第5章

生物群落演化模式与优化算法

引　　言

本章对自然界生物群落中具有不同种群关系(包括竞争、捕食、互利、共存和寄生关系)的生物群落演化模式进行建模与仿真研究,并在此基础上设计了新型的基于多群体协同进化的生物启发计算算法,分别是基于共栖关系的多群体协同粒子群优化 MSPSO-C、基于寄生关系的多群体协同粒子群优化 MSPSO-P、基于寄生关系的多群体协同粒子群优化 MOPSO-M 和多种群多目标人工蜂群算法。本章对这些算法的性能进行了实验比较,并将其应用于实际工程应用中的 RFID 网络优化问题和多目标问题。

5.1　生物群落进化中的种群演化模式

生态系统是指生物群落与其无机环境之间由于不断进行物质循环和能量交换而形成的统一整体[1]。而其中生活在一定空间内相互有直接或间接关系的有关种群的总体称为生物群落,即在自然界的生物群落系统中,由于同种生物个体之间(部分之间)的相互作用关系会形成种群;各个种群由于其中的个体(部分)之间的相互作用会表现出一定的整体属性,即群属性。各个群以其属性为基础,相互之间的作用又构成了群落。例如,在一座山或一片森林里全体生物的总体构成了群落;若仅考虑其中的两种或几种生物,如狼和野兔的捕食关系,这两种生物也可以看作是一个群落。

5.1.1　生物群落的层次性信息网络拓扑结构

生物群落拓扑结构是一个多层结构。在每个种群内,由个体间的交流拓扑构成了群体的拓扑结构;而在群落层面,种群之间的交流方式又构成了更高一层的拓扑网络。如图 5-1 所示,不同种群通过相互之间的竞争合作关系构成了群落的层次拓扑结构。

种群间的拓扑结构并不影响种群内部的拓扑结构,即种群内和种群间的拓扑构型可以各不相同。图 5-2 为一个种群之间为松散的环形链接,内部为紧密的晶体链接的层次异构结构。而在图 5-3 中,种群内部分别为环形、晶体、轮形和星形结构,种群间为全链接结构。

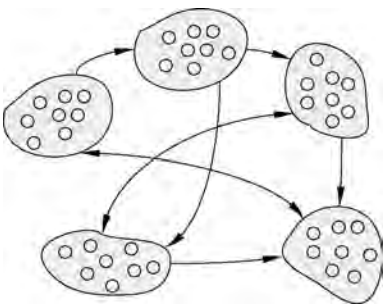


图 5-1　群落拓扑结构

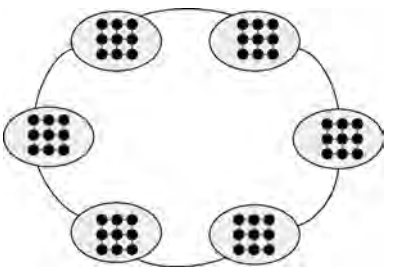


图 5-2　内部为晶格连接外部为松散链接的 2 层拓扑结构

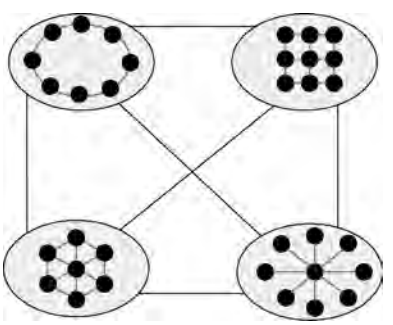


图 5-3　种群内部和种群间均为不同拓扑结构

5.1.2　生物群落内种群共生模式的多型性

在生物群体的交互关系中，有一种方式很引人注意，即不同物种之间的共生行为。共生一词在英文或是希腊文中，字面含义就是“共同”和“生活”，在生物学中指的是两种生物彼此互利地生存在一起、缺此失彼都不能生存的一类种间关系，是生物之间相互关系的高度发展。共生是生物在生理上相互分工，互换生命活动的产物，在组织上形成了新的结构[3]。术语“宿主”通常被用来指共生关系中较大的成员，较小者称为“共生体”。共生依照位置可以分为外共生、内共生。就外共生而言，共生体生活在宿主的表面，包括消化道的内表面或是外分泌腺体的导管；而在内共生机制下，共生体生活在宿主的细胞内或是个体身体内部（在细胞外）都有可能。20 世纪末的生物学研究推测，细胞内的叶绿体和线粒体也可能是内共生的形式之一。地衣是众所周知的共生实例，它是藻类和菌类的共生体。除地衣以外，在生物界的很多门类都可以举出许多共生的例子。昆虫纲等翅目的昆虫及其肠道中的鞭毛虫之间的关系就是共生关系。等翅目昆虫的肠道是鞭毛虫或细菌的栖身之所，它们帮助等翅目昆虫消化纤维素，而等翅目昆虫不仅为它们提供藏身之所，还给它们提供养料。若互相分离，两者都不能生存。

根据共生对共生关系的生物体利弊关系而言，种群的共生模式又可以分为互利共生、偏利共生和寄生。

1. 互利共生[1]（++）

互利共生指的是共生的生物体成员彼此受益。例如，小丑鱼居住在海葵的触手之间，这些鱼可以使海葵免于被其他鱼类食用，而海葵有刺细胞的触手，可使小丑鱼免于被掠食；而

小丑鱼本身则会分泌一种黏液在身体表面，保护自己不被海葵伤害。一些寄居蟹会将海葵背于壳上。一些虾虎鱼种类，可和枪虾类形成共生。枪虾会在沙中挖掘洞穴并且清理它，这两种生物就居住在这个洞穴里面，枪虾几乎是全盲的，因此若在地面（水中的地面）有天敌的状况下会变得非常脆弱，在危急的情况下虾虎鱼用尾巴碰触虾，以警告它们身处危险之中，随后两种生物都会迅速退回洞穴中保护自己。在陆地环境，有一种鸟以擅长捕食鳄鱼身上的寄生虫而出名，而鳄鱼也欢迎鸟类在身上寻找寄生虫和清理食物残渣，甚至张大口颚以利鸟儿安全地至口中觅食。对鸟来说，这不仅是现成的食物来源，也是一个很安全的环境，因为许多掠食者不敢在鳄鱼身边攻击这些鸟类。

2. 偏利共生（+0）

偏利共生是指其中一方生物体受益，却对另一方无害。例如，印首鱼会利用头部的吸盘状构造，吸附在其他鱼类的表面，但是不造成伤害，借着被附着的个体的活动而行于水中。一些植物或是蕨类，例如鸟巢蕨（又称做山苏花）会附生在其他的植物上，特别是大树上较为平坦的一小块区域都是它们选择附生的所在。

3. 寄生（+−）

寄生是指一种生物寄附于另一种生物身体内部或表面，利用被寄附生物的养分生存，寄生对一方有利而对另一方有害。主要的寄生物有细菌、病毒、真菌和原生动物。在动物中，寄生蠕虫特别常见，而昆虫是植物的主要寄生物。根据寄生程度的不同又可分为专性寄生和兼性寄生，其中前者必须以宿主为营养来源，后者也能自由活动。较为极端的是：一类寄生生物在完成一定的发育后，会将宿主食尽，对这种类型的寄生物特称为捕食性寄生物（Parasitoid）或者拟寄生物，以与一般的寄生物相区别。拟寄生物包含一大类昆虫寄生物，它们在昆虫宿主身上或体内产卵，通常会导致宿主死亡。

5.1.3　生物群落内种群的增长、迁徙和消亡模式

生物群落是不断运动的。我们知道，大自然中生物群落的动态变化受环境条件（特别是气候）周期性变化的制约，并与生物种的生活周期关联。群落的变化动态是群落本身内部的变化，是随着群落内各个物种种群的变化而变化的。组成一个群落的物种在其内部以及物种之间都存在特定的相互关系。这种关系随着外部环境条件和群落内环境的改变而不断进行调整。比如当一个种群密度增加时，种群内部的关系会紧张化，竞争能力强的种群将得以充分发展，而竞争能力弱的种群则不断缩小自己的地域，甚至被排挤到其群落之外。所以群落变化的实质是群落内种群的变化[4]。要想掌握群落的动态，必须熟悉一个种群的变化模式。下面对种群的变化模式进行介绍。

1. 种群的增长

种群中个体的数量经常在变化，也就是说，种群大小经常发生变化，这种变化是生态学工作者最感兴趣的问题。生态学工作者利用数学模型来模拟种群规模的变化。其中，以指数级增长是一种常见的方式。

在无限环境中，因种群不受任何限制因子的约束，种群的潜在增长能力得到了最大限度的发挥，种群数量呈现指数式增长格局，种群的这种增长规律称为种群的指数增长规律。常用指数模型进行描述，模型的假设：

（1）种群增长是无界的，即种群在无限的环境中生长，不受食物、空间等条件的限制；

(2) 世代不相重叠(即寿命仅一年),增长是不连续的(即一年只有一个繁殖季节)或称为离散的;

(3) 种群无迁出和迁入(生境是彼此隔离的);

(4) 种群无年龄结构。

数学模型:世代不相重叠种群的离散增长模型,通常是把世代 $t+1$ 的种群 N_{t+1} 与世代 t 的种群 N_t 联系起来的差分方程式:

$$N_{t+1}=lN_t \quad 或 \quad N_t=N_0l^t$$

式中,N 为种群大小;t 为时间;l 是种群的周期增长率。

2. 种群的迁徙

某些无脊椎动物,如东亚飞蝗、蝴蝶等;爬行类,如海龟等;哺乳类,如蝙蝠、鲸、海豹、鹿类等;还有某些鱼类都有季节性的长距离更换住处的行为,这种行为称为迁徙。动物的迁徙都是定期的、定向的,而且多是集成为大群地进行。

鸟类的迁徙每年在繁殖区和越冬区之间周期性地发生,大多发生在南、北半球之间,少数在东、西方向。人们按鸟类迁徙活动的有无把鸟类分为候鸟和留鸟。留鸟终年留居在出生地,不发生迁徙,如麻雀、喜鹊等。候鸟中夏季飞来繁殖、冬季南去的鸟类被称为夏候鸟,如家燕、杜鹃等;冬季飞来越冬、春季北去繁殖的鸟类称为冬候鸟,如某些野鸭、大雁等。鱼类的迁徙活动有一个专有名称叫“洄游”。大多数的鱼类可以说都是洄游鱼类,只有少数鱼类不表现出规律性的洄游。鱼类洄游按目的分为 3 种:生殖洄游、索饵洄游和越冬洄游。青鱼、草鱼、鲢鱼、鳙鱼、大黄鱼、小黄鱼、大马哈鱼都进行生殖洄游。哺乳动物的迁徙规模浩大,例如驯鹿的千里踏雪大迁徙。每年入冬,成千上万头的驯鹿汇集成巨大的鹿群,从北向南,朝森林冻土带的边缘地带转移。次年春天,它们再向北方的北冰洋沿岸进发。四五月份,鹿群到达它们熟知的冻土带僻静处,在此养育儿女。昆虫的迁徙有时能创造奇迹,最著名的是产于美洲的彩蝶王,它们春天从中美洲飞到加拿大,秋天又飞回中美洲,行程 4500km,历时几个月,真是令人惊叹不已。

动物的迁徙行为是一种适应现象,凭借这种活动,可以满足它们在特定的生活时期所需要的环境条件,使个体的生存和种族的繁荣得到可靠的保证。与此同时,这些动物所处区域的生态系统也会随着迁徙行为的进行发生周期性的变化。

3. 种群的消亡

种群是在一定环境下由个体组成的。个体除了具有生物学特征外,还具有出生、生长、衰弱、死亡等状态。随之带来的是种群的增长,强大、衰落与消亡。当环境发生剧烈变化时,可能会造成整个种群出生率急剧下降或死亡率急剧上升。个体大规模死亡,甚至整个种群消亡。例如,白垩纪时期所发生的恐龙灭绝就是在极端环境造成的整个物种消亡。

5.2 基于生物群落演化的计算模式设计

5.2.1 基于生物群落演化的统一优化框架

基于生物群落的优化框架是在生物群体行为优化框架的基础上加上群体之间协作与竞争等交互组成的。在群体层面上,一个群体表现出了分工、协作和通信交流等智能行为。在

群落层面上，种群之间存在信息交流与种间协同进化等现象。同时每个种群还独自表现出了增长、壮大、衰老和消亡等现象。

第 4 章中已经建立了基于生物群体行为的优化框架。本章在此基础上建立了以种群为基础的生物群落统一优化框架，如图 5-4 所示。

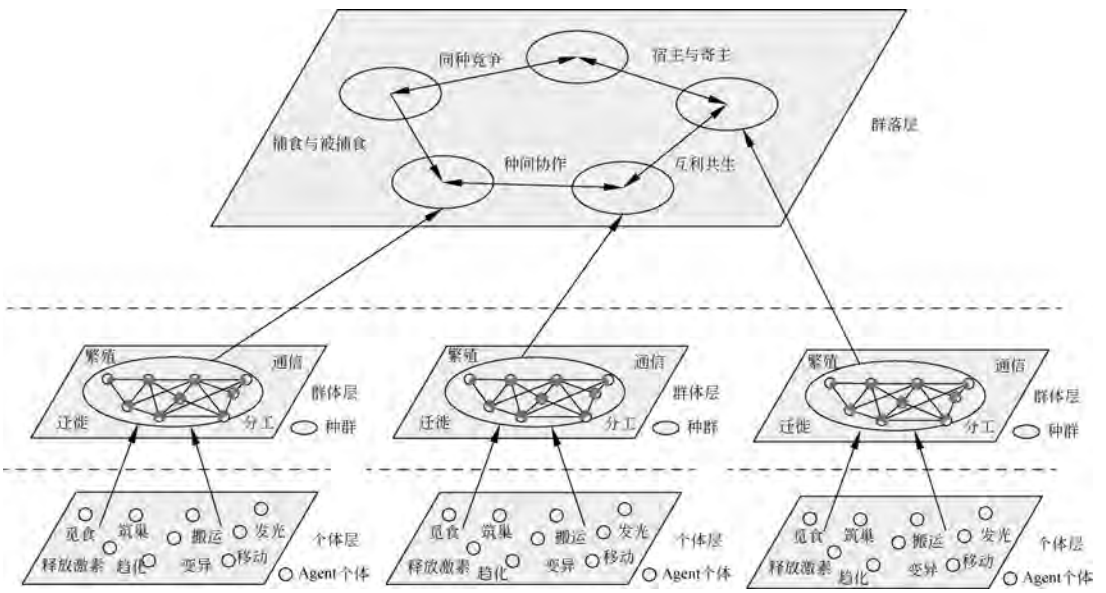


图 5-4　基于生物群体行为的优化层次框架

本章基于生物群落的统一优化框架模型可以形式化表示如下：

$$BCM=(Individual,Population,Colony,Environment,Topology,Coevolution)$$

(1) Individual：群落中的个体。

$$Individual=\{Individual_1,Individual_2,\cdots,Individual_i,\cdots,Individual_{INUM}\}$$

$$Individual_i=<IID,INOBJ,STATUS,BHV,MESS,ACT,TANS>$$

(2) Population：群落中的种群。

$$Population=\{Population_1,Population_2,\cdots,Population_i,\cdots,Population_{PNUM}\}$$

其中，每个种群都有微观特性和宏观特性。

- 种群微观特性表示如下：

$$Population_1=\{Individual_1,Individual_2,\cdots,Individual_{INUM}\}$$

$$=\begin{Bmatrix} IID_1 & INOBJ_1 & STATUS_1 & BHV_1 & MESS_1 & ACT_1 & TANS_1 \\ IID_2 & INOBJ_2 & STATUS_2 & BHV_2 & MESS_2 & ACT_2 & TANS_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ IID_{INUM} & INOBJ_{INUM} & STATUS_{INUM} & BHV_{INUM} & MESS_{INUM} & ACT_{INUM} & TANS_{INUM} \end{Bmatrix}$$

- 种群宏观特性表示如下。

$$Population=<PID,POBJ,OPT,PBHV,DRE,SPA,$$
$$N(t),DEN,STR,INPUT,OUTPUT,TRAN>$$

(3) Col：模型中的群落。

$$\mathrm{Col}=\{\mathrm{Col}_i^1\{\mathrm{Col}_j^2\{\mathrm{Col}_k^3\{\cdots\{\mathrm{Col}_x^{\mathrm{Num}}\}\}\}\}\}$$

$$\mathrm{Col}_x^{\mathrm{Num}}=<\mathrm{CID},\mathrm{CN}(t),\mathrm{CDRE},\mathrm{CSPA},\mathrm{CSTR},$$
$$\mathrm{CDEN},\mathrm{CBHV},\mathrm{INPUT},\mathrm{OUTPUT},\mathrm{TRAN}>$$

具体含义详见 5.3.1 节。

(4) Environment：环境，即优化的目标函数。

其中，Environment={Range,Food}，具体含义详见 2.2 节。

(5) Topology：群落拓扑结构。

其中，$\mathrm{Topology}_k=(t_k,T_k,\mathrm{TS},\mathrm{TD},\{A_{ik},P_{jk}^n\})$，具体含义详见 5.3.2 节。

(6) Coevolution：不同个体、群体和层次之间的协同进化规则。

其中，Coevolution={Rul,Sym}，Rul 具体表示方法详见 5.3.3 节，Sym 具体表示方法详见 5.3.3 节。

5.2.2 基于生物群落演化的基本操作

由于生物群落是以多个生物种群为基础，因此基于生物群落的协同进化的本质是种群与种群之间的相互作用、相互交流。其主要体现在以下几个方面。

1. 种群间个体交流

(1) 种群之间的个体交换。用一个群体中的部分个体代替另一个群体中的部分个体。以加快整个群落的演化速度。

(2) 种群的合并、融合。两个或多个种群合并为一个种群。

2. 种群之间的个体信息共享与传递

在共生关系中，重要的一点就是资源的互相给予和利用。豆科植物和根瘤菌在整个共生阶段，根瘤菌被包围在宿主质膜所形成的侵入线中，在宿主内合成固氮酶。豆血红蛋白则系共生作用产物。具体来讲，植物产生球蛋白，而血红素则由细菌合成。豆血红蛋白存在于植物细胞的液泡中，对氧具有很强的亲和力，因此对创造固氮作用所必需的厌氧条件是有利的。就这样细菌开始固氮。在植物体内细菌有赖于植物提供能量，而类菌体只能固氮而不能利用所固定的氮。所以豆科植物供给根瘤菌碳水化合物，根瘤菌供给植物氮素养料。这种养分的互相供给在生物启发算法中可以表现为不同种群间的信息共享。

不同种群之间信息共享方式有多种。对偏利共生型群落结构，可能共享优秀个体信息，其他种群个体可以向这些优秀个体学习。对竞争性群落结构，被捕食种群个体可能根据捕食种群个体的位置、距离发生逃避行为。

3. 种群消亡与再生[1]

种群消亡与再生的思想主要来源于生物界的物种灭绝，本意是泛指植物或动物的种类不可再生性地消失或破坏，在地球的生命演化中扮演着重要的角色。据统计，全世界每天有 75 个物种灭绝，每小时有 3 个物种灭绝。在生物群落演化的计算模式中，种群消亡与再生是对种群的一个重要操作，每个种群都有自己的生命周期，当一个生命周期结束时，种群中属性不好的部分个体就要消亡，而另一些新的个体诞生，补充进来，保持种群的规模。

