国内外研究者广泛研究和关注。对于求解多目标优化问题，传统经典的优化方法往往对问题类型有着严苛的要求，同时它们通常不具有较强的鲁棒性，并且很容易陷入局部最优。当多目标优化问题与约束结合时，问题类型更加多变，从而使得求解过程更加复杂。基于这样的背景，本文主要完成了以下两项工作：

第一，本文通过先总结出前人研究群体聚合形态的局限性，即大多数现有的用于群体聚合形态生成的基因调控网络(Genetic Regulatory Networks， GRN)的设计取决于设计人员的先验知识。为了适应复杂和动态的环境，GRN 的设计通常是反复试验的过程。然后，针对在围捕任务时的群体聚合形态的特点，一个带约束的多目标问题被设计出来。为了克服前人研究群体聚合形态的局限性和解决这个带约束的多目标问题，我们设计出一个用于 GRN 的自动化设计框架，该框架可以场景不同，从而自动生成用于群体聚合形态的 GRN。值得注意的是，所提出的框架应用了一种复合多目标基因编程算法(Composite Multi-objective Genetic Programming，CMOGP)，该方法将多目标基因编程(Multi-objective Genetic Programming，MOGP)、非支配排序遗传算法(Nondominated Sorting Genetic Algorithm II，NSGA-II)、差分进化算法(Differential Evolution Algorithm， DE)和多目标决策(Multi-objective Decision Approach)相结合。此外，所提出的框架应用了一些基本的网络元件来自动设计 GRN，而无需任何先验知识。在 GRNs 的设计过程中，MOGP 和 DE 用于并行优化 GRNs 的结构和参数。 NSGA-II 用于平衡 GRN 模型的复杂性和性能，从而获得满足当前约束环境的 GRN 模型。采用多目标决策来选择 Pareto 最优解中的所需 GRN 模型。全面的仿真结果表明，提出的框架可以有效地开发一些新颖的 GRN，其结构要比人类专家的设计更简单，并且对复杂和动态的环境更加稳定。

第二，本文通过研究前人在处理约束多目标优化问题的成果发现多目标进化算法的一个关键问题是平衡工作种群的收敛性和多样性。然而，大多数最先进的多目标进化算法在平衡它们方面表现不佳，并可能导致工作种群集中在部分 Pareto 前沿，导致在保持多样性和实现收敛之间严重的不平衡搜索。为此，本文提出了一种将

M2M 分解方法与 PPS 框架相结合的方法，即 PPS-M2M。更具体地说，该算法将一个 CMOP 分解成一组简单的 CMOPs。每个简单的 CMOP 对应于一个工作子种群，并以协作的方式解决。在处理约束时，每个子种群都遵循 PPS 搜索准则，使每个工作子种群跨越不可行区。为了评价该算法的性能，将其与 CM2M、MOEA/D-Epsilon、MOEA/D-SR、MOEA/D-CDP、C-MOEA/D、NSGA-II-CDP、CM2M2 等 7 种算法在一组 LIR-CMOPs 测试集上进行了比较。实验结果表明，该算法明显优于其他 7 种算法。此外，本文还提出了一组约束不平衡多目标优化问题(Constrained and Imbalanced Multi-objective Optimization Problems， CIMOPs) 来比较 PPS-M2M 和 PPS-MOEA/D 的性能。实验结果表明，所提出的 PPS-M2M 算法在 CIMOPs 上的性能优于 PPS-MOEA/D 算法。

论文的创新点在于：(1)针对多目标优化问题时，文中提出了一种新的进化计算方法——复合多目标基因编程算法(CMOGP)，并将这种方法用到群体聚合形态生成上。该方法不需要设计人员的先验经验，也不需要使用试错法，就能自动生成符合当前时变环境需求的群体聚合形态。(2)针对带约束多目标优化问题时，本文提出一种新的多目标约束进化算法——PPS-M2M，该方法能很好处理大不可行区域和不平衡目标且约束函数有着多样性困难的 CMOPs，并且在实验中也取得了不错的效果，为解决约束优化问题提供了新的算法工具。

参考文献

[1] Darwin C. On the origin of species by means of natural selection, or the preservation of favored races in the struggle for life[J]. London, 1859, 1: 859.

[2] Floudas C A, Pardalos P M. A collection of test problems for constrained global optimization algorithms[M]. Springer Science & Business Media, 1990.

[3] 李智勇, 黄滔, 陈少淼等. 约束优化进化算法综述[J]. Journal of Software, 2017, 28(6):1-14.

[4] 张纪元. 梯度下降法[C]// 中国数学会全国最优化数值方法学术会. 1987.

[5] 李庆扬, 王能超, 易大义. 数值分析[M]. 北京: 清华大学出版社, 2008.

[6] Glover F. Tabu Search: A Tutorial[J]. Interfaces, 1990, 20(4):74-94.

[7] Kirkpatrick S, Vecchi M P. Optimization by simulated annealing[M]// Readings in computer vision: issues, problems, principles, and paradigms. Morgan Kaufmann Publishers Inc. 1987:339-348.

[8] PARETO V, COURS D. Economics (Volume I and II) [M] .Lausanne: Rouge F, 1896.

[9] Tjalling C. Koopmans. Efficient Allocation of Resources[J]. Econometrica, 1949, 19(4):455-465.

[10] W. Vent. Rechenberg, Ingo, Evolutionsstrategie — Optimierung technischer Systeme nach Prinzipien der biologischen Evolution. 170 S. mit 36 Abb. Frommann-Holzboog-Verlag. Stuttgart 1973. Broschiert[J]. Feddes Repertorium, 1973, 86(5):337-337.

[11] Fogel L J, Owens A J, Walsh M J. Artificial intelligence through simulated evolution[J]. 1966.