(1) 消亡规则。对长期不能适应环境的个体(具体表现为适应度或累积适应度差)，将其从种群中剔除。

(2) 再生规则。为保持个体的平衡和算法的稳定性，根据变量范围随机生成与消亡等量的个体，与初始化规则相同。

5.3　生物群落建模与仿真分析

5.3.1　生物系统群落的形式化定义

群落由多个种群组成，种群间的行为及关系构成了群落微观状态。但组织格局进入到群落水平时，生物个体和群体已成为较大和较复杂生物体系中的一部分，此时，作为整体的群落出现了许多不为种群所拥有的独有的新特征，如群落密度、群落等级等，即群落宏观信息集合。群落具有一定的自然属性，这种自然属性由底层所有个体和各层次种群的自然属性和社会属性共同出现形成。

$$\mathrm{Col}_x^{\mathrm{Num}} = <\mathrm{CID},\mathrm{CN(t)},\mathrm{CDRE},\mathrm{CSPA},\mathrm{CSTR},\mathrm{CDEN},\mathrm{CBHV},\mathrm{INPUT},\mathrm{OUTPUT},\mathrm{TRAN}>$$

(1) CID：群落标识符，表示该群落在生态系统中的身份。

$$\mathrm{CID}=\{\mathrm{CI},((1,i),(2,j),\cdots,(\mathrm{Num}-2,l),(\mathrm{Num}-1,u),v)\}$$

其中，CI 表示当前种群的绝对身份，即代表整个系统中的标识。$((1,i),(2,j),\cdots,(\mathrm{Num}-2,l),(\mathrm{Num}-1,u),v)$描述的是当前种群在群落中的相对身份，即第 1 层的第 i 个子种群落，第 2 层的第 j 个子种群落，以此类推，第 Num－1 层的第 u 个子种群落，第 Num 层的第 v 个子种群落。

(2) CN(t)：此群落中种群个数，其值可以为常量，也可以为变量。

(3) CDRE：此群落内种群等级。

(4) CSPA：此群落空间分布范围。

(5) CSTR：此群落空间格局。组成群落的种群在其空间中的位置或布局，随机分布、均匀分布或集群分布。

(6) CDEN：此群落种群密度。

(7) CBHV：此群落种群间个体交流方式。

(8) INPUT：此群落接收的外界信息集合，如子种群落消亡信息、迁移信息等。这类信息是对此群落的宏观调控，并可对此群落内所有单元产生其相应作用。

(9) OUTPUT：此群落进化过程中的输出信息集合，包括涌现的知识、求解问题的阶段性成果等。

(10) TRAN：此群落群体状态的转换函数，也可称为刺激-反应规则。TRAN：INPUT×$\mathrm{STATUS}_i \rightarrow \mathrm{STATUS}_{i+1}$，表示此群落在接收到某些宏观调控信息后状态的转换。

5.3.2　群落拓扑结构的形式化定义

群落拓扑结构形式化描述如下：

$$\mathrm{Topology}_k = (t_k, T_k, \mathrm{TS}, \mathrm{TD}, \{A_{ik}, P_{jk}^n\})$$

(1) t_k：$0 \leqslant t_k \leqslant t_{\mathrm{end}}$，表示当前时间。如果拓扑结构为静态，则 t_k 为任一常数。

(2) T_k：第 k 个时间点整个群落所有层次所有进化单元的拓扑关系集合。

$$\{T_k\}=\begin{Bmatrix} T^k & & & & & & \\ T^k_{1,1} & T^k_{1,2} & \cdots & T^k_{1,C1} & & & \\ T^k_{2,1} & T^k_{2,2} & \cdots & \cdots & T^k_{2,C2} & & \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \\ T^k_{\mathrm{Num},1} & T^k_{\mathrm{Num},2} & \cdots & \cdots & \cdots & \cdots & T^k_{\mathrm{Num},\mathrm{CNum}} \end{Bmatrix}$$

其中，$\{T_k\}$中的元素均可为静态拓扑关系或动态拓扑关系，$T^k_{i,j}=(\mathrm{TS}_m)\mathrm{OR}(\mathrm{TD}_m)$。

(3) TS：整个群落所有静态拓扑关系集合。

$$\mathrm{TS}=\begin{Bmatrix} \mathrm{TS}_1 \\ \mathrm{TS}_2 \\ \vdots \\ \mathrm{TS}_{\mathrm{TSnum}} \end{Bmatrix}=\begin{Bmatrix} \mathrm{sname}_1 & \mathrm{deg}_1 & \mathrm{cc}_1 & \mathrm{apl}_1 & \mathrm{adeg}_1 & \cdots \\ \mathrm{sname}_2 & \mathrm{deg}_2 & \mathrm{cc}_2 & \mathrm{apl}_2 & \mathrm{adeg}_2 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathrm{sname}_{\mathrm{TSnum}} & \mathrm{deg}_{\mathrm{TSnum}} & \mathrm{cc}_{\mathrm{TSnum}} & \mathrm{apl}_{\mathrm{TSnum}} & \mathrm{adeg}_{\mathrm{TSnum}} & \cdots \end{Bmatrix}$$

其中，TSnum 表示觅食时间内静态拓扑关系的个数；sname 表示静态拓扑关系名称；deg 表示拓扑关系直径；cc 表示聚类系数；apl 表示平均路径长度；adeg 表示平均度。另外，也可以根据此描述方法增加拓扑关系所需的参数。

(4) TD：整个群落所有动态拓扑关系集合。

$$\mathrm{TD}=\begin{Bmatrix} \mathrm{TD}_1 \\ \mathrm{TD}_2 \\ \vdots \\ \mathrm{TD}_{\mathrm{TDnum}} \end{Bmatrix}=\begin{Bmatrix} \mathrm{dname}_1 & \mathrm{function}_1 & \mathrm{Iupdate}_1 & \cdots \\ \mathrm{dname}_2 & \mathrm{function}_2 & \mathrm{Iupdate}_2 & \cdots \\ \vdots & \vdots & \vdots & \vdots \\ \mathrm{dname}_{\mathrm{TDnum}} & \mathrm{function}_{\mathrm{TDnum}} & \mathrm{Iupdate}_{\mathrm{TDnum}} & \cdots \end{Bmatrix}$$

其中，TDnum 表示觅食时间内动态拓扑关系的个数；dname 表示动态拓扑关系名称；function 表示邻域函数；Iupdate 表示个体状态更新方式。动态拓扑关系有时很难用邻域函数表示，此时，也可以根据此描述方法增加特殊拓扑关系所需的参数。目前研究较多的静态拓扑关系有全互连结构、环形结构、冯·诺依曼结构、随机结构及小世界网络等。

(5) A_{ik}：表示第 k 个时间点个体 i 的邻域关系。

$$A_{ik}=(X^k_i,\Psi^k_p),\quad \Psi^k_p=\{\Psi^k_{1,a},\Psi^k_{2,b},\cdots,\Psi^k_{p,\cdots}\}$$

其中，X^k_i 表示第 k 个时间点个体 i 的属性；Ψ^k_p 是与个体 A_{ik} 存在协作关系的其他个体集合；p 是此集合中个数总数目；$a,b,\cdots$表示每个个体在整个个体集合中的序号，$1\leqslant a,b,\cdots\leqslant I$。

(6) P^n_{jk} 表示第 k 个时间点第 n 层第 j 个群体的邻域关系：

$$P^n_{jk}=(P^{n,k}_j,S^k_Q),\quad S^k_Q=\{S^k_{1,a},S^k_{2,b},\cdots,S^k_{Q,\cdots}\}$$

其中，$P^{n,k}_j$ 表示第 k 个时间点第 n 层 j 个群体的属性；S^k_Q 表示是与群体 P^n_{jk} 存在协作关系的其他群体集合；Q 是此集合中群体总数目。

5.3.3 基于不同种群关系生物群落演化建模与仿真

1. 种群共生关系

生物种群生活在一定空间里相互会有着直接或间接的共生进化关系。在自然生态系统中，共生进化现象非常普通。物种在生存区域内，通常存在多个群体，每个群体都自己的个体类型。不仅群体中的个体可以相互交流，而且不同类型的群体之间也可以共生进化，从而

实现这片生存区域内所有个体的共生。

(1) 互利共生(mutualism)。互利共生是指两种生物生活在一起,彼此有利,两者分开以后都不能独立生活。图 5-5 很好地描述了两个物种随着进化的进行而展现出的两者之间的关系。可以看到,两个种群都能够迅速地收敛于最优点,并且收敛速度几乎同步。这说明它们之间相互合作共享了最优值的信息。

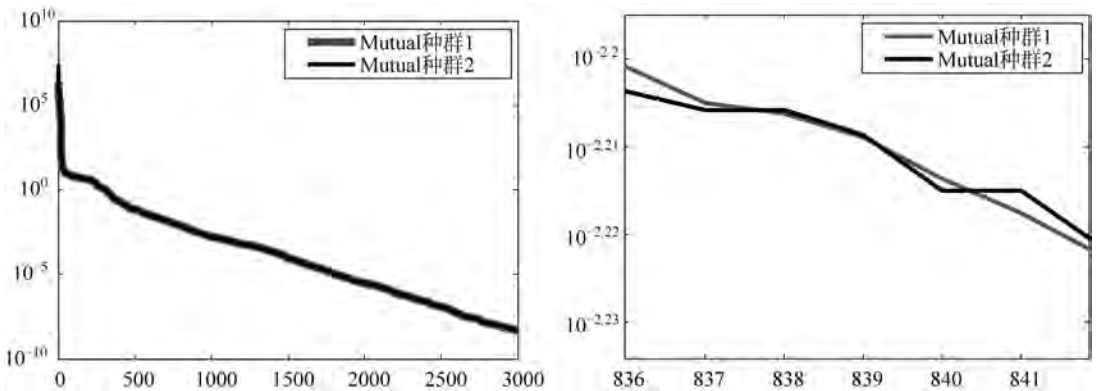


图 5-5　互利共生的两物种之间进化关系图

(2) 共栖(commensalism)。共栖是指两种生物生活在一起,对一方有利,对另一方也无害,或者对双方都有利,两者分开以后都能够独立生活。例如,有些附生植物附着在大树上,借以得到充足的光照,但是并不吸收大树体内的营养。海葵常常固着在寄居蟹的外壳上,海葵靠刺细胞防御敌害,能对寄居蟹间接地起到保护作用而寄居蟹到处爬动,可以使海葵得到更多的食物,但是它们分开以后仍能独立生活。由图 5-6 可以看出,Host 种群独自进行优化,而 Commensal 种群借鉴了 Host 种群的信息使其迅速收敛,该图很好地反映了这种共栖状态下两种生物的进化关系。

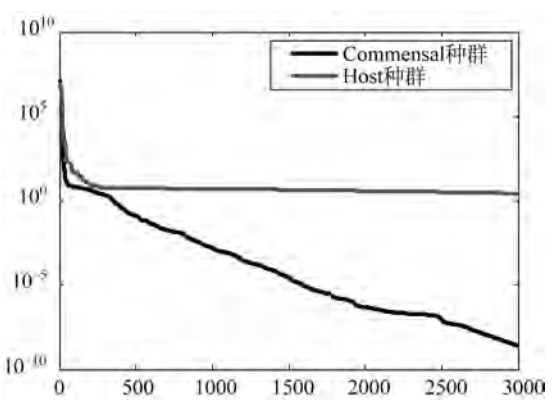


图 5-6　共栖的两物种之间进化关系图

(3) 寄生(parasitism)。寄生是指两种生物在一起生活,一方受益,另一方受害,后者给前者提供营养物质和居住场所,这种生物的关系称为寄生。图 5-7 描绘了具有寄生的两物种之间的关系。当 Host 种群通过逃离 Parasite 种群时,其种群能够壮大生长。而当 Parasite 种群追踪到 Host 种群后利用了 Host 发现的最优区域,因此在迭代后期收敛效果较好。

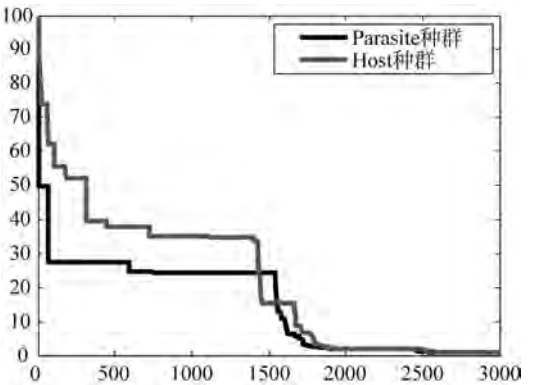


图 5-7　竞争的两物种之间进化关系图

(4) 捕食者-猎物(predator-prey)。捕食者与猎物之间的竞争与寄生类似,也是一方受益而另一方受害。捕食者与被捕食者通过这种进攻与防御的竞争,互相不断进化。这是典型的一种协同进化。在此不多做介绍。

基于生物群落的优化思想主要来源于以上介绍的协同进化概念,应用到实际优化中,可理解为更广义的概念。不仅是不同物种的个体生活在一起,形成相互都受益的关系,而且也可以有相同物种的不同种群中的个体互相合作的关系。同样,上面中提到的例如互利共生的概念也使用较广义的术语定义,所以此术语也可指相互离开也可正常生存的生物组合中物种间的关系。

2. 共生进化模型

生物在长期进化过程中的共生关系多种多样,依照对共生关系的生物体利弊关系而言,共分为 4 类:无关共生、合作共生、寄生共生和竞争共生,如表 5-1 和图 5-8 所示。

表 5-1　共生单元间的共生类型

类　　型		单元 A	单元 B	特　　征
无关共生关系[图 5-8(a)]		0	0	双方都无益无损
合作	互利共生[图 5-8(b)]	+	+	共生双方都得到好处
	偏利共生[图 5-8(c)]	+	0	对一方有利,对另一方没有影响
寄生关系(捕食关系)[图 5-8(d)]		+	−	对其中一方有利,对另一方有害
竞争	竞争共生[图 5-8(e)]	−	−	共生双方都受损
	偏害共生[图 5-8(f)]	−	0	对一方有害,对另一方没有影响

"+"表示有利,"−"表示有害,0 表示既无利也无害。

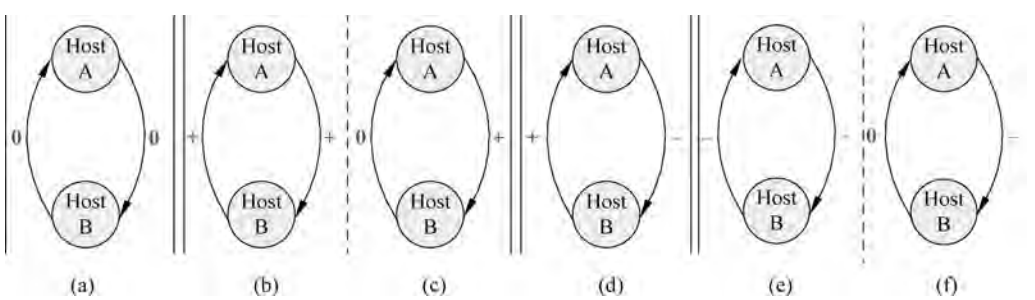


图 5-8　共生关系分类

整个群落内的种群和子种群落的共生关系都可按上述关系进行划分，因此，统一框架模型中的共生关系可按如下形式来描述：

$$\mathrm{Sym}=\{\mathrm{unitA}_i\}=(\{\mathrm{signa}\},\{a\},\{\mathrm{unitB}\},\{\mathrm{signb}\},\{b\})$$

$$=\begin{Bmatrix}\mathrm{signa}_1 & a_1 & \mathrm{unitB}_1 & \mathrm{signb}_1 & b_1\\ \mathrm{signa}_2 & a_2 & \mathrm{unitB}_2 & \mathrm{signb}_1 & b_2\\ \vdots & \vdots & \vdots & \vdots & \vdots\\ \mathrm{signa}_{\mathrm{unum}} & a_{\mathrm{unum}} & \mathrm{unitB}_{\mathrm{unum}} & \mathrm{signb}_1 & b_{\mathrm{unum}}\end{Bmatrix}_i$$

其中{unitB}表示与 unitA 有共生关系的单元集合，unitA 和{unitB}中的元素可为种群，也可为子种群落。{signa}和{signb}中的元素表示它们之间的共生关系。如{signa}＝＋，{signb}＝＋，则表示它们之间为互利共生。如{signa}＝－，{signb}＝0，则表示它们之间为偏害共生。{a}和{b}中的数值分别表示他们之间的共生程度。如{signa}＝＋，{signb}＝－，$a_j=5$，$b_j=4$，则表示 unitA 和 unitB_j 之间为寄生关系（捕食关系），unitA 受益程度为 5，unitB_j 受害程度为 4。如果省略这两个数值，则表示它们为完全共生。

3. 举例：子种群协作群搜索算法

基于群搜索算法，考虑到约束优化问题的特点，利用偏利共生型协同进化思想，设计了子种群协作群搜索算法。算法中包括共同生活在某个区域内且位置相邻的两个群体，满足约束条件的个体组成可行子种群（feasible subswarm）；反之，不满足约束条件的个体组成了不可行子种群（infeasible subswarm）；这两个子种群按照偏利共生型协同进化方式共生。可行子种群为主群，不可行子种群为从群。首先，每个群的成员都能按照群的特有生存方式生活，互不干扰。其次，主群能从从群获得利益并使主群内的成员更好地进化，但这种利益关系对于从群成员无关紧要。

在可行子种群和不可行子种群之间存在着一个约束边界。在偏利共生型协同进化方式中它起到了“单向围墙”作用，如图 5-9 所示。可行子种群内的个体可在可行域内自由搜索，一旦当个体通过搜索移动到了不可行域，这个围墙会阻止个体的这次行为，让它回到原来的可行域内的另一个位置。不可行子种群内的个体既可以继续在不可行域内搜索，也可以通过搜索进入可行域，从而成为可行子种群成员。

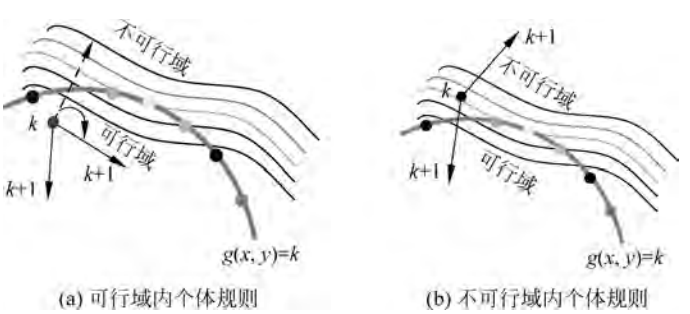


图 5-9　个体行为规则

可行子种群的搜索策略与标准群搜索算法类似，群内成员同样被分为 3 类：发现者、加入者和游荡者。在每轮迭代中，当前位置最佳的个体为此轮的发现者，其他个体随机地被选择为加入者或游荡者。发现者和加入者的搜索策略与标准群搜索算法中的相同，唯一不同

的是游荡者搜索策略。在可行子种群内，每个个体都会保存它整个搜索过程中的历史最优位置信息。当个体的角色为游荡者时，个体会基于它的历史最优位置进行随机步长的独立搜索，如式(5-1)所示。

$$X_i^{k+1} = X_p^k + l_i D_i^k(\varphi_i^{k+1}) \tag{5-1}$$

在不可行子种群内，不存在全局最优个体，因此只有搜索者和游荡者两类成员。每个个体都可以在这两种角色中切换。这两种角色的个体的搜索策略与标准群搜索算法相同，此处不再赘述。