[12] Holland J H. Adaptation in Natural and Artificial Systems: An Introductory Analysis With Applications to Biology, Control, and Artificial Intelligence[J]. Quarterly Review of Biology, 1975, 6(2):126–137.

[13] Goldberg D E. Genetic algorithms in search, optimization, and machine learning [J]. Boston: Addison-Wesley, 1989.

[14] Fonseca C M, Fleming P J. Genetic Algorithms for Multiobjective

Optimization: Formulation Discussion and Generalization [C]// ICGA. 1993, 93: 416-423.

[15] Srinivas N, Deb K. Muiltiobjective optimization using nondominated sorting in genetic algorithms [J]. Evolutionary computation, 1994, 2(3): 221-248.

[16] Horn J, Nafpliotis N, Goldberg D E. A niched Pareto genetic algorithm for multiobjective optimization [C]// Evolutionary Computation, 1994. IEEE World Congress on Computational Intelligence., Proceedings of the First IEEE Conference on. 1994: 82-87.

[17] Zitzler E, Thiele L. Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach [J]. Evolutionary computation, IEEE transactions on, 1999, 3(4): 257-271.

[18] Corne D W, Knowles J D, Oates M J. The Pareto envelope-based selection algorithm for multiobjective optimization [C]// Parallel Problem Solving from Nature PPSN VI. Springer Berlin Heidelberg, 2000: 839-848.

[19] Knowles J D, Corne D W. Approximating the Nondominated Front Using the Pareto Archived Evolution Strategy[M]. MIT Press, 2000.

[20] Zitzler E, Laumanns M. SPEA2: Improving the strength Pareto evolutionary algorithm [J]. 2001.

[21] Deb K, Pratap A, Agarwal S, et al. A fast and elitist multiobjective genetic algorithm: NSGA-II [J]. Evolutionary Computation, IEEE Transactions on, 2002, 6(2): 182-197.

[22] Erickson M, Mayer A, Horn J. The niched pareto genetic algorithm 2 applied to the design of groundwater remediation systems [C]// Evolutionary Multi-Criterion Optimization. Springer Berlin Heidelberg, 2001: 681-695.

[23] Koza J R. Genetic programming: on the programming of computers by means of natural selection[M]. MIT Press, 1992.

[24] Yu X, Gen M. Introduction to evolutionary algorithms [M]. Springer Science & Business Media, 2010.

[25] Zitzler E, Künzli S. Indicator-based selection in multiobjective search[C]//International Conference on Parallel Problem Solving from Nature. Springer, Berlin, Heidelberg, 2004: 832-842.

[26] Sierra M R, Coello C A C. Improving PSO-based multi-objective optimization using crowding, mutation and ε-dominance [C]// Evolutionary

Multi-Criterion Optimization. Springer Berlin Heidelberg, 2005: 505-519.

[27] Zhang Q, Li H. MOEA/D: A multiobjective evolutionary algorithm based on decomposition [J]. Evolutionary Computation, IEEE Transactions on, 2007, 11(6): 712-731.

[28] 郑金华. 多目标进化算法及其应用[M]. 科学出版社, 2007.

[29] Deb K, Jain H. An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point-Based Nondominated Sorting Approach, Part I: Solving Problems with Box Constraints[J]. Evolutionary Computation, IEEE Transactions on, 2014, 18(4):577-601.

[30] Seada H, Deb K. U-NSGA-III: a unified evolutionary optimization procedure for single, multiple, and many objectives: proof-of-principle results[C]//International conference on evolutionary multi-criterion optimization. Springer, Cham, 2015: 34-49.

[31] Kumar M, Guria C. The elitist non-dominated sorting genetic algorithm with inheritance (i-NSGA-II) and its jumping gene adaptations for multi-objective optimization[J]. Information sciences, 2017, 382: 15-37.

[32] Lu Z , Whalen I , Boddeti V , et al. NSGA-NET: A Multi-Objective Genetic Algorithm for Neural Architecture Search[J]. 2018.

[33] Li H, Zhang Q. Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II[J]. IEEE Transactions on evolutionary computation, 2009, 13(2): 284-302.

[34] Fan Z , Fang Y , Li W , et al. MOEA/D with angle-based constrained dominance principle for constrained multi-objective optimization problems[J]. Applied Soft Computing, 2018.

[35] Cai X, Li Y, Fan Z, et al. An external archive guided multiobjective evolutionary algorithm based on decomposition for combinatorial optimization[J]. IEEE Transactions on Evolutionary Computation, 2015, 19(4): 508-523.

[36] Cai X, Yang Z, Fan Z, et al. Decomposition-based-sorting and angle-based-selection for evolutionary multiobjective and many-objective optimization[J]. IEEE transactions on cybernetics, 2017, 47(9): 2824-2837.

[37] Li H, He F, Yan X. IBEA-SVM: an indicator-based evolutionary algorithm based on pre-selection with classification guided by SVM[J]. Applied Mathematics-A Journal of Chinese Universities, 2019, 34(1): 1-26.

[38] Zitzler, Eckart, and Simon Künzli. "Indicator-based selection in multiobjective search." International conference on parallel problem solving from nature. Springer, Berlin, Heidelberg, 2004.

[39] Li W, Özcan E, John R, et al. A modified indicator-based evolutionary algorithm (mIBEA)[C]//2017 IEEE Congress on Evolutionary Computation (CEC). IEEE, 2017: 1047-1054.

[40] Bader, Johannes, and Eckart Zitzler. "HypE: An algorithm for fast hypervolume-based many-objective optimization." Evolutionary computation 19.1 (2011): 45-76.

[41] Dorigo M. Ottimizzazione, apprendimento automatico, ed algoritmi basati su metafora naturale[J]. Unpublished doctoral dissertation, Politecnico di Milano, Italy, 1992.