本书将子种群协作搜索算法应用于 1 个优化函数和 3 个机械优化设计问题。

1) Himmelblau's 函数

Himmelblau's 函数是非线性优化问题，包括 5 个设计变量和 6 个非线性约束条件。

目标函数：$f(X)=5.257\,854\,7x_3^2+0.835\,689x_1x_5+37.293\,239x_1-40.792\,141$

约束条件：$g_1(X)=85.334\,407+0.005\,685\,8x_2x_5+0.000\,626\,2x_1x_4-0.002\,205\,3x_3x_5\leqslant 92$

$g_2(X)=80.512\,49+0.007\,131\,7x_2x_5+0.002\,995\,5x_1x_2+0.002\,181\,3x_3^2\leqslant 110$

$g_3(X)=9.300\,961+0.004\,702\,6x_3x_5+0.001\,254\,7x_1x_3+0.001\,908\,5x_3x_4\leqslant 25$

其中，$78\leqslant X_1\leqslant 102$，$33\leqslant X_2\leqslant 45$，$27\leqslant X_3\leqslant 45$，$27\leqslant X_4\leqslant 45$ 及 $27\leqslant X_5\leqslant 45$。

2) 压力容器

工程上常见的半球形封头压力容器设计简单，如图 5-10 所示，广泛应用于石油及化学等工业领域。在力求满足强度等要求的前提下，以压力容器重量为目标函数。此问题共 4 个约束条件和 4 个优化变量。

目标函数：$f(X)=0.6224X_1X_3X_4+1.7781X_2X_3^2+3.1661X_1^2X_4+19.84X_1^2X_3$

约束条件：$g_1(X)=0.0193X_3-X_1\leqslant 0$

$g_2(X)=0.009\,54X_3-X_2\leqslant 0$

$g_3(X)=1\,296\,000-\pi X_3^2X_4-\frac{4}{3}\pi X_3^3\leqslant 0$

$g_4(X)=X_4-240\leqslant 0$

其中，X_1 和 X_2 分别为封头(T_h)和筒体壁厚(T_s)，$0.0625\leqslant X_1, X_2\leqslant 6.1875$；$X_3$ 为筒体及封头底面半径(R)；X_4 为筒体长度(L)，$10\leqslant X_3, X_4\leqslant 200$。在 4 个变量中，$X_1$ 和 X_2 是间隔为 0.0625 的均匀离散变量，X_3 和 X_4 是连续变量。

3) 压缩弹簧

压缩弹簧广泛使用在一些机械中，如内燃机和压缩机等，如图 5-11 所示，因而弹簧性能的好坏直接影响到机器的工作性态。压缩弹簧通常以质量最小作为最优化设计目标。此问题共 4 个约束条件和 3 个优化变量。

目标函数：$f(X)=(X_3+2)X_2X_1^2$

约束条件：$g_1(X)=1-\frac{X_2^3X_3}{71\,785X_1^4}\leqslant 0$

$$g_2(X)=\frac{4X_2^2-X_1X_2}{12\,566(X_2X_1^3-X_1^4)}+\frac{1}{5108X_1^2}-1\leqslant 0$$

$$g_3(X)=1-\frac{140.45X_1}{X_2^2X_3}\leqslant 0;\ g_4(X)=\frac{X_2+X_1}{1.5}-1\leqslant 0$$

其中，X_1 为弹簧线径(d)，$0.05\leqslant X_1\leqslant 2$；$X_2$ 为弹簧圈均径(D)，$0.25\leqslant X_2\leqslant 1.3$；$X_3$ 为弹簧线圈数目，$2\leqslant X_3\leqslant 15$。这 3 个变量均为连续变量。

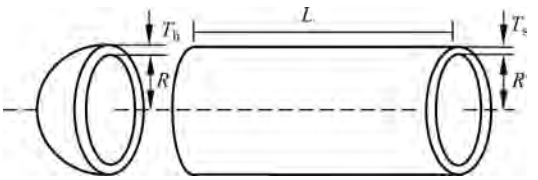


图 5-10　压力容器

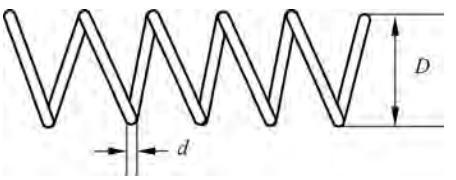


图 5-11　压缩弹簧

4) 焊接悬臂梁

焊接梁优化设计问题以最小化焊接梁的总费用为优化目标，如图 5-12 所示。此问题共 7 个约束条件和 4 个优化变量。

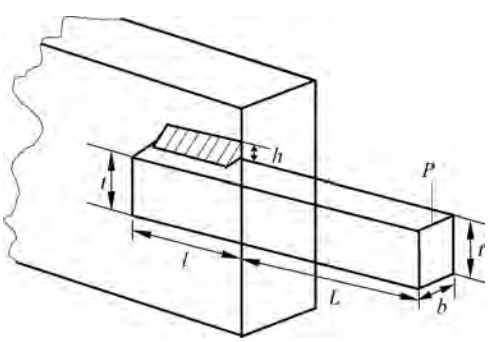


图 5-12　焊接悬臂梁结构示意图

目标函数：$f(X)=1.104\,71x_1^2x_2+0.048\,11x_3x_4(14.0+x_2)$

约束条件：$g_1(X)=\tau(X)-13\,000\leqslant 0$；$g_2(X)=\sigma(X)-30\,000\leqslant 0$

$$g_3(X)=x_1-x_4\leqslant 0$$

$$g_4(X)=0.104\,71x_1^2+0.048\,11x_3x_4(14.0+x_2)-5\leqslant 0$$

$$g_5(X)=0.125-x_1\leqslant 0;\ g_6(X)=\delta(X)-0.25\leqslant 0$$

$$g_7(X)=6000-P_c(X)\leqslant 0$$

$$\tau(X)=\sqrt{(\tau')^2+2\tau'\tau''\frac{x_2}{2R}+(\tau'')^2},\quad \tau'=\frac{6000}{\sqrt{2}x_1x_2},\tau''=\frac{MR}{J},\quad M=6000\left(L+\frac{x_2}{2}\right),$$

$$R=\sqrt{\frac{x_2^2}{4}+\left(\frac{x_1+x_3}{2}\right)^2},\quad \delta(X)=\frac{504\,000}{x_3^3x_4},\quad J=2\left\{\frac{x_1x_2}{\sqrt{2}}\left[\frac{x_2^2}{12}+\left(\frac{x_1+x_3}{2}\right)^2\right]\right\}$$

$$\sigma(X)=2.1952/x_4x_3^2,P_c(X)=64\,746.022(1-0.028\,234\,6x_3)x_3x_4^3$$

其中，优化目标为最小化焊接梁的总费用。焊接厚度 $h=X_1$，焊接接头长度 $l=X_2$，梁的宽度 $t=X_3$ 和梁的厚度 $b=X_4$。$0.1\leqslant X_1\leqslant 2.0$，$0.1\leqslant X_2\leqslant 10$，$0.1\leqslant X_3\leqslant 10$，$0.1\leqslant X_4\leqslant 2.0$。

变量 X_1 和 X_2 是间隔为 0.0065 的均匀离散变量，X_3 和 X_4 是连续变量。

表 5-2 列出了 iGSO 算法基于上述问题分别执行 30 次获得的最优值、平均最优值、最差值和标准差。可以看出，iGSO 算法能够获得较优秀的解。除此之外，表 5-3 显示出 iGSO 最终求得的机械变量设计方案没有违反任何约束条件。因此 iGSO 算法不仅对约束优化测试函数有效，而且可以用于实际的工程应用。

表 5-2　函数测试结果

结　　果	Himmelblau's	压力容器	压缩弹簧	焊接梁
最优值	−30 665.3647	6059.7140	0.0127	1.7217
平均值	−30 665.3600	6238.8010	0.0127	1.7380
最差值	−30 370.5522	6820.4100	0.0130	1.9174
标准差	72.2230	194.3200	5.1×10^{-5}	0.0345

表 5-3　测试实例最优设计方案

问　　题		结　　果
Himmelblau's function	优化方案	(78,33,29.9959,44.9987,36.7746)
	约束值	(92,98.8409,20)
Pressure Vessel	优化方案	(0.8125,0.437 500,42.0984,176.6366)
	约束值	(0,−0.0359,0,−63.3634)
Spring	优化方案	(0.0517,0.3568,11.2862)
	约束值	(0,0,−4.0539,−0.7277)
Welded Beam	优化方案	(0.205 709,3.4484,9.0366,0.2057)
	约束值	(−0.0009,−0.0007,−3.9601,−3.4350,−0.0807,−0.2355,−0.0003)

为验证 iGSO 优化算法的收敛速度，本书将 iGSO 同 iPSO 和 iGA 这两个算法进行了对比。iPSO 和 iGA 分别是指将 iGSO 的约束处理方法附加在标准的 PSO 和 GA 上。对于压缩弹簧测试实例，iGA、iPSO 和 iGSO 3 个算法优化的结果分别是 0.015 892、0.013 442 及 0.012 664；对于压力容器测试实例，iGA、iPSO 和 iGSO 3 个算法优化的结果分别是 6924.1653、6311.9632 及 6059.7143。从实验数据可以看出，iGSO 的优化性能优于另外两个算法的优化结果。图 5-13 给出了 3 个算法优化这两个实例的迭代进化曲线。同 iPSO 和

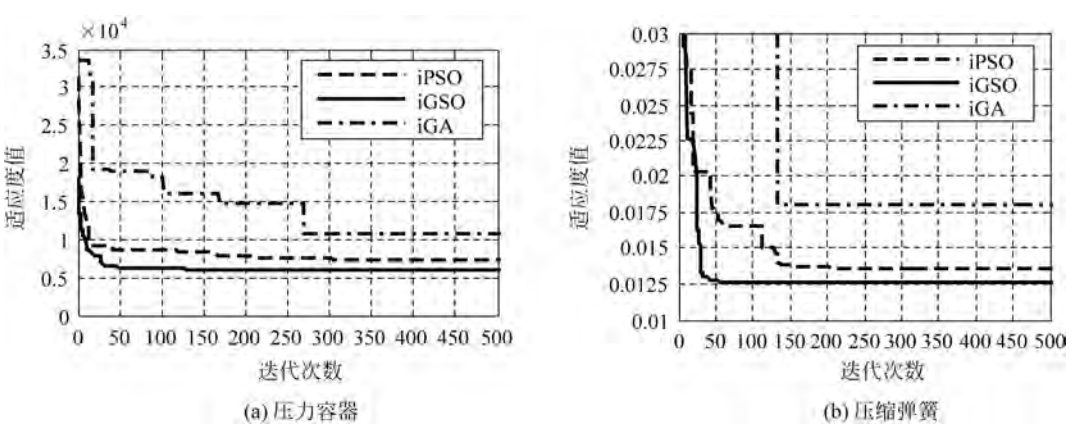


图 5-13　两个机械设计实例的收敛曲线比较图

iGA 相比，iGSO 的最优解值优化性能并不突出，但 iGSO 具有较高收敛速度，即可以以最快的速度找到最优解，在时间特性上有了比较好的改善。

5.4　基于生物群落演化的优化模型与算法实例设计

5.4.1　协同进化算法的发展现状

近几十年随着群落生态学的发展以及对多种群协同进化理论的深入研究，许多以协同进化为基础的进化算法相继被提出并得到了很好的优化效果。协同进化算法在进化算法的基础上，考虑了种群与环境之间、种群与种群之间在进化过程中的协调。虽然协同进化算法起步较晚，但由于其算法的优越性，目前已成为当前进化计算、群体智能领域的一个研究热点[7]。

根据生态学上物种间的捕食者与猎物(predator-prey)、寄生物与宿主(host-parasite)、相互竞争(competition)、互利共生(mutualism) 4 种关系，可以将协同关系划分为两类，即竞争关系和合作关系。因此协同进化算法同样也分为体现竞争关系的竞争型协同进化算法(Competitive Co-Evolutionary Algorithm)和体现合作关系的合作型协同进化算法(Cooperative Co-Evolutionary Algorithm)[8-11]。

在合作型协同进化算法方面，Potter 和 De Jong 是合作型协同进化算法的先驱，提出了用于函数优化问题的协同进化遗传算法(Cooperative Coevolutionary Genetic Algorithm，CGGA)[12,13]。考虑一个具有 N 个变量的待优化函数，CCGA 对每个变量维持一个种群，并用传统遗传算法对每个种群进行优化。由于各个种群的单个个体只是部分解，个体适应度的计算需要从各个种群选出个体构成问题的一个完整解来赋值。郑浩然等提出了多模式共生进化算法(Multi-Pattern Symbiotic Evolutionary Algorithm，MSEA)[16]，与 CCGA 算法不同之处在于：其一，MSEA 将优化函数变量分成几组，称为模式，各个模式采用不同的进化方式；其二，个体适应度的计算方法不同，对于某个种群内的个体 a，MSEA 首先从各个种群随机选取 n 个个体与 a 组成完整解，然后计算这个 n 个完整解的适应度，把这些解的适应度的平均值作为个体 a 的适应度。除此之外，刘静等对协同进化计算模型进行了分析，并提出了一种组织协同进化算法用于海量数据分类问题[14]。Seredynski 针对 N 人博弈问题提出了协同进化多智能体系统模型。Garcia-Pedrajas 等将合作型协同进化算法用于神经网络集成的设计，并通过一组真实的分类问题测试了其有效性[15]。

在竞争型协同进化算法方面，曹先彬、郑浩然等人[17-19]提出了基于生态竞争模型的协同进化算法。该算法把种群分成若干子种群，算法在每次迭代中都依次进行进化和协同过程，其中进化过程采用遗传算法的操作方法，协同过程通过种群竞争方程计算种群密度，并根据计算出的种群密度调整各个子种群的规模，从而实现根据适应度的情况动态地调整各个模式在种群中的比例的目的。随着协同进化理论在进化计算方面的应用越发成熟，本书在下面给出了基于自然界多群体群落共生协同进化的思想，集成了 3 种共生模式设计出的一种新型多群体协同统一优化模型，从而指导新型的多群体智能算法设计。

5.4.2　多群体协同进化统一模型

本节根据前面提到的互利共生、共栖、寄生 3 种种群之间的关系，设计了一个新型多群

体协同统一优化模型。在该优化模型中，基于主(Master)-从(Slave)式的结构来表示多群体群落，并基于有向加权图来定义群体之间的共生关系。整个群落被分为 N 个子种群，由 N_m 个主群(Master species)和 N_s 个从群(Slave species)组成，$N_m+N_s=N$。这里规定：

(1) 从群之间在进化时没有信息交流。

(2) 主群之间或主群与从群之间可以进行信息交流。

(3) 每个种群均包含相同的个体数，基于不同共生模式的种群间信息交流模式由集合 $I=\{\text{Mutualism}, \text{Commensalism}, \text{Parasitism}\}$ 定义，其中 3 种共生关系 Mutualism，Commensalism，Parasitism 分别由 Master $A \overset{+}{\longleftrightarrow}$ Master B，Slave $A \overset{+}{\longrightarrow}$ Master B，Slave $A \underset{-}{\overset{+}{\rightleftharpoons}}$ Master B 定义。

(4) 每个种群均可以应用不同的进化计算或群体智能优化算法作为进化规则。

由此建立的多种群共生协同进化优化模型如图 5-14 所示，图中箭头表示在不同共生关系下种群间的信息的传递。根据不同的主、从群数以及信息交流模式设置，该多种群共生协同进化统一模型可以表示不同的共生模式。

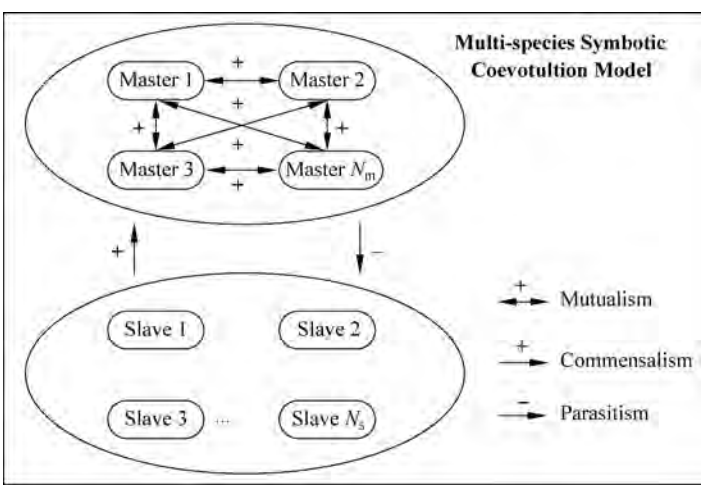


图 5-14　多种群共生进化模型

(1) $N_m=0, N_s=N, I\in\phi$：各种群独立进化，种群之间没有信息交流，即基于基本进化计算或群体智能算法的并行优化算法模型。

(2) $N_m=1, N_s=N-1, I=\{\text{Commensalism}\}$：主群接收多个从群传递的信息从而受益，从群之间无信息交流，即基于共栖模式的多种群协同进化。

(3) $N_m=1, N_s=N-1, I=\{\text{Parasitism}\}$：主群与从群直接具有信息交互，主群受益而从群受害，从群之间无信息交流，即基于寄生模式的多种群协同进化。

(4) $N_m=N, N_s=0, I=\{\text{Mutualism}\}$：所有的种群共享信息而互相受益，即基于互利共生模式的多种群协同进化。

5.4.3　多种群共生协同进化粒子群优化算法

根据 5.4.2 节提出的多群体协同进化统一模型，本节提出了多种群共生模式的协同进

化粒子群优化算法(Multi-species Symbiotic Particle Swarm Optimizer,MSPSO)。首先对3种模式下的MSPSO给出了一个统一的形式化描述。紧接着基于该形式化描述详细描述了3种多种群协同进化粒子群优化算法。3种算法分别是基于共栖关系的粒子群优化算法MSPSO-C、基于寄生关系的粒子群优化算法MSPSO-P和基于寄生关系的粒子群优化算法MOPSO-M。最后对3种算法的性能进行了实验比较。

1. 多群体共生协同进化粒子群算法的形式化描述

MSPSO算法的进化模型是由个体层、群体层和群落层构成。定义如下：

MSPSO=(Individual,Population,Colony,Environment,Topology,Coevolution)

1) 个体层

$$\text{Individual}=\{\text{Individual}_1,\text{Individual}_2,\cdots,\text{Individual}_i,\cdots,\text{Individual}_{\text{INUM}}\}$$

第i个个体表示为$\text{Individual}_i=<\text{IID}_i,\text{POSI}_i,\text{FITNESS}_i,\text{BHV}_i>$,它们分别表示个体$i$的序号、位置、适应度和行为集合。

2) 群体层

$$P=\{P_1,P_2,\cdots,P_M\}$$

其中,群体层的任一个种群可表示为Population=<OPT,PBHV,N>其中,OPT为群内最优个体；PBHV表示群内个体间行为,PBHV=$\{C^A\}$；N为种群大小。

3) 群落层

$$\text{Col}=\{\text{Col}_i^1\{\text{Col}_j^2\{\text{Col}_k^3\{\cdots\{\text{Col}_l^{\text{Num}}\}\}\}\}\}$$

4) 环境

Environment由优化函数定义。

5) 拓扑结构

$$\text{Topology}=(\text{TS},\{A_{ik},P_{jk}^n\})$$

其中,TS表示此算法静态拓扑结构；A_{ik}：表示第k个时间点个体i的邻域关系；P_{jk}^n表示第k个时间点第n层第j个群体的邻域关系。

6) 协同进化

$$\text{Coevolution}=\{\text{Sym}\},\text{Sym}=\{\text{unitA}_i\}=(\{\text{signa}\},\{\text{unitB}\},\{\text{signb}\})_i$$