[42] Kennedy J, Eberhart R. Particle swarm optimization[J]. Proc. of 1995 IEEE Int. Conf. Neural Networks, (Perth, Australia), Nov. 27-Dec. 2011, 4(8):1942-1948 vol.4.

[43] Storn R, Price K. Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces[J]. Journal of Global Optimization, 1997, 11(4):341-359.

[44] Moscato P. On Evolution, Search, Optimization, Genetic Algorithms and Martial Arts : Towards Memetic Algorithms[J]. Caltech Concurrent Computation Program, 1989.

[45] Castro L R D, Timmis J. Artificial Immune Systems: A New Computational Intelligence Paradigm[M]. Springer-Verlag New York, Inc. 2002.

[46] Coello C A C. Evolutionary multi-objective optimization: a historical view of the field[J]. IEEE computational intelligence magazine, 2006, 1(1): 28-36.

[47] Van Veldhuizen D A, Lamont G B. Multiobjective evolutionary algorithm research: A history and analysis[R]. Technical Report TR-98-03, Department of Electrical and Computer Engineering, Graduate School of Engineering, Air Force Institute of Technology, Wright-Patterson AFB, Ohio, 1998.

[48] Ishibuchi H, Tsukamoto N, Nojima Y. Evolutionary many-objective optimization: A short review[C]//Evolutionary Computation, 2008. CEC 2008.(IEEE World Congress on Computational Intelligence). IEEE Congress on. IEEE, 2008: 2419-2426.

[49] Kaisa Miettinen. Nonlinear Multiobjective Optimization[M]. Birkhaüser

Verlag, 1998.

[50] Messac A , Ismail-Yahaya A , Mattson C A . The normalized normal constraint method for generating the Pareto frontier[J]. Structural & Multidisciplinary Optimization, 2003, 25(2):86-98.

[51] Coello, Carlos A. Coello, and Margarita Reyes Sierra. "Multiobjective evolutionary algorithms: classifications, analyses, and new innovations." Evolutionary Computation. 1999.

[52] Coello, Coello, Reyes, Sierra M . A Study of the Parallelization of a Coevolutionary Multi-objective Evolutionary Algorithm[C]// Proceedings of the Third Mexican International Conference on Artificial Intelligence (MICAI'2004). 2004.

[53] While L, Hingston P, Barone L, et al. A faster algorithm for calculating hypervolume[J]. IEEE transactions on evolutionary computation, 2006, 10(1): 29-38.

[54] Powell, Michael JD. "The convergence of variable metric methods for nonlinearly constrained optimization calculations." Nonlinear programming 3. Academic Press, 1978: 27-63.

[55] Mitra, Subhasish, Nirmal R. Saxena, and Edward J. McCluskey. "A design diversity metric and analysis of redundant systems." IEEE Transactions on Computers 2002, 51(5): 498-510.

[56] Ernst, Frederick J. "Removal of the nodal singularity of the C - metric." Journal of Mathematical Physics 1976, 17(4): 515-516.

[57] Yu G, Jin Y , Olhofer M . Benchmark Problems and Performance Indicators for Search of Knee Points in Multiobjective Optimization[J]. IEEE Transactions on Cybernetics, 2019:1-14.

[58] Wolpert D H, Macready W G. No free lunch theorems for optimization[J]. IEEE transactions on evolutionary computation, 1997, 1(1): 67-82.

[59] Zhao, Huimin. "A multi-objective genetic programming approach to developing Pareto optimal decision trees." Decision Support Systems 2007, 43(3): 809-826.

[60] Brauers W K M, Zavadskas E K, Peldschus F, et al. Multi-objective decision-making for road design[J]. Transport, 2008, 23(3): 183-193.

[61] Koza J R, Koza J R. Genetic programming: on the programming of computers by means of natural selection[M]. MIT press, 1992.

[62] Brameier M F, Banzhaf W. Linear genetic programming[M]. Springer

Science & Business Media, 2007.

[63] Langdon W B. Genetic programming and data structures: genetic programming+ data structures=automatic programming! [M]. Springer Science & Business Media, 2012.

[64] Langdon W B. Size fair and homologous tree genetic programming acrossovers[C]//Proceedings of the 1st Annual Conference on Genetic and Evolutionary Computation-Volume 2. Morgan Kaufmann Publishers Inc., 1999: 1092-1097.

[65] Oh H, Shirazi A R, Sun C, et al. Bio-inspired self-organizing multi-robot pattern generation: A review[J]. Robotics and Autonomous Systems, 2017, 91: 83-100.

[66] Li J, Tan Y. A two-stage imitation learning framework for the multi-target search problem in swarm robotics[J]. Neurocomputing, 2019, 334: 249-264.

[67] Bakhshipour M, Ghadi M J, Namdari F. Swarm robotics search & rescue: A novel artificial intelligence-inspired optimization approach[J]. Applied Soft Computing, 2017, 57: 708-726.

[68] Levchenko, M. Keidar, J. Cantrell, Y.-L. Wu, H. Kuninaka, K. Bazaka, S. Xu, Explore space using swarms of tiny satellites, Nature, 2018, 185–187.

[69] Bezouska W A, Barnhart D A. Visual sensor selection for satellite swarm cooperative localization[C]//Sensors and Systems for Space Applications XII. International Society for Optics and Photonics, 2019, 11017: 1101705.

[70] Vásárhelyi G, Virágh C, Somorjai G, et al. Optimized flocking of autonomous drones in confined environments[J]. Science Robotics, 2018, 3(20): eaat3536.

[71] Xie H, Sun M, Fan X, et al. Reconfigurable magnetic microrobot swarm: Multimode transformation, locomotion, and manipulation[J]. Science Robotics, 2019, 4(28): eaav8006.

[72] Feng Q, Zhang C, Lu J, et al. Source localization in dynamic indoor environments with natural ventilation: An experimental study of a particle swarm optimization-based multi-robot olfaction method[J]. Building and Environment, 2019: 106228.

[73] Amjadi A S, Raoufi M, Turgut A E, et al. Cooperative pollution source localization and cleanup with a bio-inspired swarm robot aggregation[J]. arXiv preprint arXiv:1907.09585, 2019.

[74] Mamei M, Vasirani M, Zambonelli F. Experiments of morphogenesis in

swarms of simple mobile robots[J]. Applied Artificial Intelligence, 2004, 18(9-10): 903-919.

[75] Kondacs A. Biologically-inspired self-assembly of two-dimensional shapes using global-to-local compilation[C]//Proceedings of the 18th international joint conference on Artificial intelligence. Morgan Kaufmann Publishers Inc., 2003: 633-638.

[76] Jin Y, Meng Y. Morphogenetic robotics: An emerging new field in developmental robotics[J]. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 2010, 41(2): 145-160.

[77] Slavkov I, Carrillo-Zapata D, Carranza N, et al. Morphogenesis in robot swarms[J]. Science Robotics, 2018, 3(25): eaau9178.

[78] Kondo S, Miura T. Reaction-diffusion model as a framework for understanding biological pattern generation[J]. Science, 2010, 329(5999): 1616-1620.

[79] Bai L, Eyiyurekli M, Breen D E. An emergent system for self-aligning and self-organizing shape primitives[C]//2008 Second IEEE International Conference on Self-Adaptive and Self-Organizing Systems. IEEE, 2008: 445-454.

[80] Cussat-Blanc S, Harrington K, Banzhaf W. Artificial Gene Regulatory Networks—A Review[J]. Artificial life, 2019, 24(4): 296-328.

[81] Jin Y, Guo H, Meng Y. A hierarchical gene regulatory network for adaptive multirobot pattern generation[J]. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 2012, 42(3): 805-816.

[82] Peng X, Zhang S, Lei X. Multi-target trapping in constrained environments using gene regulatory network-based pattern formation[J]. International Journal of Advanced Robotic Systems, 2016, 13(5): 1729881416670152.

[83] H. Oh, Y. Jin, Evolving hierarchical gene regulatory networks for morphogenetic pattern generation of swarm robots, in: 2014 IEEE Congress on Evolutionary Computation (CEC), IEEE, 2014, pp. 776–783.

[84] Milo R, Shen-Orr S, Itzkovitz S, et al. Network motifs: simple building blocks of complex networks[J]. Science, 2002, 298(5594): 824-827.

[85] Bowers P M, Cokus S J, Eisenberg D, et al. Use of logic relationships to decipher protein network organization[J]. Science, 2004, 306(5705): 2246-2249.

[86] Hansen N, Ostermeier A. Completely derandomized self-adaptation in evolution strategies[J]. Evolutionary computation, 2001, 9(2): 159-195.

[87] Eiben, A.E, Smith, J.E. [Natural Computing Series] Introduction to

Evolutionary Computing || Interactive Evolutionary Algorithms[J]. 10.1007/978-3-662-44874-8(Chapter 14):215-222.

[88] Kennedy J. Swarm Intelligence[M]// Swarm intelligence. 2001.

[89] Michalewicz. How to Solve It: Modern Heuristics[M]. 2004.

[90] Bèack T. Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms / T. Bäck. [M]. 1998.

[91] Engelbrecht A P. Fundamentals of Computational Swarm Intelligence[M]. 2006.

[92] C.A.C. Coello, Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art, Comput. Meth. Appl. Mech. Eng. 2002,191(11–12):1245–1287.

[93] T. BDack, F. Hoffffmeister, H. Schwefel, A survey of evolution strategies, in: Proceedings of the 4th International Conference on Genetic Algorithms, 1991, pp. 2–9.

[94] Homaifar, Abdollah, Charlene X. Qi, and Steven H. Lai. "Constrained optimization via genetic algorithms." Simulation 62.4 (1994): 242-253.

[95] Joines J A, Houck C R. On the use of non-stationary penalty functions to solve nonlinear constrained optimization problems with GA's[C]// IEEE Conference on Evolutionary Computation, IEEE World Congress on Computational Intelligence. IEEE, 1994.

[96] Bean, James C., and Atidel ben Hadj-Alouane. A dual genetic algorithm for bounded integer programs. 1993.

[97] Coit, David W, Smith, Alice E, Tate, David M. Adaptive Penalty Methods for Genetic Optimization of Constrained Combinatorial Problems[J]. Informs Journal on Computing, 1996, 8(2):173-182.

[98] Hadj-Alouane A B, Bean J C. A Genetic Algorithm for the Multiple-Choice Integer Program[J]. Operations Research, 1997, 45(1):92-101.

[99] Woldesenbet Y G, Yen G G, Tessema B G. Constraint Handling in Multiobjective Evolutionary Optimization[C]// IEEE, 2009: 514-525.

[100] Takahama, Tetsuyuki, and Setsuko Sakai. "Constrained optimization by ε constrained particle swarm optimizer with ε-level control." Soft computing as transdisciplinary science and technology. Springer, Berlin, Heidelberg, 2005: 1019-1029.

[101] Runarsson, Thomas P., and Xin Yao. "Stochastic ranking for constrained

evolutionary optimization." IEEE Transactions on evolutionary computation 2000, 4(3): 284-294.

[102] Wang Y, Cai Z, Zhou Y, et al. An Adaptive Tradeoff Model for Constrained Evolutionary Optimization[J]. IEEE transactions on evolutionary computation, 2008, 12(1):80-92.

[103] Qu B Y, Suganthan P N. Constrained multi-objective optimization algorithm with an ensemble of constraint handling methods[J]. Engineering Optimization, 2011, 43(4):403-416.