对于算法的每一次迭代t,个体的进化规则C^A表示如下：

$$\alpha_{ik}^{t+1}=c_1r_1(\text{pbest}_{ik}^t-X_{ik}^t)+c_2r_2(\text{sbest}_k^t-X_{ik}^t)+B\beta_k^{t+1} \tag{5-2}$$

$$\beta_k^{t+1}=c_3r_3(\text{cbest}_p^t-X_k^t) \tag{5-3}$$

式(5-2)中α_{ik}^{t+1}表示个体层X_{ik}^t在下一时刻$(t+1)$的加速度；pbest_{ik}^t表示个体在解空间中的最优历史位置；sbest_{ik}^t表示个体层群体k中和个体X_{ik}^t有协作关系的个体在解空间中的当前最优位置；r_1和r_2为均匀分布在[0,1]区间的随机数；c_1和c_2为学习因子；$B\beta_k^{t+1}$表示个体X_{ik}^t从其他种群中所获得的最优位置信息。其中B为控制变量,$B=\{-1,0,1\}$。当$B=-1$时,表示个体X_{ik}^t所处种群与其他种群是受害关系,受到其他种群伤害。当$B=0$时,表示个体X_{ik}^t所处种群与其他种群是互不影响的关系。当$B=1$时,表示个体X_{ik}^t所处种群从其他种群获利。

其中式(5-3)中cbest_p^t表示群体层中和X_k^t所在种群有协作关系的其他种群最优位置信息；r_3为均匀分布在[0,1]区间的随机数；c_3为学习因子。则整个系统的进化可以表

示为：

$$V_{ik}^{t+1}=\chi(V_{ik}^{t}+\alpha_{ik}^{t+1}+B\beta_{ik}^{t+1}) \tag{5-4}$$

$$\text{if}(k\in e)\quad X_{ik}^{t+1}=r\times(u\text{b}-\text{lb})+\text{ub} \tag{5-5}$$

$$\text{else}\quad X_{ik}^{t+1}=X_{ik}^{t}+V_{ik}^{t+1} \tag{5-6}$$

式(5-5)表示所有物种(群体层的个体)承受同样的外部环境压力，当压力超过某一阈值时，一些物种将消失以缓解环境压力，新的物种将同时产生以增加系统的生物多样性。其中 e 表示将要灭绝的物种集合。基于上述形式化描述，下面给出了多种群共生协同进化粒子群优化算法的一个流程框架，如图 5-15 所示。

图 5-15　多群体共生协同粒子群优化算法框架

2. 多种群共栖粒子群算法 MSPSO-C

根据统一模型，此时 $N_{\text{m}}=1$，$N_{\text{s}}=N-1$，$I=\{\text{Commensalism}\}$。在每次迭代过程中，每一个从群执行标准的 PSO 算法进行种群内个体速度与位置更新，更新方程式如下：

$$v_{id}^{S_j}=wv_{id}^{S_j}+c_1r_1(p_{id}^{S_j}-x_{id}^{S_j})+c_2r_2(p_g^{S_j}-x_{id}^{S_j}) \tag{5-7}$$

$$x_{id}^{S_j}=x_{id}^{S_j}+v_{id}^{S_j} \tag{5-8}$$

在所有的从群在进行下一步状态更新之前，把到目前为止发现的最好位置信息发送给主群，然后主群根据自身位置信息和其他从群中最好位置信息进行状态的更新，更新方程式如下：

$$v_{id}^{M}=wv_{id}^{M}+c_1r_1(p_{id}^{M}-x_{id}^{M})+c_2r_2(p_g^{M}-x_{id}^{M})+c_3r_3(p_g^{Q}-x_{id}^{M}) \tag{5-9}$$

$$x_{id}^{M}=x_{id}^{M}+v_{id}^{M} \tag{5-10}$$

式中各变量的含义如下：

M：主群；

S：从群集合；

j：从群集合中的种群编号，$j=1,2,\cdots,N-1$；

Q：从群最好位置信息集合；

c_3：共生学习因子，通常取 2.0；

r_3：(0,1)之间的随机数；

p_g^Q：Q 中的最优位置信息。

其余变量的含义见基本 PSO 算法介绍。

可见，在多种群共栖粒子群算法的更新机制中，主群中每个粒子根据自己本身的最优历史经验、其隶属种群的当前最优经验以及所有从群的当前最优经验进行状态更新。

3. 多种群寄生粒子群算法 MSPSO-P

此时 $N_m=1, N_s=N-1, I=\{\text{Parasitism}\}$，主群与从群分别模拟寄生物与宿主。在每次迭代过程中，每一个从群因与发生信息交互而受害，其表现形式为远离主群当前最优位置所在的区域，更新方程式如下：

$$v_{id}^{S_j}=wv_{id}^{S_j}+c_1r_1(p_{id}^{S_j}-x_{id}^{S_j})+c_2r_2(p_g^{S_j}-x_{id}^{S_j})-c_3r_3(p_g^{M}-x_{id}^{S_j}) \tag{5-11}$$

$$x_{id}^{S_j}=x_{id}^{S_j}+v_{id}^{S_j} \tag{5-12}$$

在所有的从群进行下一步状态更新之前，把到目前为止发现的最好位置信息发送给主群，然后作为寄生物的主群根据自身位置信息和其他从群中最好位置信息进行状态的更新，更新方程式如下：

$$v_{id}^{M}=wv_{id}^{M}+c_1r_1(p_{id}^{M}-x_{id}^{M})+c_2r_2(p_g^{M}-x_{id}^{M})+c_3r_3(p_g^{Q}-x_{id}^{M}) \tag{5-13}$$

$$x_{id}^{M}=x_{id}^{M}+v_{id}^{M} \tag{5-14}$$

可见，在多种群寄生粒子群算法的更新机制中，主群粒子追踪从群粒子加速了向全局最优解靠近，从群粒子受主群粒子排斥跳出局部最优点，逃离到新的搜索区域。这样的寄生进化模式能够有效保持种群多样性，克服了基本粒子群算法“趋同性”早熟收敛问题。

表 5-4　MSPSO 算法伪代码

```
Algorithm MSPSO
Begin
  Initialize the population                    //主群和从群
  Evaluate the fitness value of each particle in each swarm
  Repeat
    Do in parallel                             //从群进化
      Swarm i, 1≤i≤N_s
    End Do in parallel
    Barrier synchronization                    //等待所有进程完成
      Select the fittest global individual p_g^s from all the slave swarms
    Do in parallel                             //主群进化
      Swarm j, 1≤j≤N_m
    End Do in parallel
    Barrier synchronization
      Evolve the slave and master swarm
    Evaluate the fitness value of each particle in each swarm
  Until a terminate-condition is met
End
```

4. 多种群互利粒子群算法 MSPSO-M

此时 $N_m=N, N_s=0, I=\{\text{Mutualism}\}$，所有种群共享各自发现的最优位置信息而互相受益。在每次迭代过程中，每个种群更新方程式如下：

$$v_{id}^{M_j} = wv_{id}^{M_j} + c_1 r_1 (p_{id}^{M_j} - x_{id}^{M_j}) + c_2 r_2 (p_g^{M_j} - x_{id}^{M_j}) + c_3 r_3 (p_g^{L} - x_{id}^{M_j}) \quad (5\text{-}15)$$

$$x_{id}^{M_j} = x_{id}^{M_j} + v_{id}^{M_j} \quad (5\text{-}16)$$

公式中各变量的含义如下：

- M——主群集合；
- j——主群集合中的种群编号，$j=1,2,\cdots,N$；
- L——主群最好位置信息集合；
- p_g^L——L 中的最优位置信息。

可见，多种群互利协同进化过程不仅加快了算法的收敛速度，而且有效地保持了整个系统的多样性，在克服“趋同性”早熟收敛问题的同时，显著提高了算法的收敛精度与收敛速度。

上述 3 种版本多群体共生粒子群优化算法可统一由表 5-4 的 MSOPSO 算法伪代码表述。

5.4.4 算法性能分析

1. 实验设置

为测试 3 个版本 MSPSO 算法的性能，选择了常用的一组标准测试函数，即 Sphere()、Rosenbrock()、Ackley()、Rastrigrin()、Griewank()函数。其中，Sphere()、Rosenbrock()、Ackley()为单峰函数，Rastrigrin()、Griewank()为多峰函数。

为了检验 MSPSO 模型的性能，本节设计了两个实验，分别为算法优化性能实验和多种群协同进化动态特性实验。第 1 个实验用于检验 3 个版本 MSPSO 算法的综合优化（全局、局部搜索能力）性能，并与基本粒子群算法 PSO 进行比较分析；第 2 个实验用于深入分析 MSPSO 模型中不同共生关系的种群进化动态特性。第 1 个实验选用了上述 5 个测试函数，第 2 个实验选用了单峰函数 Rosenbrock()和多峰函数 Rastrigrin()、Griewank()。

PSO 参数设置为 $V_{\max}$、$V_{\min}$ 为自变量边界范围的一半，c_1 和 c_2 设为 2.0，惯性权重设置为随着迭代次数的增加从 0.9 线性减小到 0.4。对于 3 个版本的 MSPSO 算法，3 个学习因子均取值为 1.494，惯性权重设为 0.729。在所有的试验仿真中算法的群体规模设置为 80，MSPSO 分为 2 个子种群，每个子种群的群体数都为 40。算法终止的条件为满足最大迭代次数 1000。每个算法独立运行 30 次。

2. 实验 1：函数优化

5 种算法对所有 30 维测试函数的运行 30 次的平均值见表 5-5，所有算法针对每个函数的优化过程收敛曲线比较见图 5-16。

对于较为简单只有一个全局最优解的单峰函数，收敛速度是考查优化算法性能的重要指标。从图 5-16(b)可以观察到，所有的算法都能迅速收敛到 Sphere()函数的全局最优解。但是，3 种协同进化算法比基本 PSO 算法具有更快的收敛速度，这体现了共生协同进化模型中不同种群之间的信息交流显著提高了算法的收敛速度。Rosenbrock()函数被称为“香蕉问题”，其全局最优解隐藏在了一条狭长扁平的通道中。发现通道并不是很难，难的是发现通道中的最优点。对于 Rosenbrock()函数，协同进化算法仍然要由于 PSO 算法。其中，MSPSO-C 以最快的收敛速度达到了该函数的次优解。说明 MSPSO-C 模型中主群通过从

群的全局最优信息，增强了整个群体的寻优能力。

函数 Ackley()是一个相对简单的多峰函数，从优化结果可以看出，所有算法都收敛到了全局最优解，但是协同进化算法仍然表现出了突出的收敛速度，这从另一个方面体现了协同进化算法的鲁棒性。

函数 Rastrigrin()是复杂的多模态函数。协同进化算法与 PSO 算法的最终收敛结果基本趋同，但协同进化算法的收敛速度仍然是优于 PSO 算法的。

对于最后一个复杂多峰函数 Griewank()，MSPSO-P 体现出了优异的性能。从图 5-15 可以看到，PSO 算法在 500 代后趋于停滞，陷入局部最优，其收敛曲线表现为一条直线。而 MSPSO-P 以非常快的速度向优优解收敛，并在 750 代附近定位到函数的全局最优。

通过对表 5-5 及图 5-16 的研究，可以得出本章提出的多群体共生协同进化方法是一类性能优越的优化算法，与经典 PSO 算法相比，它在收敛速度、求解精度与稳定性等方面都有很大程度的提高。因此，有理由相信 MSPSO 的突出优化能力来源于在进化过程中多群体间的共生机制对算法多样性保持起到了很大作用。

表 5-5　所有算法的测试结果

函数 / 算法	MSPSO-C	MSPSO-P	MSPSO-M	PSO
Sphere()	1.8453e－044	2.1518e－036	5.9234e－031	2.2973e－012
Rosenbrock()	13.3491	13.9419	0.1975	24.4461
Ackley()	9.2371e－015	6.9870e－015	7.3423e－015	2.3529e－005
Rastrigrin()	44.7068	32.5020	36.2172	28.4593
Griewank()	0.0084	0	2.4653e－004	0.0183

3. 实验 2：协同进化仿真

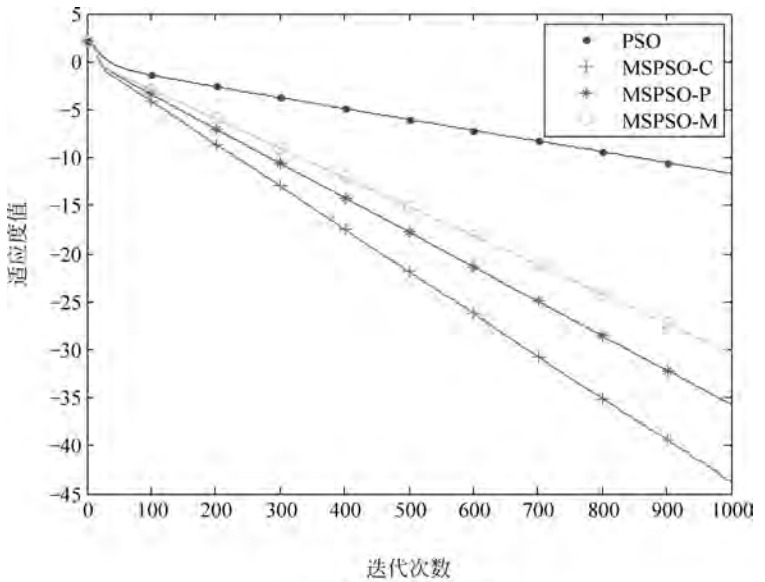
为了进一步分析 MSPSO 模型中多群体进化的动态特性，图 5-17～图 5-19 给出了基于 3 种共生机制的算法在测试函数定义的环境中的协同进化过程。可以看到，3 种算法都有效地模拟了自然界中的共生模式。

对于模拟共栖模式的 MSPSO-C 模型，Host 种群从多数图中可以观察到从群在迭代初期迅速收敛，在迭代后期一般会产生早熟收敛停滞不前；而 Commensal 种群通过接收 Host 种群的最优解信息，在迭代初期迅速收敛到最优解区域，进而在迭代后期定位全局最优解。

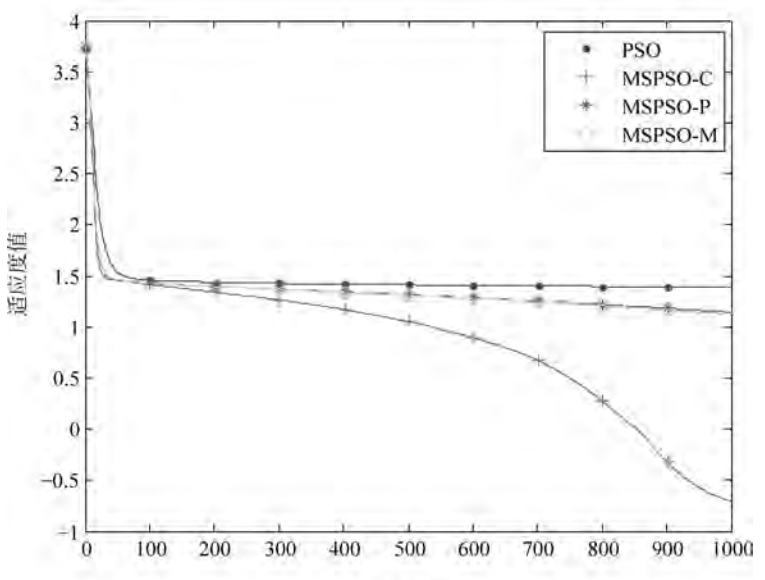
对于模拟寄生模式的 MSPSO-P 模型，由于竞争关系，在迭代初期收敛缓慢，这是因为 Host 种群通过逃离 Parasite 种群所在的区域，从而能够进行大范围最优值探索；而 Parasite 种群一方面对自己种群发现的最优解区域进行开发，另一方面又可以追踪 Host 种群发现的最优解区域，从而能够在迭代后期定位更为优秀的最优值。

对于模拟互利共生模式的 MSPSO-M 模型，两个种群共享各自发现的最优解信息，因此两个种群均能够以最快的收敛速度发现全局最优或次优解。

综上，通过对自然界共生现象的模拟，本章提出的多群体协同进化优化算法能够有效保持整个群体的多样性，在一定程度上平衡了算法的探索能力和开发能力，使其优化性能得到提高。