[104] Fan, Zhun, et al. "An improved epsilon constraint-handling method in MOEA/D for CMOPs with large infeasible regions." Soft Computing 2019, 23: 12491-12510.

[105] Liu, Hai-Lin, Gu, Fangqing, Zhang, Qingfu. Decomposition of a Multiobjective Optimization Problem Into a Number of Simple Multiobjective Subproblems[J]. IEEE Transactions on Evolutionary Computation, 2019, 18(3):450-455.

[106] Fan, Zhun, et al. "Push and pull search for solving constrained multi-objective optimization problems." Swarm and evolutionary computation 2019, 44: 665-679.

[107] Liu H L, Peng C, Gu F, et al. A Constrained Multi-Objective Evolutionary Algorithm Based on Boundary Search and Archive[J]. International Journal of Pattern Recognition & Artificial Intelligence, 2015, 30(1):1-9.

[108] Jan M A, Khanum R A. A study of two penalty-parameterless constraint handling techniques in the framework of MOEA/D[J]. Applied Soft Computing, 2013, 13(1):128-148.

[109] Asafuddoula M, Ray T, Sarker R, et al. An adaptive constraint handling approach embedded MOEA/D[C]// Evolutionary Computation (CEC), 2012 IEEE Congress on. IEEE, 2012.

[110] Peng C, Liu H L, Gu F. An evolutionary algorithm with directed weights for constrained multi-objective optimization[J]. Applied Soft Computing, 2017, 60.

[111] Huband S, Hingston P, Barone L, et al. A review of multiobjective test problems and a scalable test problem toolkit[J]. IEEE Transactions on Evolutionary Computation, 2006, 10(5): 477-506.

[112] Fan, Zhun, et al. "Difficulty adjustable and scalable constrained multiobjective test problem toolkit." Evolutionary computation 2019: 1-40.

攻读硕士学位期间主要的工作成果

一、发表的学术论文

(一)已见刊或录用论文

[1] Zhun Fan, Yi Fang, Wenji Li, Yutong Yuan, Zhaojun Wang and Xinchao Bian, LSHADE44 with an Improved Epsilom Constraint-handling Method for Solving Constrained Single-objective Optimization Problems[C]// Evolutionary Computation (CEC), 2018 IEEE Congress on. IEEE, 2018.

[2] Zhun Fan, Zhaojun Wang, Yi Fang, Wenji Li, Yutong Yuan, Xinchao Bian. "Adaptive Recombination Operator Selection in Push and Pull Search for Solving Constrained Single-Objective Optimization Problems." International Conference on Bio-Inspired Computing: Theories and Applications. Springer, Singapore, 2018.

[3] Yutong Yuan, Zhun Fan, Xiaomin Zhu, Meng Wu, Li Ma, Taosheng Fang, Zhaojun Wang, Weidong Bao, Yun Zhou, Huangke Chen, and Wenji Li. "TH-GRN Model Based Collective Tracking in Confined Environment." *International Conference on Swarm Intelligence*. Springer, Cham, 2019.

[4] Zhun Fan, Zhaojun Wang, Wenji Li, Yutong Yuan, Yugen You, Zhi Yang, Fuzan Sun and Jie Ruan. "Push and pull search embedded in an M2M framework for solving constrained multi-objective optimization problems." Swarm and Evolutionary Computation 54 (2020): 100651.

二、发表软件著作权

[1] 机器人设计自动化平台 V1.0，登记号：2018SR212300.

致谢

时光穿梭，转眼间已是我在汕头大学待过的第三个年头了。此时的校园正值万物欣然，金凤花开的时刻，正预示着我在这里三年的研究生生涯即将走向结束。回望过往，我欣喜于我在汕头大学得到了许多学问、见识及成长，而这些都得益于我的母校、我的导师、我的同学还有其他所有给予过我帮助的人。我竭诚感恩，因为有了你们一直以来的支持、鼓励与陪伴，使我受益良多，我莫不敢忘。

首先我要感谢我的导师，尊敬的范衡教授。学术上您是一位学识渊博、治学严谨、视野开拓的学者，是您在学业上对我进行了悉心的指导，也是您引领我走进神圣的科研殿堂。工作中您求真务实，踏实勤奋，您的作风深深地感染了我，您是我学习的榜样。生活中您不仅是一名良师，同时也是一位乐于关心倾听学生的益友，您常常以您丰富的人生经验给我带来许多宝贵的意见和帮助，使我受益匪浅。没有您无私的付出，也不会有我这三年里这么快速的成长。在此我由衷地感谢您，我的导师范衡老师。

其次我要感谢在汕头大学学习的这段时间里为我传道授业的崔岩老师、陈耀文老师、姜永权老师、陈力老师、李旭涛老师、闫敬文老师、庄哲民老师、唐雅娟老师、魏楚亮老师、蔡泽民老师、赖李洋老师、袁野老师和其他电子系的老师们，谢谢您们教授了我专业方面的知识，并给予了我许多生活上的关心和帮助。

然后我要感谢人工智能与机器人实验室的李文姬师兄、朱贵杰师兄和游煜根师兄，谢谢你们在科研上学术调研、论文撰写、专利申请等的指导，也谢谢你们在生活中给我带来了关心，你们始终都是值得我学习的兄长。感谢黄龙涛、袁宇彤、卞新超、姜涛、安康同学，你们不仅在学习上给我带来了帮助，也是我生活中的挚友。我还要感谢杨知、孙福赞、陈梓泓、龙周彬、马培立、林培涵、李晓明等可爱的师弟师妹们，学习上我们不仅经常在一起进行讨论，生活中我们也成为了无话不谈的朋友。

最后，我要感谢我的家人。这么多年来我的父母含辛茹苦地抚育了我。您们始终以我为骄傲，并用最无私的爱默默地关怀支持着我，让我能够毫无后顾之忧地完成我的硕士学业。所以在这里对于我的家人，我由衷地谢谢您们。