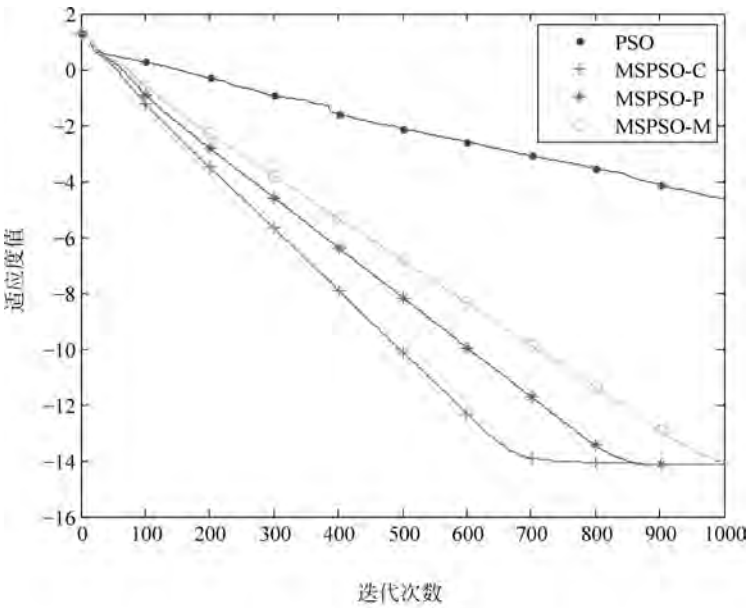


(a) Rosenbrock()函数

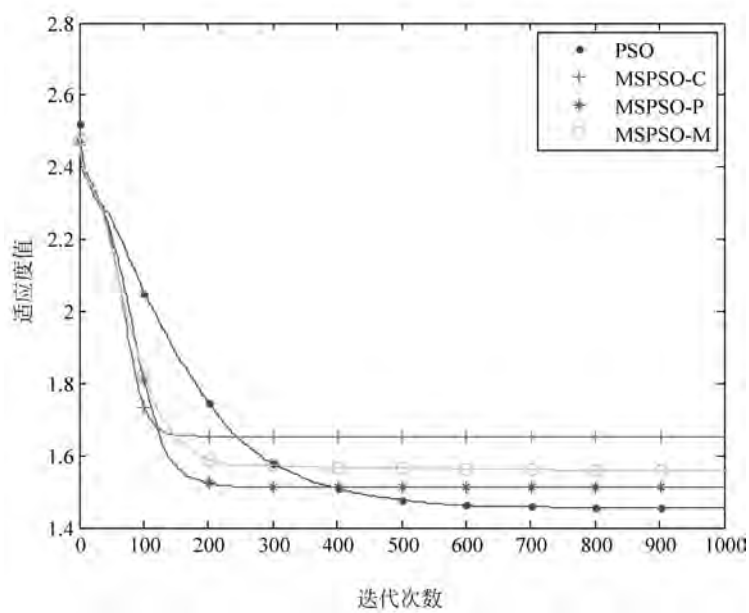


(b) Sphere()函数

图 5-16　测试函数的收敛曲线比较图

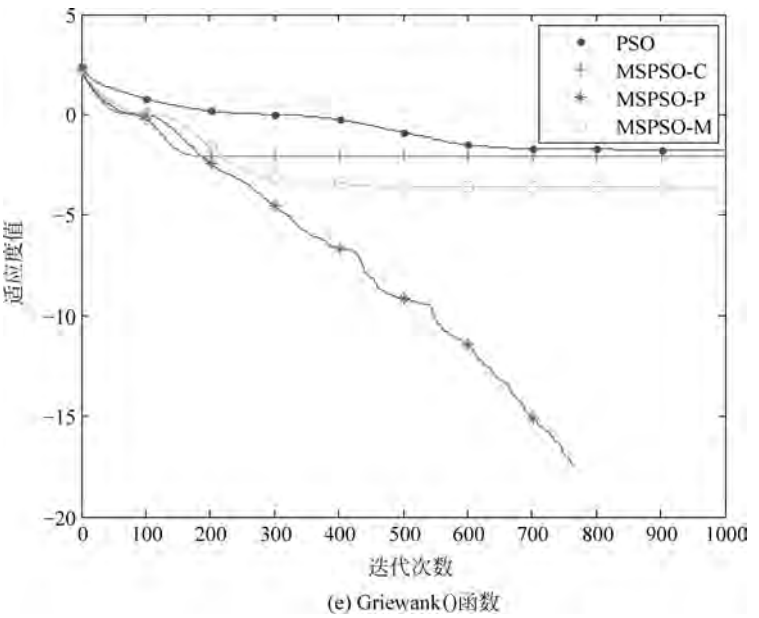


(c) Ackley()函数



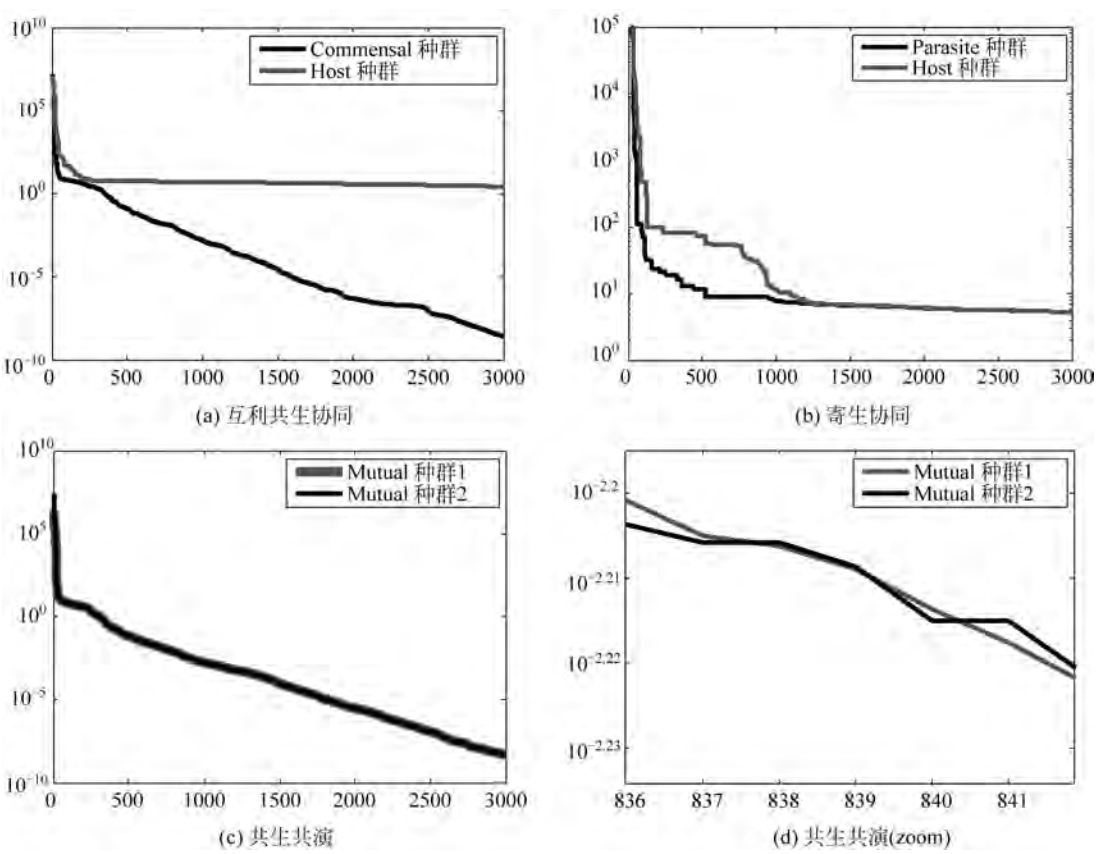
(d) Rastrigin()函数

图 5-16　（续）



(e) Griewank()函数

图 5-16 （续）



(a) 互利共生协同　(b) 寄生协同

(c) 共生共演　(d) 共生共演(zoom)

图 5-17　Rosenbrock 的协同进化过程

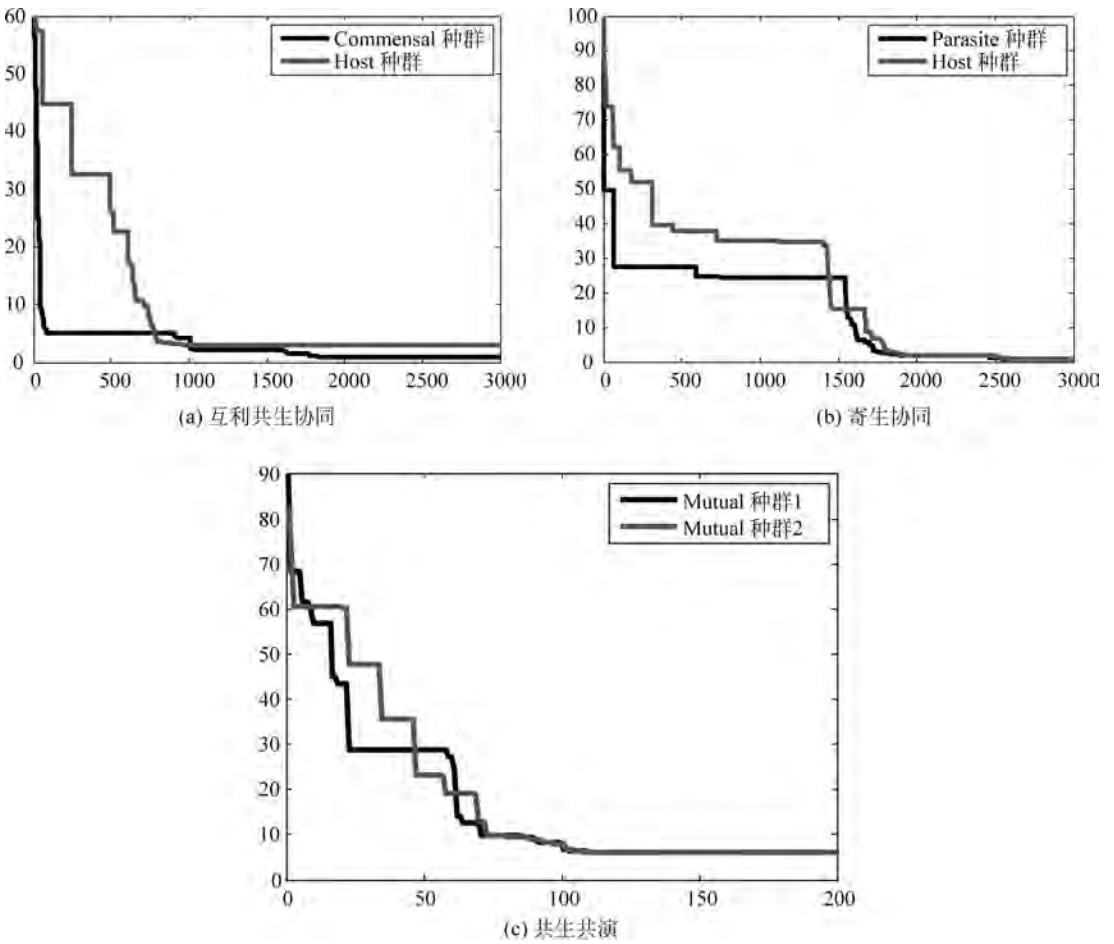


图 5-18　Rastrigrin 的协同进化过程

5.4.5　基于 MSPSO 的 RFID 网络读写器调度

1. RFID 网络读写器调度问题背景

在大规模 RFID 应用中，在工作区域内部署多个 RFID 读写器，构成密集读写器网络，其拓扑结构通常如图 5-20 所示[5]。网络中每个 RFID 读写器分别对其识读区域内的电子标签进行读写，并将采集到的标签数据发送到中央控制系统进行处理。由于密集读写器网络要对物理环境中海量标签进行全方位覆盖（防止漏读），某些读写器不可避免地会发生识读区域的相互重叠情况。为了便于说明，图 5-20 给出了密集读写器网络环境下的读写器冲突情况。即图 5-20 中给出了基于损耗模型的读写器电磁波分贝值，每个黑色圆周代表一个读写器的识读区域，白色圆周代表读写器发生频率干扰的区域，圆点代表相应的读写器。如果两个读写器的识读区域或者频率干扰区域发生相互重叠，就会产生读写器冲突，甚至使整个 RFID 系统无法正常工作。

2. RFID 网络读写器调度模型

RFID 网络的读写器冲突可以用图论方法描述。对于有 n 个读写器的 RFID 网络，由干扰图 $G=(V,E)$ 表示其读写器冲突。$V(g)=\{v_1,v_2,\cdots,v_n\}$ 代表网络中读写器集合称作顶

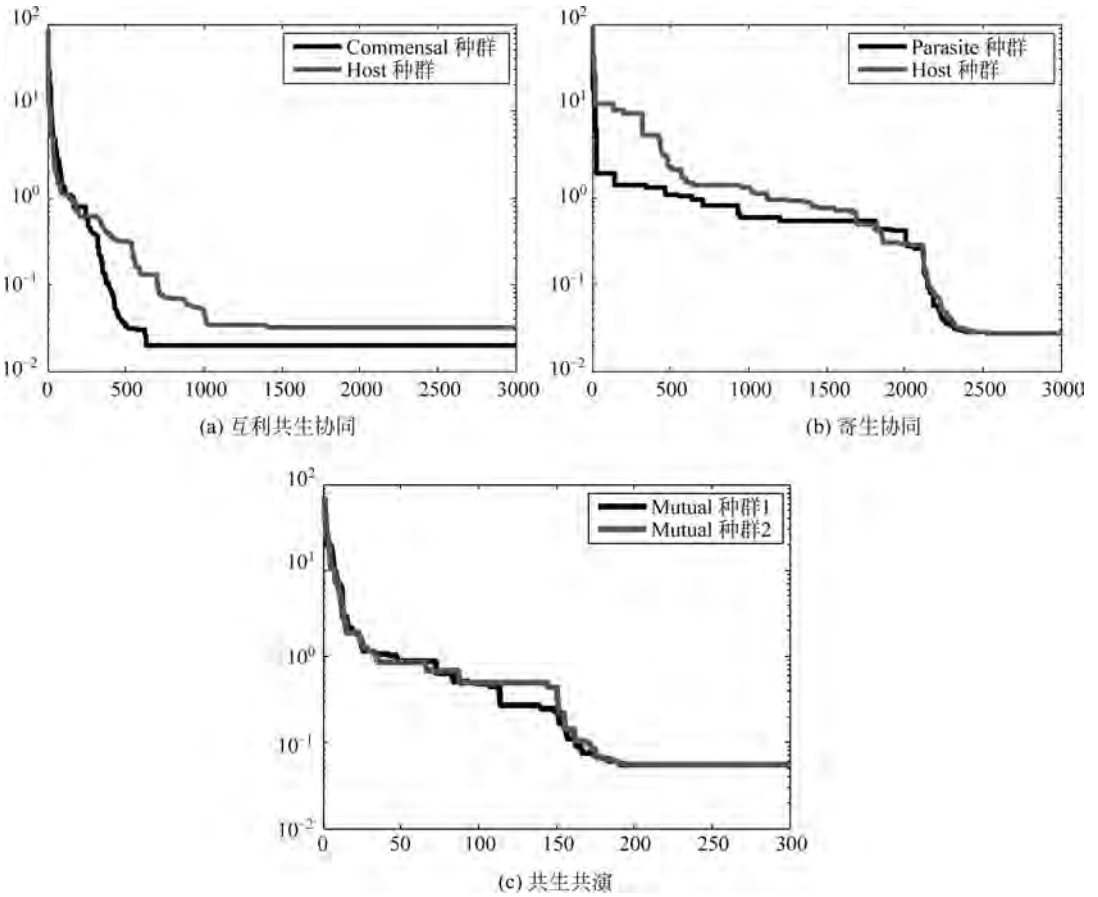


图 5-19　Griewank 的协同进化过程

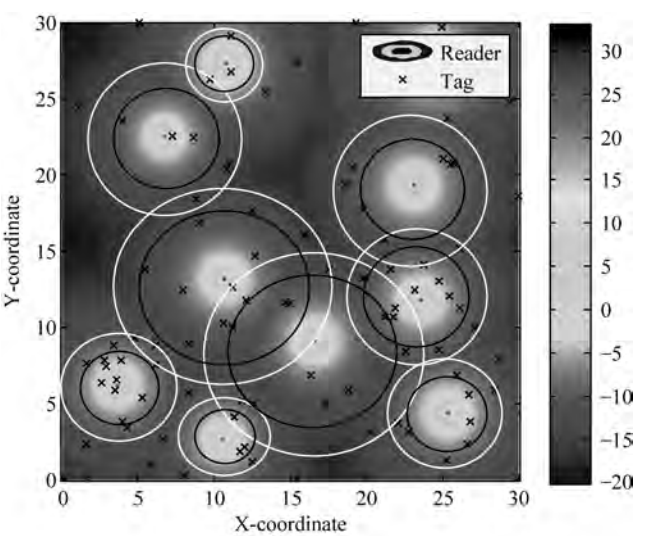


图 5-20　RFID 读写器冲突

点集合，$E(g)=\{e_1,e_2,\cdots,e_n\}$代表所有读写器之间的冲突关系称作边集。对于如图 5-21(a)所示的 RFID 网络，其对应的干扰图 G 如图 5-21(b)所示。为此，RFID 网络的读写器冲突问题可以简化为在图 G 上求解图分割问题。

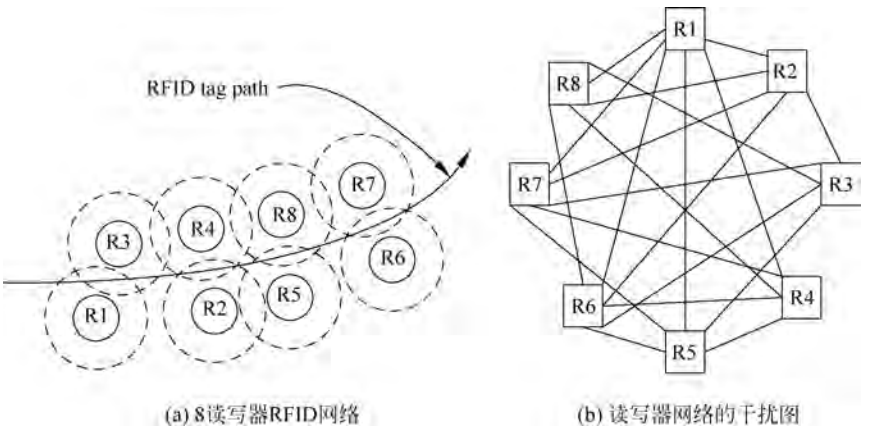


图 5-21　RFID 读写器网络

下面面向大规模 RFID 网络环境下的读写器调度问题，综合考虑读写器频率冲突和网络总处理时间等要素，定义调度规则：

(1) 在每个时隙中，始终最大化 RFID 读写器网络子图的读写器数量，从而尽量减少整个调度过程中的时隙总数。

(2) 尽量最小化每个时隙中 RFID 读写器网络子图的读写器处理时间方差，从而减少整个网络的总处理时间。

(3) 在每个时隙中尽量选择冲突关系最多的读写器，从而降低下一时隙调度的复杂性。

基于上述规则建立 RFID 网络读写器调度的数学优化模型如下：

$$\text{Max } f(s^t)=w_1 f_\alpha(s^t)+w_2 f_\beta(s^t)+w_3 f_\gamma(s^t)+w_4 f_\lambda(s^t)-\eta f_p(s^t) \quad (5\text{-}17)$$

模型中符号含义如下：

s^t——在时隙 t 中能够共同工作的 RFID 读写器网络子图，即可行解；

f_α——RFID 读写器网络子图 s^t 中读写器的最大处理时间；

f_β——RFID 读写器网络子图 s^t 中的读写器数量；

f_γ——RFID 读写器网络子图 s^t 中的读写器与其他读写器冲突数量和；

f_λ——RFID 读写器网络子图 s^t 读写器的平均处理时间；

f_p——惩罚函数；

w_1,w_2,w_3,w_4,η——不同子目标的权重。

3. RFID 网络读写器调度模型的 MSPSO 算法求解

对于 RFID 网络优化问题，一般难以采用传统最优化方法来进行精确求解。下面应用 MSPSO 优化算法来求解上面提出的 RFID 网络读写器调度模型。

Step1：编码

选择区间{0,1}上的二进制编码，RFID 网络读写器调度模型的编码表示为 $s^t=(s_1^t,s_2^t,\cdots,s_{n(t)}^t)$，这里 $n(t)$表示时隙 t 的 RFID 网络调度子图中的读写器数量。例如，可行解[0 1 0 0

1 0]，第 1 位值为零，表示读写器 1 在当前时隙 t 为休眠状态；第 2 位值为 1，表示读写器 2 在当前时隙为工作状态。由于不同的时隙会分配不同数量的读写器，因此在整个调度过程中，可行解的维度是动态变化的，如图 5-22 所示。

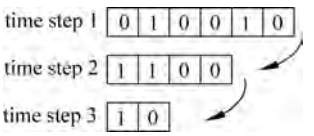


图 5-22　RFID 网络读写器调度的二进制编码及其动态变化

Step2：初始化

初始化 RFID 读写器网络，具体包括读写器的读写距离以及干扰范围、处理时间、网络中读写器数量。基于上述生成 RFID 读写器网络干扰图如图 5-23 所示。

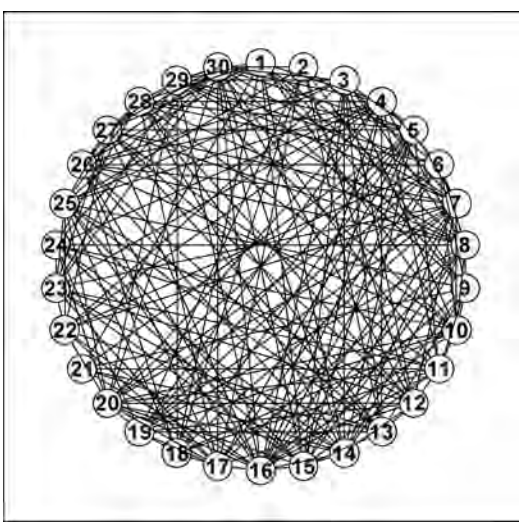


图 5-23　30 读写器的 RFID 读写器网络

初始化 MSPSO 算法种群，确定种群规模、共生进化模式、迭代次数、收敛精度、惯性权重、最小速度、最大速度和学习因子等参数。其中，粒子的维度由当前 RFID 网络工作读写器的数量确定。

Step3：粒子的更新

由下式检测每个粒子适应度值大小：

$$f(s_i^t)=w_1\max P(s_i^t)+w_2\sum_{j=1}^{n(t)}s_{ij}^t+w_3\sum_{j=1}^{n(t)}C(s_{ij}^t=1)+w_4\sum P(s_i^t)\Big/\sum_{j=1}^{n(t)}s_{ij}^t-\eta\sum_{j=1}^{n(t)}s_{ij}^t \tag{5-18}$$

这里数组 $P(\cdot)$代表可行解中在时隙 t 的读写器工作数量；$C(\cdot)$代表每个读写器与网络中其他读写器的冲突数量和，这里要求 $w_1\gg w_2\gg w_3\gg w_4$；η 取值为 $1000\times n(t)$。

更新共生种群的速度，选择个体极值、种群极值和全局极值。这里由于 RFID 网络调度问题是离散问题，根据下式对粒子位置进行离散操作：

$$\text{if}(\text{rand}()<S(v_{\text{id}}))\quad \text{那么}\quad x_{\text{id}}=1;\quad \text{而且}\ x_{\text{id}}=0 \tag{5-19}$$

$$S(v_{id})=\frac{1}{1+\exp(-v_{id})} \tag{5-20}$$

Step4：终止准则

本书将最大迭代次数设定为终止准则。如果本次迭代后仍有读写器未完成操作，则在RFID干扰图中删除本时隙中工作的读写器子图，转到Step3；否则，输出RFID网络读写器调度结果，包括RFID网络总处理时间、调度时隙以及每个时隙中的RFID网络子图等。

基于MSPSO其求解程序流程如图5-24所示。

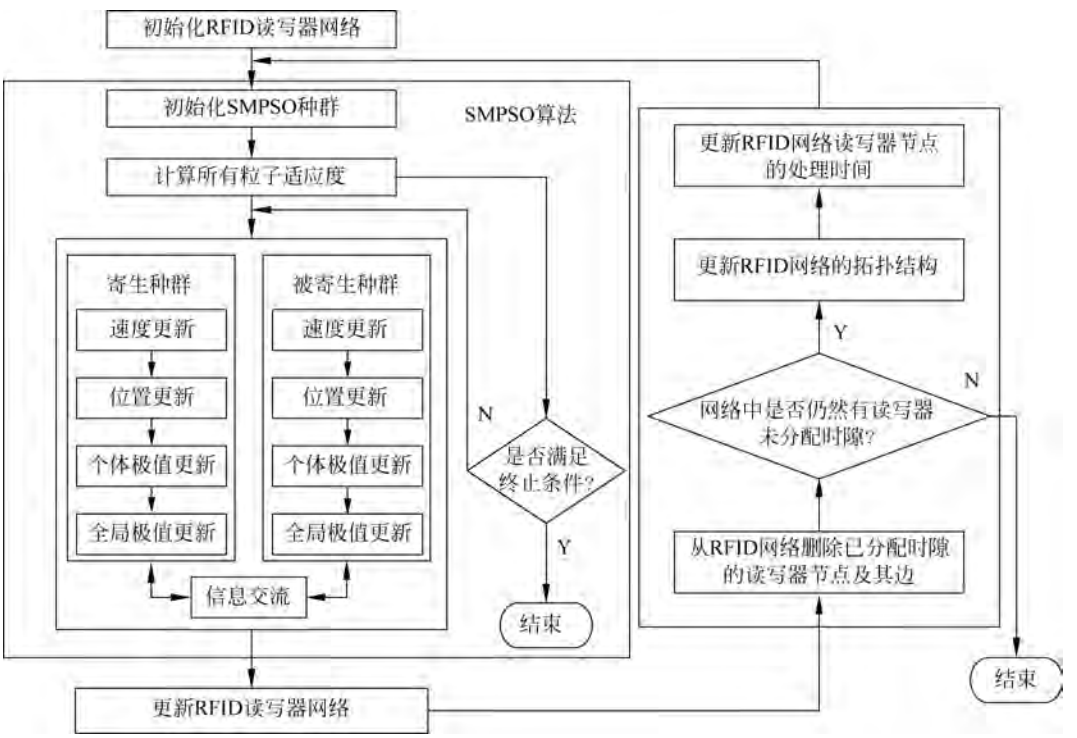


图5-24　基于MSPSO的RFID读写器网络调度求解流程

4. 实例研究

本节将应用MSPSO算法求解4个不同规模的RFID网络。4个RFID网络的读写器数量分别为30、60、120和200。每个读写器的处理时间随机生成并限制在[0,20]区间内，4个RFID网络的处理时间列表见表5-6～表5-9。图5-25给出了30个读写器网络和60个读写器网络的干扰图，受绘图软件条件限制，这里未给出其他网络的干扰图。

在应用MSPSO时，选用优化性能比较全面的互利共生算法模型。整个种群规模设置为120，即两个互助种群的大小都等于60，学习因子c_1、c_2和c_3均取值为2.0。为了保证比较的MSPSO算法求解RFID网络读写器调度问题的性能，本节选用二进制PSO算法和GA算法进行比较分析。

表5-10给出了3种算法在30次试验中在满足读写器间干扰约束条件下的调度结果。图5-26给出了3种算法的收敛曲线。从调度结果可以看出，MSPSO算法在4个RFID读写器网络调度实例中都取得了理想的效果。与其他两种算法相比，MSPSO算法具有更快的收敛速度、更精确的调度结果和更好的鲁棒性等性能。另一方面，随着RFID网络规模的

表 5-6　读写器处理时间(30 个读写器)

13.525 48	17.193 68	4.624 877	5.616 217	1.220 905	8.042 459	3.640 804	6.416 281	10.840 12	8.071 299
8.898 302	4.016 561	8.902 058	9.368 979	7.288 725	16.819 76	13.928 32	12.779 21	5.299 241	1.819 877
15.498 09	10.311 67	1.081 713	4.626 878	9.942 46	14.739 37	17.927 75	19.449 54	11.048 17	10.307 37

表 5-7　读写器处理时间(60 个读写器)

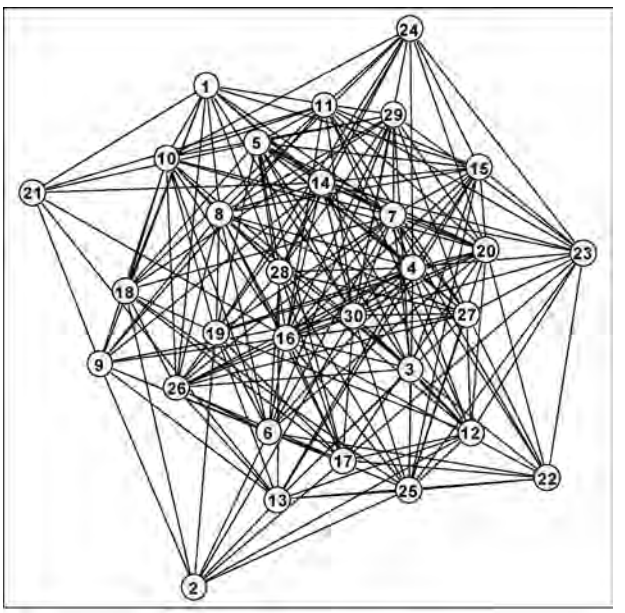
5.543 611	3.299 203	10.418 02	5.142 341	5.464 432	18.014 54	11.814 97	17.908 01	14.166 76	16.729 32
12.310 15	18.491 24	9.633 878	6.320 432	13.945 36	5.140 527	6.097 433	5.499 538	9.460 382	18.855 79
15.545 54	14.452 19	16.096 47	19.013 06	5.864 725	18.696 19	2.499 671	4.364 77	15.566 24	18.676 56
3.777 875	6.334 708	16.131 14	1.777 857	14.358 88	4.237 007	2.106 325	7.151 49	2.482 952	19.243 36
11.992 56	8.972 224	10.204 72	1.848 974	15.782 88	11.716 23	9.195 118	15.789 27	5.535 629	14.144 83
7.232 996	19.879 31	18.454 99	5.360 257	14.911 12	19.967 42	7.555 89	19.228 92	19.017 77	15.846 04

表 5-8　读写器处理时间(120 个读写器)

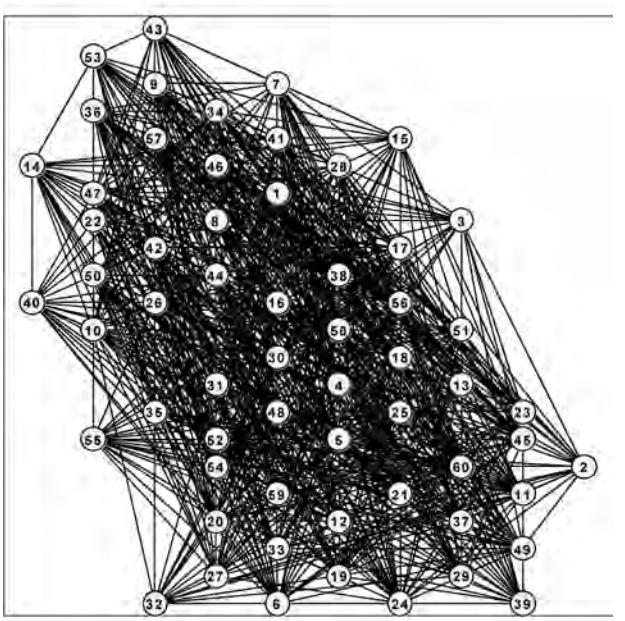
13.982 57	4.4484	5.974 297	7.935 937	13.601 43	12.900 53	13.601 93	14.755 38	17.5928	19.7617
13.9321	4.086 228	12.848 48	2.515 656	5.482 441	5.308 871	10.321 92	15.870 39	3.286 298	17.238 89
5.372 417	17.518 21	13.923 63	15.082 42	13.7519	17.353 74	10.314 18	5.754 558	6.147 715	11.669 81
12.018 92	15.555 64	3.164 29	2.158 847	7.388 128	7.212 494	5.160 591	13.207 79	9.989 83	14.539 07
1.226 096	15.769 41	11.6008	13.037 84	15.251 44	5.102 305	9.964 459	15.759 02	4.039 513	16.179 76
17.7784	19.254 35	14.838 12	16.474 41	17.299 81	2.439 073	13.866 97	11.241 21	13.757 59	9.012 546
7.545 015	10.265 41	13.185 83	1.499 178	15.217 03	9.717 298	13.739 72	13.643 46	19.183 62	19.262 57
17.830 73	7.748 361	9.325 252	18.821 83	6.138 582	19.5999	14.569 07	4.871 072	19.242 48	1.148 687
17.201 79	2.780 121	4.850 948	8.227 206	19.0092	14.350 27	2.675 843	8.539 25	10.955 31	1.916 034
17.013 64	15.847 76	12.093 49	12.402 82	19.017 65	10.624 42	8.520 499	2.857 366	9.750 608	19.170 48
11.897 35	17.282 69	16.749 27	17.5292	14.452 48	19.130 23	10.417 55	18.037 76	8.508 594	3.383 246
2.795 132	9.983 064	2.198 696	11.514 67	11.244 12	17.454 01	15.471 54	17.975 64	1.835 638	11.088 76

表 5-9　读写器处理时间(200 个读写器)

1.518 84	3.783 418	9.166 71	16.150 32	15.420 92	9.695 181	17.895 96	6.627 435	19.544 48	19.419 64
14.045 16	12.483 98	7.276 353	18.338 11	17.150 27	6.256 508	11.446 41	6.7131	19.110 12	10.354 65
10.675 72	15.198 58	19.044 91	15.832 13	18.2155	13.819 67	5.250 767	2.992 365	9.582 196	5.045 153
3.390 219	15.639 22	4.137 562	14.185 82	14.786 62	2.160 091	17.033 11	17.6848	14.141 97	7.375 756
14.824 29	6.585 405	4.978 127	12.480 34	3.850 949	14.207 21	2.070 146	15.320 81	19.646 52	15.132 17
15.112 63	12.145 89	4.996 854	8.818 109	17.262 91	4.404 496	8.034 467	19.271 48	17.147 36	7.361 117
6.581 828	10.420 16	2.470 441	13.641 71	19.543 84	7.292 233	4.692 136	12.644 39	6.927 223	11.286 51
5.548 938	10.558 24	8.039 744	19.327 74	12.740 11	9.772 256	16.7461	6.274 774	2.448 462	11.968 55
8.178 512	11.9926	6.488 77	9.215 711	6.448 42	13.158 82	1.767 648	15.668 63	18.030 95	14.071 18
8.697 67	13.941 34	6.136 621	1.387 83	17.549 13	10.081 52	18.4847	18.7378	2.187 869	10.861 69
12.2176	18.050 43	11.086 46	19.099 83	10.763 94	19.632 97	7.252 18	14.610 82	17.354 01	14.444 36
15.556 92	5.451 374	8.891 699	16.850 57	2.485 426	4.889 247	18.185 03	7.560 418	11.832 38	15.241 88
4.701 646	4.764 26	4.482 432	2.959 989	15.994 57	2.805 077	13.111 08	6.021 978	13.283 84	18.4006
19.239 01	13.872 65	4.680 523	13.320 25	2.246 156	5.586 437	10.693 72	11.785 44	13.9436	8.712 922
3.700 088	14.699 56	19.423 34	3.858 59	17.290 46	17.638 59	16.2631	5.591 866	16.081 58	16.559 67
1.985 547	12.557 36	15.0847	12.306 87	13.1604	18.429 45	11.638 55	3.794 11	18.507 18	10.460 88
2.952 955	17.287 72	19.142 49	13.501 76	12.719 83	7.865 552	19.690 87	11.330 29	19.953 07	16.571 86
9.572 096	13.6778	11.186 27	10.3269	18.4919	10.687	12.4907	1.739 175	18.254 81	8.458 165
3.456 293	4.764 416	2.808 688	14.110 06	5.736 131	19.877 46	7.706 997	15.702 69	15.5875	15.330 49
12.711 46	18.341 36	3.406 744	10.240 63	18.137 59	19.805 22	2.448 031	16.402 22	3.148 019	16.575 16



(a) 30个读写器的网络



(b) 60个读写器的网络

图 5-25　RFID 网络干扰图

表 5-10　RFID 网络调度结果

	MSPSO			PSO			GA		
读写器数量	最优值	平均值	标准差	最优值	平均值	标准差	最优值	平均值	标准差
30	9	7.443	0.8001	7	8.633	0.6149	8	9.100	0.7589
60	12	13.222	1.3301	13	14.367	0.8087	14	15.267	0.6397
120	21	20.178	1.1891	22	23.900	0.9595	23	24.833	1.0199
200	38	36.990	1.0056	39	41.233	1.3566	35	36.833	1.2888

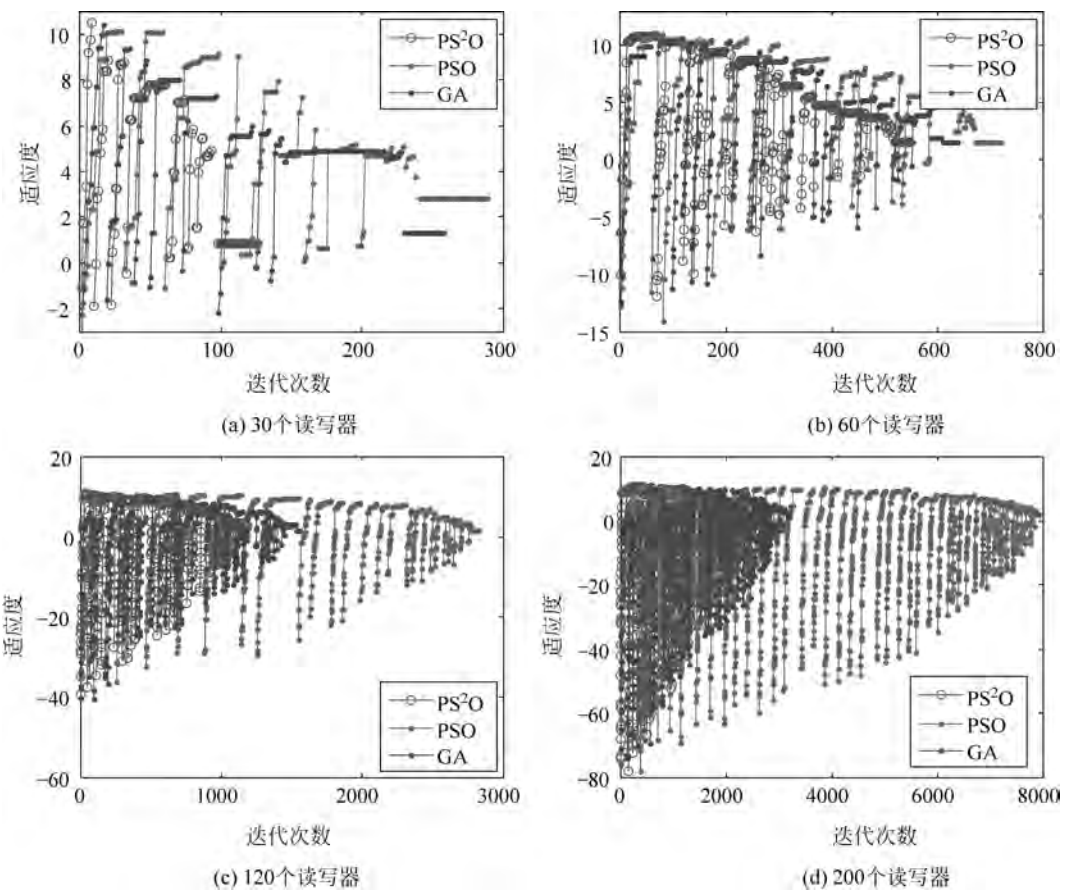


图 5-26　算法收敛曲线

提升，PSO 和 GA 算法的求解性能逐渐下降，而 MSPSO 算法性能却逐渐提高。可见这里提出的方法更具有求解大规模 RFID 网络调度问题的潜力。

为了更清晰地表现出 RFID 网络调度方法的求解过程，图 5-27 中的时间片 1～12（Time Step 1～Time Step 12）给出了 MSPSO 算法对 30 个读写器 RFID 网络进行调度时的网络干扰图变化过程。