作者：王诏君

2020 年 06 月 02 日

附录

附表 I　测试问题 LIR-CMOP1-14 目标函数及其约束条件

编号	目标函数	约束条件
1	$f_1(x)=x_1+g_1(x)$ $f_2(x)=1-x_1^2+g_2(x)$ $g_1(x)=\sum_{j\in J_1}(x_j-sin(0.5\pi x_1))^2$ $g_2(x)=\sum_{j\in J_2}(x_j-cos(0.5\pi x_1))^2$ $J_1=\{3,5,...,29\},J_2=\{2,4,...,30\}$	$c_1(x)=(a-g_1(x))(g_1(x)-b)\geq 0$ $c_2(x)=(a-g_2(x))(g_2(x)-b)\geq 0$ $a=0.51,b=0.5$ $x\in[0,1]^{30}$
2	$f_1(x)=x_1+g_1(x)$ $f_2(x)=1-\sqrt{x_1}+g_2(x)$ $g_1(x)=\sum_{j\in J_1}(x_j-sin(0.5\pi x_1))^2$ $g_2(x)=\sum_{j\in J_2}(x_j-cos(0.5\pi x_1))^2$ $J_1=\{3,5,...,29\},J_2=\{2,4,...,30\}$	$c_1(x)=(a-g_1(x))(g_1(x)-b)\geq 0$ $c_2(x)=(a-g_2(x))(g_2(x)-b)\geq 0$ $a=0.51,b=0.5$ $x\in[0,1]^{30}$
3	$f_1(x)=x_1+g_1(x)$ $f_2(x)=1-x_1^2+g_2(x)$ $g_1(x)=\sum_{j\in J_1}(x_j-sin(0.5\pi x_1))^2$ $g_2(x)=\sum_{j\in J_2}(x_j-cos(0.5\pi x_1))^2$ $J_1=\{3,5,...,29\},J_2=\{2,4,...,30\}$	$c_1(x)=(a-g_1(x))(g_1(x)-b)\geq 0$ $c_2(x)=(a-g_2(x))(g_2(x)-b)\geq 0$ $c_3(x)=\sin(c\pi x_1)-0.5\geq 0$ $a=0.51,b=0.5,c=20$ $x\in[0,1]^{30}$
4	$f_1(x)=x_1+g_1(x)$ $f_2(x)=1-\sqrt{x_1}+g_2(x)$ $g_1(x)=\sum_{j\in J_1}(x_j-sin(0.5\pi x_1))^2$ $g_2(x)=\sum_{j\in J_2}(x_j-cos(0.5\pi x_1))^2$ $J_1=\{3,5,...,29\},J_2=\{2,4,...,30\}$	$c_1(x)=(a-g_1(x))(g_1(x)-b)\geq 0$ $c_2(x)=(a-g_2(x))(g_2(x)-b)\geq 0$ $c_3(x)=\sin(c\pi x_1)-0.5\geq 0$ $a=0.51,b=0.5,c=20$ $x\in[0,1]^{30}$
5	$f_1(x)=x_1+10g_1(x)+0.7057$ $f_2(x)=1-\sqrt{x_1}+10g_2(x)+0.7057$ $g_1(x)=\sum_{i\in J_1}\left(x_i-sin(\frac{0.5i}{30}\pi x_1\right))^2$ $g_2(x)=\sum_{j\in J_2}(x_j-cos(\frac{0.5j}{30}\pi x_1))^2$ $J_1=\{3,5,...,29\},J_2=\{2,4,...,30\}$	$c_k(x)=((f_1-p_k)cos\theta_k-(f_2-q_k)$ $sin\theta_k)^2/a_k^2+((f_1-p_k)sin\theta_k+(f_2$ $-q_k)cos\theta_k)^2/b_k^2\geq r$ $p_k=[1.6,2.5],q_k=[1.6,2.5]$ $a_k=[2,2],b_k=[4,8]$ $r=0.1,\theta_k=-0.25\pi$ $x\in[0,1]^{30},k=1,2$
6	$f_1(x)=x_1+10g_1(x)+0.7057$	$c_k(x)=((f_1-p_k)cos\theta_k-(f_2-q_k)$

	$f_2(x)=1-x_1^2+10g_2(x)+0.7057$ $g_1(x)=\sum_{i\in J_1}\left(x_i-sin(\frac{0.5i}{30}\pi x_1\right))^2$ $g_2(x)=\sum_{j\in J_2}(x_j-cos(\frac{0.5j}{30}\pi x_1))^2$ $J_1=\{3,5,...,29\},J_2=\{2,4,...,30\}$	$sin\theta_k)^2/a_k^2+((f_1-p_k)sin\theta_k+(f_2$ $-q_k)cos\theta_k)^2/b_k^2\geq r$ $p_k=[1.8,2.8],q_k=[1.8,2.8]$ $a_k=[2,2],b_k=[4,8]$ $r=0.1,\theta_k=-0.25\pi$ $x\in[0,1]^{30},k=1,2$
7	$f_1(x)=x_1+10g_1(x)+0.7057$ $f_2(x)=1-\sqrt{x_1}+10g_2(x)+0.7057$ $g_1(x)=\sum_{i\in J_1}\left(x_i-sin(\frac{0.5i}{30}\pi x_1\right))^2$ $g_2(x)=\sum_{j\in J_2}(x_j-cos(\frac{0.5j}{30}\pi x_1))^2$ $J_1=\{3,5,...,29\},J_2=\{2,4,...,30\}$	$c_k(x)=((f_1-p_k)cos\theta_k-(f_2-q_k)$ $sin\theta_k)^2/a_k^2+((f_1-p_k)sin\theta_k+(f_2$ $-q_k)cos\theta_k)^2/b_k^2\geq r$ $p_k=[1.2,2.25,3.5],$ $q_k=[1.2,2.25,3.5]$ $a_k=[2,2.5,2.5],b_k=[6,12,10]$ $r=0.1,\theta_k=-0.25\pi$ $x\in[0,1]^{30},k=1,2,3$
8	$f_1(x)=x_1+10g_1(x)+0.7057$ $f_2(x)=1-x_1^2+10g_2(x)+0.7057$ $g_1(x)=\sum_{i\in J_1}\left(x_i-sin(\frac{0.5i}{30}\pi x_1\right))^2$ $g_2(x)=\sum_{j\in J_2}(x_j-cos(\frac{0.5j}{30}\pi x_1))^2$ $J_1=\{3,5,...,29\},J_2=\{2,4,...,30\}$	$c_k(x)=((f_1-p_k)cos\theta_k-(f_2-q_k)$ $sin\theta_k)^2/a_k^2+((f_1-p_k)sin\theta_k+(f_2$ $-q_k)cos\theta_k)^2/b_k^2\geq r$ $p_k=[1.2,2.25,3.5],$ $q_k=[1.2,2.25,3.5]$ $a_k=[2,2.5,2.5],b_k=[6,12,10]$ $r=0.1,\theta_k=-0.25\pi$ $x\in[0,1]^{30},k=1,2,3$
9	$f_1(x)=1.7057x_1(10g_1(x)+1)$ $f_2(x)=1.7057(1-x_1^2)(10g_2(x)+1)$ $g_1(x)=\sum_{i\in J_1}\left(x_i-sin(\frac{0.5i}{30}\pi x_1\right))^2$ $g_2(x)=\sum_{j\in J_2}\left(x_j-cos(\frac{0.5j}{30}\pi x_1\right))^2$ $J_1=\{3,5,...,29\},J_2=\{2,4,...,30\}$	$c_1(x)=((f_1-p_1)cos\theta_1-(f_2-q_1)$ $sin\theta_1)^2/a_1^2+((f_1-p_1)sin\theta_1+(f_2$ $-q_1)cos\theta_1)^2/b_1^2\geq r$ $c_2(x)=f_1sin\alpha+f_2cos\alpha-$ $sin(4\pi(f_1cos\alpha-f_2sin\alpha))-2\geq 0$ $p_1=1.4,q_1=1.4,a_1=1.5,b_1=6.0$ $r=0.1,\alpha=0.25\pi,\theta_1=-0.25\pi$ $x\in[0,1]^{30}$
10	$f_1(x)=1.7057x_1(10g_1(x)+1)$ $f_2(x)=1.7057(1-\sqrt{x_1})(10g_2(x)+1)$ $g_1(x)=\sum_{i\in J_1}\left(x_i-sin(\frac{0.5i}{30}\pi x_1\right))^2$ $g_2(x)=\sum_{j\in J_2}\left(x_j-cos(\frac{0.5j}{30}\pi x_1\right))^2$ $J_1=\{3,5,...,29\},J_2=\{2,4,...,30\}$	$c_1(x)=((f_1-p_1)cos\theta_1-(f_2-q_1)$ $sin\theta_1)^2/a_1^2+((f_1-p_1)sin\theta_1+(f_2$ $-q_1)cos\theta_1)^2/b_1^2\geq r$ $c_2(x)=f_1sin\alpha+f_2cos\alpha-$ $sin(4\pi(f_1cos\alpha-f_2sin\alpha))-1\geq 0$ $p_1=1.1,q_1=1.2,a_1=2.0,b_1=4.0$ $r=0.1,\alpha=0.25\pi,\theta_1=-0.25\pi$ $x\in[0,1]^{30}$
11	$f_1(x)=1.7057x_1(10g_1(x)+1)$ $f_2(x)=1.7057(1-\sqrt{x_1})(10g_2(x)+1)$	$c_1(x)=((f_1-p_1)cos\theta_1-(f_2-q_1)$ $sin\theta_1)^2/a_1^2+((f_1-p_1)sin\theta_1+(f_2$ $-q_1)cos\theta_1)^2/b_1^2\geq r$

	$g_1(x)=\sum_{i\in J_1}\left(x_i-sin\left(\frac{0.5i}{30}\pi x_1\right)\right)^2$ $g_2(x)=\sum_{j\in J_2}\left(x_j-cos\left(\frac{0.5j}{30}\pi x_1\right)\right)^2$ $J_1=\{3,5,...,29\},J_2=\{2,4,...,30\}$	$c_2(x)=f_1\sin\alpha+f_2\cos\alpha-\sin(4\pi(f_1\cos\alpha-f_2\sin\alpha))-2.1\geq 0$ $p_1=1.2,q_1=1.2,a_1=1.5,b_1=5.0$ $r=0.1,\alpha=0.25\pi,\theta_1=-0.25\pi$ $x\in[0,1]^{30}$
12	$f_1(x)=1.7057x_1(10g_1(x)+1)$ $f_2(x)=1.7057(1-x_1^2)(10g_2(x)+1)$ $g_1(x)=\sum_{i\in J_1}\left(x_i-sin\left(\frac{0.5i}{30}\pi x_1\right)\right)^2$ $g_2(x)=\sum_{j\in J_2}\left(x_j-cos\left(\frac{0.5j}{30}\pi x_1\right)\right)^2$ $J_1=\{3,5,...,29\},J_2=\{2,4,...,30\}$	$c_1(x)=((f_1-p_1)cos\theta_1-(f_2-q_1)sin\theta_1)^2/a_1^2+((f_1-p_1)sin\theta_1+(f_2-q_1)cos\theta_1)^2/b_1^2\geq r$ $c_2(x)=f_1\sin\alpha+f_2\cos\alpha-\sin(4\pi(f_1\cos\alpha-f_2\sin\alpha))-2.5\geq 0$ $p_1=1.6,q_1=1.6,a_1=1.5,b_1=6.0$ $r=0.1,\alpha=0.25\pi,\theta_1=-0.25\pi$ $x\in[0,1]^{30}$
13	$f_1(x)=(1.7057+g_1)\,cos(0.5\pi x_1)\,cos(0.5\pi x_2)$ $f_2(x)=(1.7057+g_1)\,cos(0.5\pi x_1)\,sin(0.5\pi x_2)$ $f_3(x)=(1.7057+g_1)\,sin(0.5\pi x_1)$ $g_1=\sum_{i\in J}10(x_i-0.5)^2$ $J=\{3,4,...,30\}$	$c_1(x)=(g(x)-9)(g(x)-4)$ $c_2(x)=(g(x)-3.61)(g(x)-3.24)$ $g(x)=f_1^2+f_2^2+f_3^2$ $x\in[0,1]^{30}$
14	$f_1(x)=(1.7057+g_1)\,cos(0.5\pi x_1)\,cos(0.5\pi x_2)$ $f_2(x)=(1.7057+g_1)\,cos(0.5\pi x_1)\,sin(0.5\pi x_2)$ $f_3(x)=(1.7057+g_1)\,sin(0.5\pi x_1)$ $g_1=\sum_{i\in J}10(x_i-0.5)^2$ $J=\{3,4,...,30\}$	$c_1(x)=(g(x)-9)(g(x)-4)$ $c_2(x)=(g(x)-3.61)(g(x)-3.24)$ $c_3(x)=(g(x)-3.0625)(g(x)-2.56)$ $g(x)=f_1^2+f_2^2+f_3^2$ $x\in[0,1]^{30}$