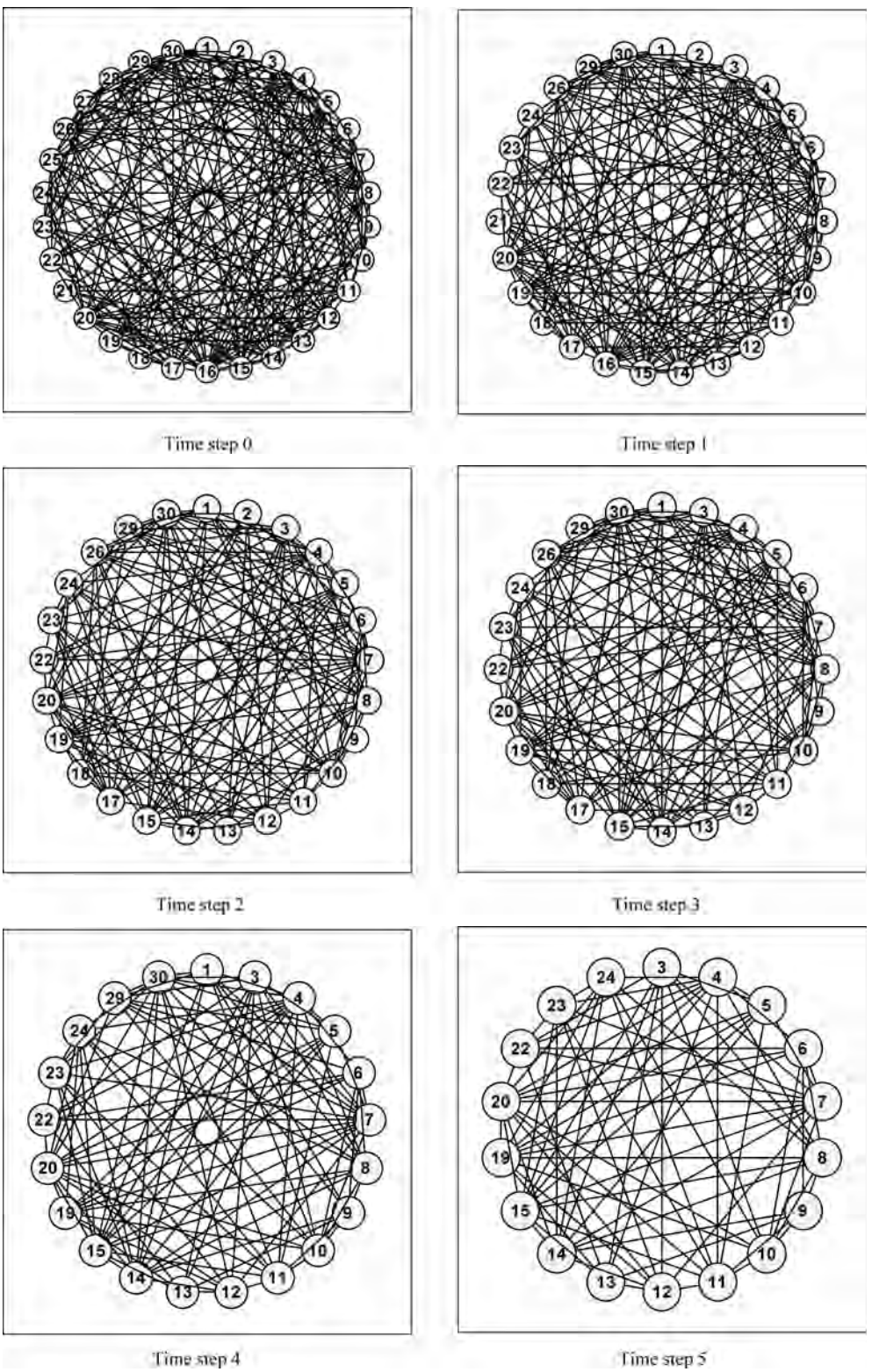
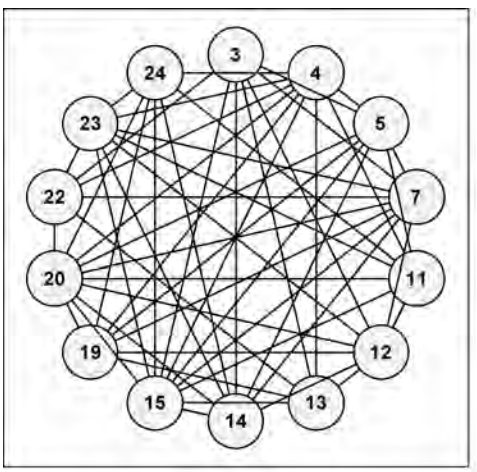
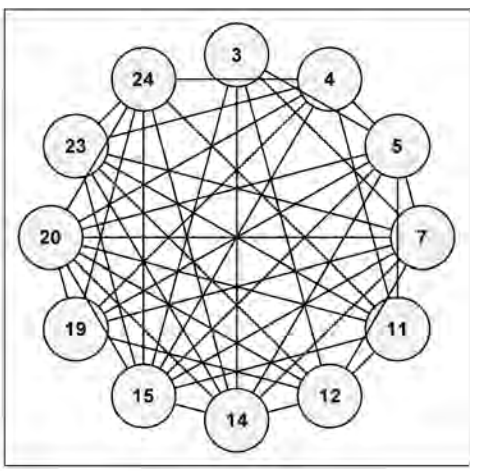

Time step 0
Time step 1
Time step 2
Time step 3
Time step 4
Time step 5

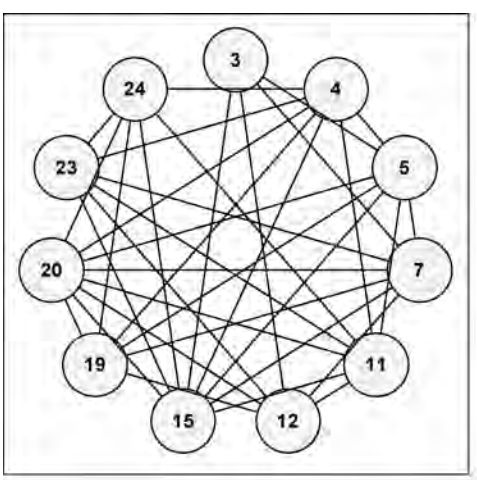
图 5-27　网络调度过程



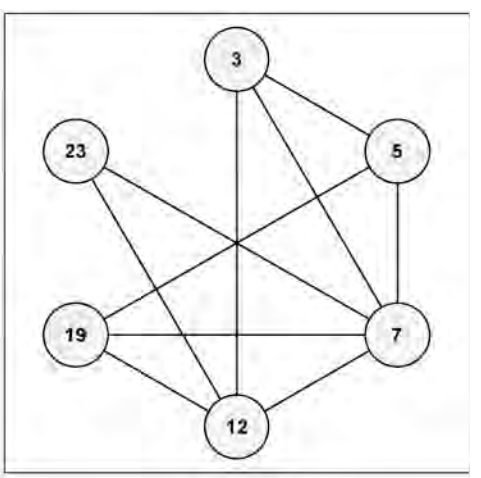
Time step 6



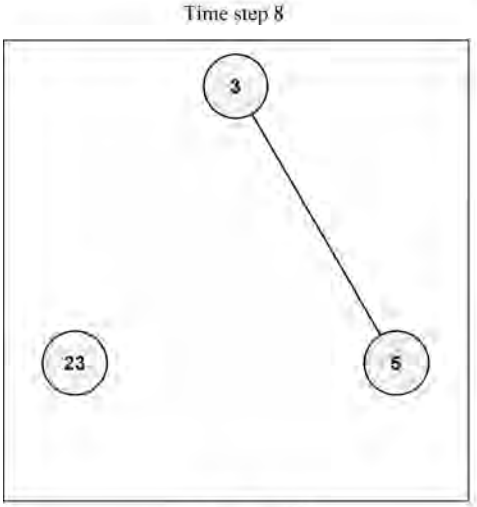
Time step 7



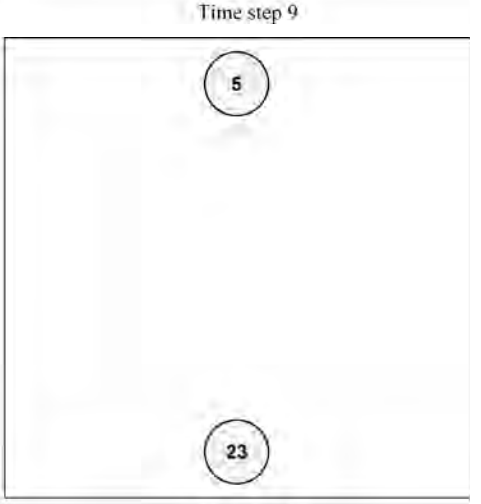
Time step 8



Time step 9



Time step 10



Time step 11

图 5-27　（续）

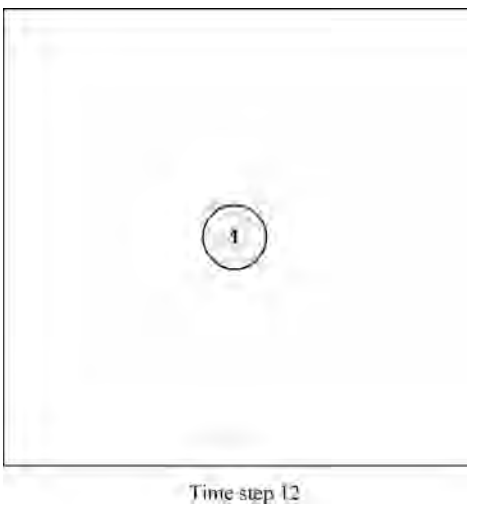


图 5-27 （续）

5.5 多种群多目标人工蜂群算法

5.5.1 算法基本思想与流程

多种群多目标人工蜂群算法(MSMOABC)采用了多个蜂群来协同进化，每个蜂群内部采用原始的蜜蜂邻域搜索策略，向其随机邻居的某一维度学习。群体之间的交流是通过优秀个体替换实现的。每经过若干次迭代(由参数 Tex 控制)，会进行一次替换交流。对每个种群，选出一部分优秀个体和较差个体分别进入发送列表和被替换列表。对各个种群以环形的方式进行替换，用发送列表中优秀个体替换下一种群的被替换列表中的个体，以达到整体进化的目的。整个群落的拓扑结构如图 5-28 所示，每个种群将自己的优良信息传递给下一个种群，相当于一种偏利共生型合作关系。

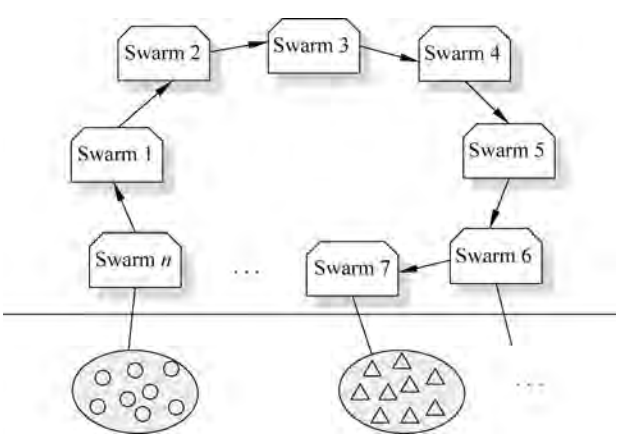


图 5-28 MSMOABC 群体拓扑结构

为用于求解多目标问题，非支配快速排序和基于拥挤距离的选择机制也加入到了算法中。对每个种群中的个体，计算其非支配等级和拥挤距离。在对个体的优劣程度进行排序

时，首先按照非支配等级进行排序，等级值越低表明个体越优秀。对相同等级的个体，再按拥挤距离排序，拥挤距离大的个体优于距离较小的个体。

算法的流程如下：

```
MSMOABC 算法流程
(1) 初始化各个种群,T = 0
(2) While 终止条件不满足
(3) For each 种群
    根据 ABC 算法规则产生新的跟随蜂;
    采用非支配快速排序求取非支配解;
    计算拥挤距离;
    根据拥挤距离选择个体。
(4) End for
(5) IF Mod(T,Tex) = 0
    准备发送列表和被替换列表;
    将发送列表中优秀个体替代下一种群的被替换列表。
(6) End if
```

5.5.2　算法的形式化描述

MSMOABC 算法的进化模型由个体层、群体层和群落层构成。定义如下：

$$\text{MSMOABC}=(\text{Individual},\text{Population},\text{Colony},\text{Environment},\text{Topology},\text{Coevolution})$$

1. 个体层

$$\text{Individual}=\{\text{Individual}_1,\text{Individual}_2,\cdots,\text{Individual}_i,\cdots,\text{Individual}_{\text{INUM}}\},$$

第 i 个个体表示为 $\text{Individual}_i=<\text{IID}_i,\text{POSI}_i,\text{FITNESS}_i,\text{BHV}_i>$，它们分别表示个体 i 的序号、位置、适应度和行为集合。

2. 群体层

$$P=\{P_1,P_2,\cdots,P_M\}$$

群体层的任一个种群可表示为 $\text{Population}=<\text{OPT},\text{PBHV},N>$其中，OPT 为群内最优个体；PBHV 表示群内个体间行为，$\text{PBHV}=\{C^A\}$；N 为种群大小。

3. 群落层

$$\text{Col}=\{\text{Col}_i^1\{\text{Col}_j^2\{\text{Col}_k^3\{\cdots\{\text{Col}_l^{\text{Num}}\}\}\}\}\}$$

4. 环境

Environment 由优化函数定义。

5. 拓扑结构

$$\text{Topology}=(\text{TS},\{A_{ik},P_{jk}^n\})$$

其中，TS 表示此算法静态拓扑结构；A_{ik} 表示第 k 个时间点个体 i 的邻域关系；P_{jk}^n 表示第 k 个时间点第 n 层第 j 个群体的邻域关系。

6. 协同进化

$$\text{Coevolution}=\{\text{Sym}\},\text{Sym}=\{\text{unitA}_i\}=(\{\text{signa}\},\{\text{unitB}\},\{\text{signb}\})_i$$

5.5.3 算法性能分析

1. 测试函数

算法在 ZDT1、ZDT2、ZDT3、ZDT6、DTLZ2 和 DTLZ6 共 6 个基准函数上进行了测试。

1) ZDT1

ZDT1 是一个有 30 个变量的优化问题，具有凸形帕累托前端。函数描述如表达式(5-21)所示。

$$\text{ZDT1}:\begin{cases}\text{Minimize} & f_1(x)=x_1\\ \text{Minimize} & f_2(x)=g(x)\left[1-\sqrt{x_1/g(x)}\right]\\ & g(x)=1+9\left(\sum_{i=2}^{n}x_i\right)\Big/(n-1)\end{cases}\tag{5-21}$$

其中所有变量都在[0,1]区间内。帕累托最优解为：$0\leqslant x_1^*\leqslant 1$ 并且 $x_i^*=0(i=2,3,\cdots,30)$。

2) ZDT2

ZDT2 是一个有 30 个变量的优化问题。函数描述如表达式(5-22)所示。

$$\text{ZDT2}:\begin{cases}\text{Minimize} & f_1(x)=x_1\\ \text{Minimize} & f_2(x)=g(x)\left[1-(x_1/g(x))^2\right]\\ & g(x)=1+9\left(\sum_{i=2}^{n}x_i\right)\Big/(n-1)\end{cases}\tag{5-22}$$

其中所有变量都在[0,1]区间内。帕累托最优解为：$0\leqslant x_1^*\leqslant 1$ 并且 $x_i^*=0(i=2,3,\cdots,30)$。

3) ZDT3

ZDT3 是一个有 30 个变量的优化问题，但其帕累托前端是由一组不连续的曲线构成。函数描述如表达式(5-23)所示。

$$\text{ZDT3}:\begin{cases}\text{Minimize} & f_1(x)=x_1\\ \text{Minimize} & f_2(x)=g(x)\left[1-\sqrt{x_1/g(x)}-\dfrac{x_1}{g(x)}\sin(10\pi x_1)\right]\\ & g(x)=1+9\left(\sum_{i=2}^{n}x_i\right)\Big/(n-1)\end{cases}\tag{5-23}$$

其中所有变量都在[0,1]区间内。帕累托最优解为：$0\leqslant x_1^*\leqslant 1$ 并且 $x_i^*=0(i=2,3,\cdots,30)$，但并非所有满足 $0\leqslant x_1^*\leqslant 1, x_i^*=0$ 的点都在帕累托前端上。

4) ZDT6

ZDT6 是一个具有 10 个变量的多目标优化问题。然而该问题的帕累托最优前端的密度并非均匀分布。函数描述如表达式(5-24)所示。

$$\text{ZDT6}:\begin{cases}\text{Minimize} & f_1(x)=1-\exp(-4x_1)\sin^6(6\pi x_1)\\ \text{Minimize} & f_2(x)=g(x)\left[1-(f_1(x)/g(x))^2\right]\\ & g(x)=1+9\left[\left(\sum_{i=2}^{n}x_i\right)\Big/(n-1)\right]^{0.25}\end{cases}\tag{5-24}$$

其中所有变量都在[0,1]区间内。帕累托最优解为：$0 \leqslant x_1^* \leqslant 1$ 并且 $x_i^* = 0 (i = 2, 3, \cdots, 10)$。

5) DTLZ2

当取 $M = 3$ 时，DTLZ2 测试问题的帕累托最优边界是第一象限内的单位球面，其描述如表达式(5-25)所示。

$$\text{DTLZ2}: \begin{cases} \text{Minimize} & f_1(x) = (1 + g(x_M))\cos(x_1\pi/2)\cdots\cos(x_{M-1}\pi/2) \\ \text{Minimize} & f_2(x) = (1 + g(x_M))\cos(x_1\pi/2)\cdots\sin(x_{M-1}\pi/2) \\ \vdots & \vdots \\ \text{Minimize} & f_M(x) = (1 + g(x_M))\sin(x_1\pi/2) \\ \text{Subject to} & 0 \leqslant x_i \leqslant 1, \quad i = 1, 2, \cdots, n \\ \text{Where} & g(x_M) = \sum_{x_i \in x_M} (x_i - 0.5)^2 \end{cases} \tag{5-25}$$

该问的帕累托最优解为 $x_i^* = 0.5$ ($x_i^* \in x_M$)并且所有目标函数值必须满足 $\sum_{m=1}^{M} (f_m^*)^2 = 1$。和 DTLZ1 一样，这里取 $k = |x_M| = 10$。总的决策变量 $n = M + k - 1$。

6) DTLZ6

DTLZ6 测试问题具有 $2^M - 1$ 个不连续的帕累托最优边界，其描述如式(5-26)所示。

$$\text{DTLZ6}: \begin{cases} \text{Minimize} & f_1(x) = x_1 \\ \text{Minimize} & f_2(x) = x_2 \\ \vdots & \vdots \\ \text{Minimize} & f_{M-1}(x) = x_{M-1} \\ \text{Minimize} & f_M(x) = (1 + g(x_M))h(f_1, f_2, \cdots, f_{M-1}, g) \\ \text{Subject to} & 0 \leqslant x_i \leqslant 1, \quad i = 1, 2, \cdots, n \\ \text{Where} & g(x_M) = 1 + \dfrac{g}{|x_M|} \sum_{x_i \in x_M} x_i \\ & h = M - \sum_{i=1}^{M-1} \left[\dfrac{f_i}{1+g} (1 + \sin(3\pi f_i)) \right] \end{cases} \tag{5-26}$$

该问题能够检测一个 MOEA 使最优个体在不同帕累托最优边界保持较好的分布的能力。这里函数 g 取 $k = |x_M|$，总决策变量数量为 $n = M + k - 1$。建议取 $k = 20$。

2. 评价指标

为了便于定量评估多目标优化算法的性能，这里给出了两种性能量度：

(1) 解集的收敛性量度 Υ。该量度测量的是 MOEA 算法所求出的最优解集向真实帕累托前端逼近的程度，计算式如下：

$$\Upsilon = \frac{\sum_{i=1}^{N} d_i}{N} \tag{5-27}$$

其中，N 是一个 MOEA 算法获得的非支配解的数量；d_i 是每个非支配解到真实帕累托前端的欧氏距离。为了计算该量度，设置 $H=10\,000$ 个帕累托最优解，使这些解的目标函数值均匀分布在真实帕累托前端上。对于每个算法获得的每个解，计算该解到 H 的最近距离。每个算法所有解求得的平均距离被记作收敛性量度 Υ。图 5-29 显示了该量度的计算过程。