附表 II　测试问题 CIMOP1-7 目标函数及其约束条件

编号	目标函数	约束条件
1	$f_1(x)=x_1(1+g(x))$ $f_2(x)=(1+g(x))(1-\sqrt{x_1})$ $g(x)=2\sin(\pi x_1)\sum_{i=2}^{n}(-0.9t_i^2+\lvert t_i^{0.6}\rvert)$ $t_i=x_i-\sin(0.5\pi x_1)$ $n=10,x\in[0,1]^{10}$	$c_1(x)=sin(a\pi x_1)-b\geq 0$ $a=20,b=0$

2	$f_1(x) = x_1(1+g(x))$ $f_2(x) = (1+g(x))(1-x_1^2)$ $g(x) = 10sin(\pi x_1)\sum_{i=2}^{n}\frac{\lvert t_i\rvert}{1+e^{5\lvert t_i\rvert}}$ $t_i = x_i - \sin(0.5\pi x_1)$ $n = 10, x \in [0,1]^{10}$	$c_1(x) = sin(a\pi x_1) - b \geq 0$ $a = 20, b = 0$
3	$f_1(x) = (1+g(x))\,cos\frac{\pi x_1}{2}$ $f_2(x) = (1+g(x))sin\frac{\pi x_1}{2}$ $g(x) = 10sin(\frac{\pi x_1}{2})\sum_{i=2}^{n}\frac{\lvert t_i\rvert}{1+e^{5\lvert t_i\rvert}}$ $t_i = x_i - \sin(0.5\pi x_1)$ $n = 10, x \in [0,1]^{10}$	$c_1(x) = sin(a\pi x_1) - b \geq 0$ $a = 20, b = 0$
4	$f_1(x) = x_1(1+g(x))$ $f_2(x) = (1+g(x))(1-x_1^{0.5}cos^2(2\pi x_1))$ $g(x) = 1 + 10sin(\pi x_1)\sum_{i=2}^{n}\frac{\lvert t_i\rvert}{1+e^{5\lvert t_i\rvert}}$ $t_i = x_i - \sin(0.5\pi x_1)$ $n = 10, x \in [0,1]^{10}$	$c_1(x) = sin(a\pi x_1) - b \geq 0$ $a = 20, b = 0$
5	$f_1(x) = x_1(1+g(x))$ $f_2(x) = (1+g(x))(1-\sqrt{x_1})$ $g(x) = 2\lvert\cos(\pi x_1)\rvert\sum_{i=2}^{n}(-0.9t_i^2 + \lvert t_i^{0.6}\rvert)$ $t_i = x_i - \sin(0.5\pi x_1)$ $n = 10, x \in [0,1]^{10}$	$c_1(x) = sin(a\pi x_1) - b \geq 0$ $a = 20, b = 0$
6	$f_1(x) = x_1x_2(1+g(x))$ $f_2(x) = x_1(1+g(x))(1-x_2)$ $f_3(x) = (1+g(x))(1-x_1)$ $g(x) = 2\sin(\pi x_1)\sum_{i=2}^{n}(-0.9t_i^2 + \lvert t_i^{0.6}\rvert)$ $t_i = x_i - x_1x_2$ $n = 10, x \in [0,1]^{10}$	$c_1(x) = sin(a\pi x_1) - b \geq 0$ $c_2(x) = cos(a\pi x_2) - b \geq 0$ $a = 20, b = 0$
7	$f_1(x) = (1+g(x))\,cos\frac{\pi x_1}{2}cos\frac{\pi x_2}{2}$	$c_1(x) = sin(a\pi x_1) - b \geq 0$ $c_2(x) = cos(a\pi x_2) - b \geq 0$ $a = 20, b = 0$

	$f_2(x) = (1 + g(x))cos\frac{\pi x_1}{2}\ sin\frac{\pi x_2}{2}$ $f_3(x) = (1 + g(x))sin\frac{\pi x_2}{2}$ $g(x) = 2\sin(\pi x_1)\sum_{i=2}^{n}(-0.9t_i^2 + \lvert t_i^{0.6}\rvert)$ $t_i = x_i - x_1x_2$ $n = 10, x \in [0,1]^{10}$	