(2) 解集的多样性分布量度 Δ。该量度测量的是 MOEA 算法所求出的最优解集的分布程度，即所获得的最优解覆盖真实帕累托前端的程度。计算式如下：

$$\Delta=\frac{d_f+d_l+\sum_{i=1}^{N-1}\mid d_i-\bar{d}\mid}{d_f+d_l+(N-1)\bar{d}} \tag{5-28}$$

其中，d_i 是所获得的非支配解集中连续两个解之间的欧氏距离；N 是一个 MOEA 算法所获得的非支配解集中解的个数；$\bar{d}$ 是所有 d_i 的平均值；d_f 和 d_l 是理论上的极限解与所获得的解集中的边界解之间的距离，描述如图 5-30 所示。

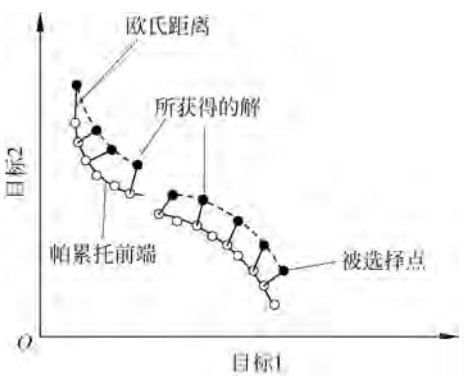


图 5-29　收敛量度 Υ

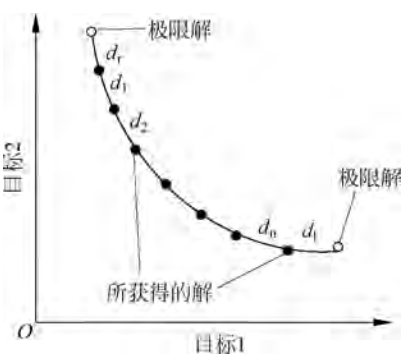


图 5-30　多样性量度 Δ

3. 测试结果

多目标优化结果如表 5-11 所示。

表 5-11　多目标优化结果

测试函数	标准		MSMOABC	NSGA2(pop=100)
ZDT1 $D=30$ 理论帕累托前端是当 $g=1$ 时，参数：	收敛性 越小越好	平均值	8.5264e−004	2.2278e−001
		中值	8.3455e−004	1.4410e−001
		最优值	7.5656e−004	7.2072e−002
		最差值	1.0510e−003	8.7348e−001
		标准差	8.0445e−005	2.3806e−001
	多样性 越小越好	平均值	6.1502e−001	5.9362e−001
		中值	6.0783e−001	5.2694e−001
		最优值	5.3456e−001	4.5810e−001
		最差值	6.7678e−001	9.1451e−001
		标准差	4.4416e−002	1.5649e−001

续表

测试函数	标准		MSMOABC	NSGA2(pop=100)
ZDT2 $D=30$ 理论帕累托前端是当 $g=1$ 时，参数：NSGA5 容易收敛于一点	收敛性 越小越好	平均值	8.2512e−002	2.9476e−001
		中值	8.3727e−004	1.8105e−001
		最优值	6.2466e−004	1.0668e−001
		最差值	4.7223e−001	9.8218e−001
		标准差	1.4789e−001	2.6288e−001
	多样性 越小越好	平均值	6.7809e−001	7.6302e−001
		中值	6.7379e−001	7.2999e−001
		最优值	6.1118e−001	4.6617e−001
		最差值	7.5581e−001	1.0542e+000
		标准差	4.4541e−002	2.3567e−001
ZDT3 $D=30$ 理论帕累托前端是当 $g=1$ 时，参数：	收敛性 越小越好	平均值	4.0779e−003	2.1662e−001
		中值	4.0750e−003	1.5563e−001
		最优值	3.5541e−003	7.9710e−002
		最差值	4.7440e−003	6.3936e−001
		标准差	3.2377e−004	1.7221e−001
	多样性 越小越好	平均值	6.3788e−001	7.0351e−001
		中值	6.4409e−001	6.9255e−001
		最优值	5.5876e−001	6.5156e−001
		最差值	6.7171e−001	8.1501e−001
		标准差	3.3904e−002	4.6110e−002
ZDT6 $D=10$ 理论帕累托前端是当 $g=1$ 时，参数：	收敛性 越小越好	平均值	5.0334e−001	3.3642e+000
		中值	1.0989e−002	3.9688e+000
		最优值	9.6425e−004	6.9186e−001
		最差值	1.7650e+000	4.7179e+000
		标准差	7.4049e−001	1.4822e+000
	多样性 越小越好	平均值	7.9631e−001	1.1219e+000
		中值	7.6843e−001	1.1187e+000
		最优值	6.0809e−001	1.0231e+000
		最差值	1.1885e+000	1.2849e+000
		标准差	1.9418e−001	8.8914e−002
DTLZ2	收敛性 越小越好	平均值	2.8571e−003	8.7363e−002
		中值	2.8582e−003	8.9419e−002
		最优值	2.5627e−003	6.0319e−002
		最差值	3.1086e−003	1.1310e−001
		标准差	1.5407e−004	1.6418e−002
	多样性 越小越好	平均值	4.3011e−001	4.0903e−001
		中值	4.2726e−001	4.0881e−001
		最优值	3.9502e−001	3.7879e−001
		最差值	4.6634e−001	4.3358e−001
		标准差	2.2715e−002	1.8498e−002

续表

测 试 函 数	标　　准		MSMOABC	NSGA2(pop=100)
DTLZ6	收敛性 越小越好	平均值	1.5498e−002	6.6160e−001
		中值	1.5033e−002	2.8640e−001
		最优值	1.2131e−002	1.6724e−001
		最差值	2.0374e−002	3.1497e+000
		标准差	2.6337e−003	9.2366e−001
	多样性 越小越好	平均值	5.2003e−001	5.7305e−001
		中值	5.1893e−001	5.5839e−001
		最优值	4.6557e−001	4.5813e−001
		最差值	5.9150e−001	7.0776e−001
		标准差	3.7954e−002	7.2624e−002

由表 5-11 和图 5-31～图 5-33 可以看出，在收敛性指标上，MSMOABC 在 6 个测试函数上均要优于 NSGA2 算法；在分布性指标上，除在 ZDT1 和 DTLZ2 上略差于 NSGA2 外，其他均比 NSGA2 好。

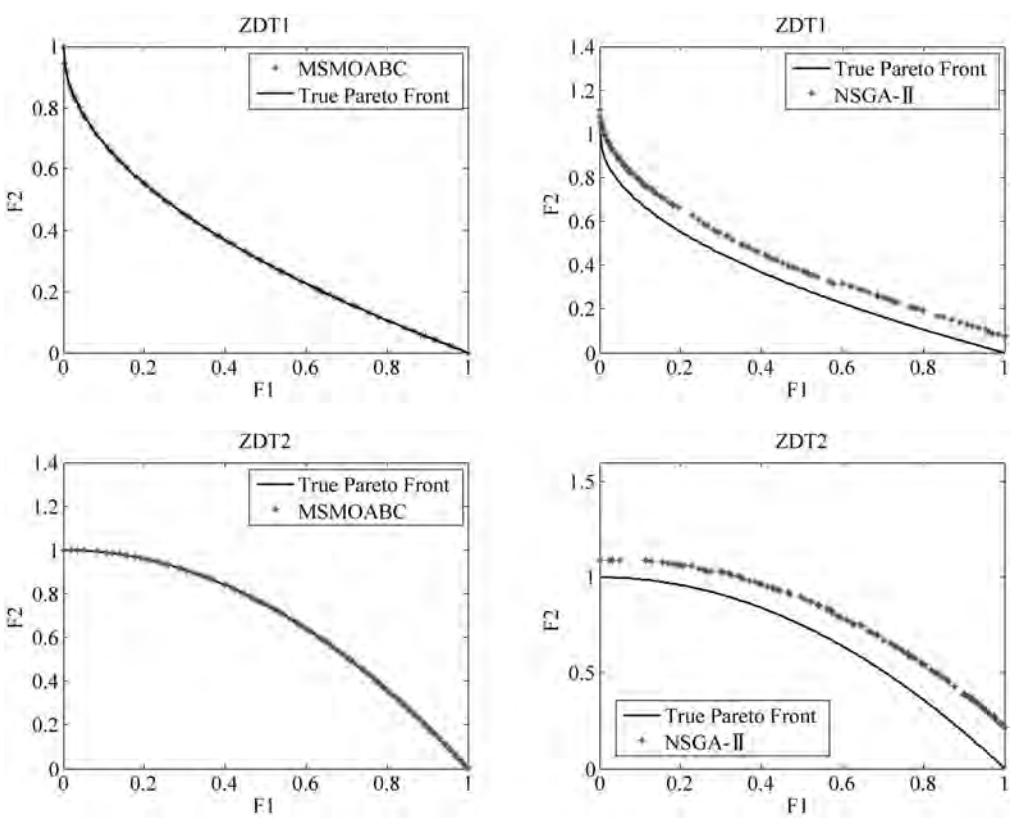


图 5-31　优化结果

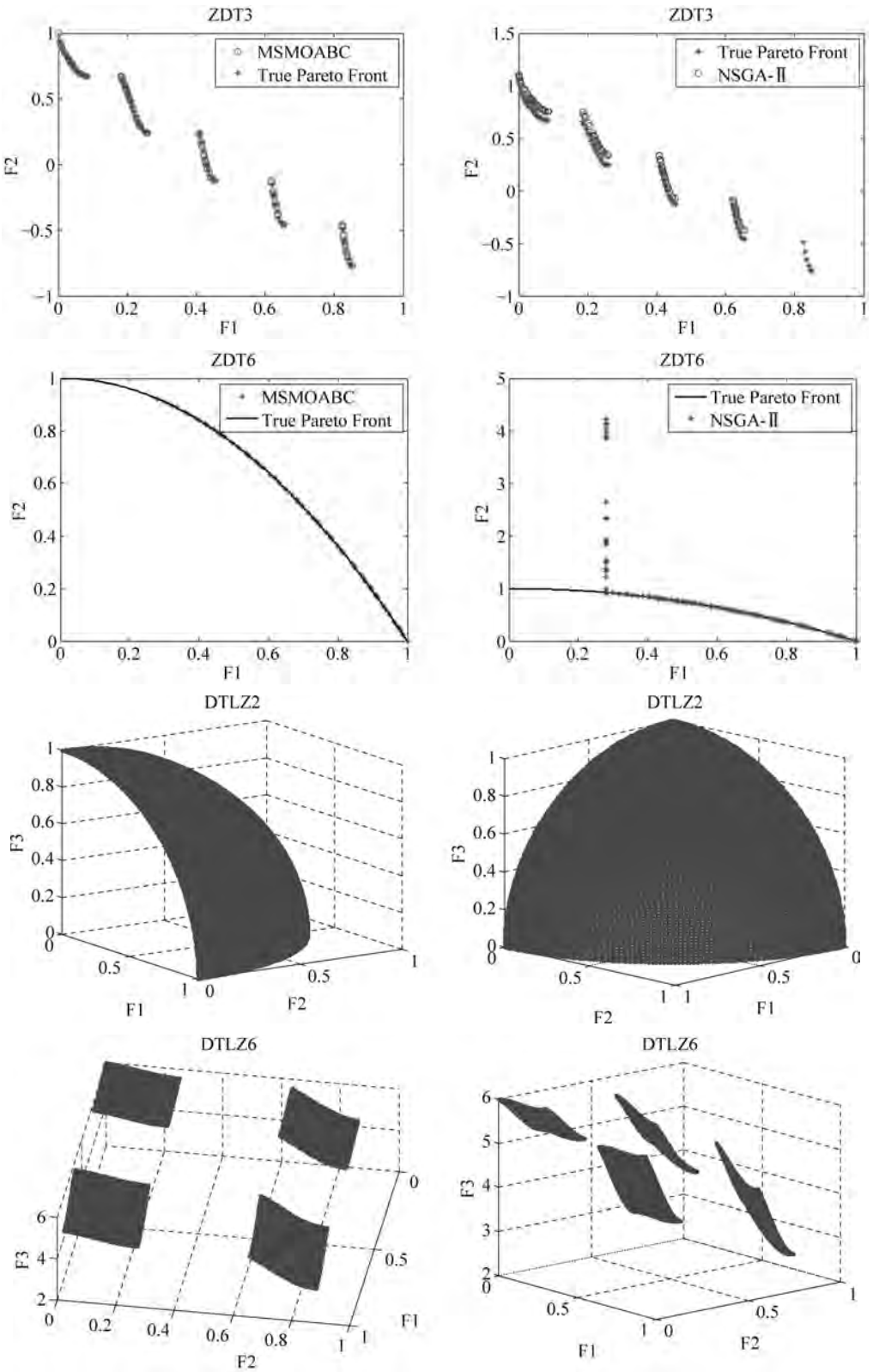

ZDT3
MSMOABC
True Pareto Front
F1
F2
ZDT3
True Pareto Front
NSGA-Ⅱ
F1
F2
ZDT6
MSMOABC
True Pareto Front
F1
F2
ZDT6
True Pareto Front
NSGA-Ⅱ
F1
F2
DTLZ2
F1
F2
F3
DTLZ2
F2
F1
F3
DTLZ6
F2
F1
F3
DTLZ6
F1
F2
F3

图 5-31　（续）

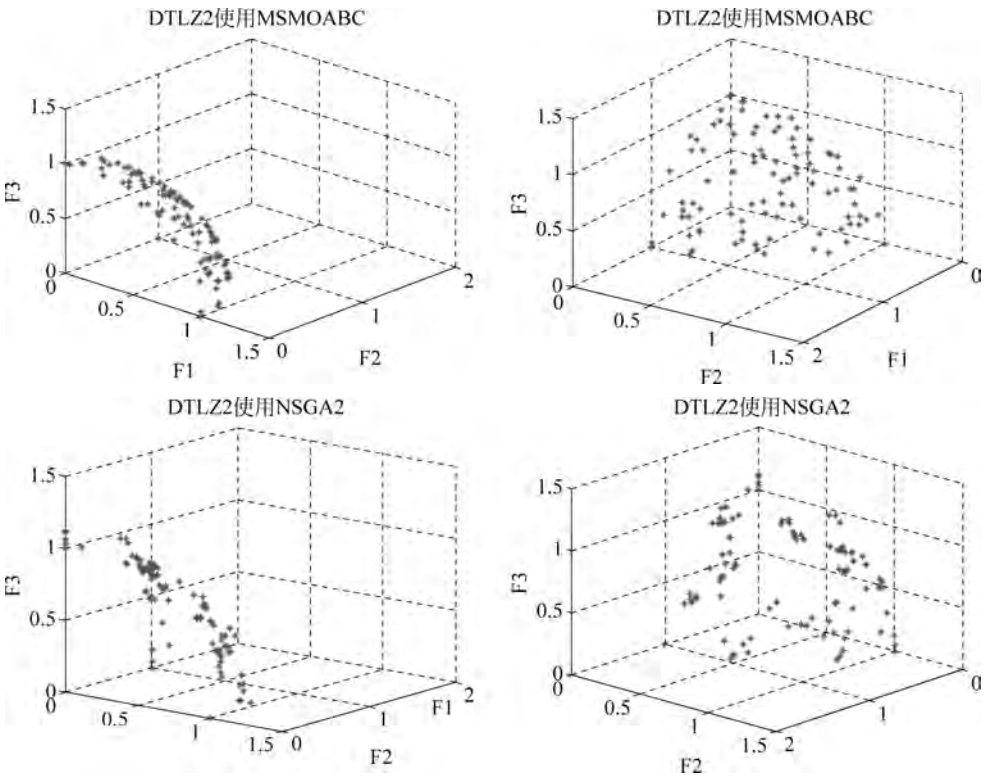


图 5-32　DTLZ2 真实帕累托前端

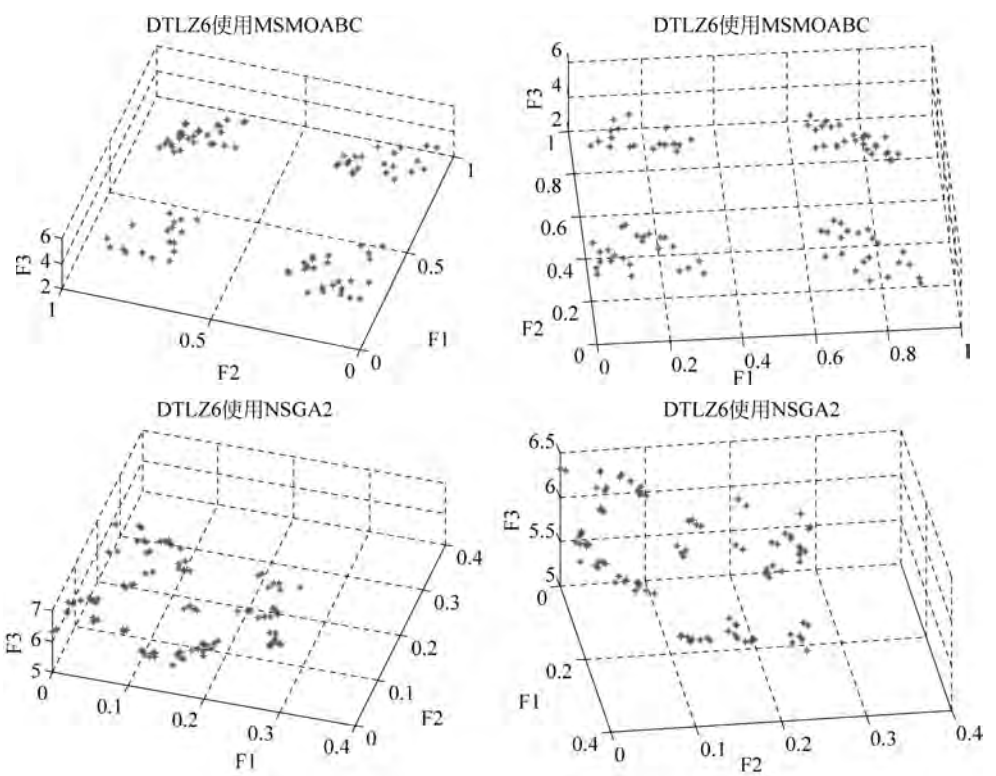


图 5-33　DTLZ6 真实帕累托前端

5.6　基于 p-最优性准则的多种群多目标优化算法

5.6.1　算法基本思想与流程

基于 p-最优性准则的多种群多目标优化算法(p-MSMOEA)是一类应用 p-最优性准则和多种群策略的多目标进化算法的总称,包括基于 p-最优性准则的多种群非支配排序基因算法-Ⅱ(p-MSNSGA-Ⅱ)、基于 p-最优性准则的多种群多目标差分进化算法(p-MSMODE)和基于 p-最优性准则的多种群多目标分解算法(p-MSMOEA/D),以下说明以 p-MSNSGA-Ⅱ算法为例。p-最优性标准用于进化过程中的选择算子,通过一个最优函数 p-function 来确定位于相同非支配等级的解决方案中最可行的解决方案。

$$p\text{-function}(a)=\left(\sum_{i=1}^{k}(1-P_i(a))^p\right)^{\frac{1}{p}}$$

其中,$P_i(a)$是个体 a 在第 i 个目标函数上优于其他个体的概率。

同时,在多群策略中设计了子种群间的竞争与合作机制,采用了多个子种群来协同进化,每个子种群内部采用相应的原始多目标进化算法搜索策略。种群之间的交流是通过优秀个体替换实现的。每经过若干次迭代(由一参数 SMCN 控制),会进行一次替换交流。对每个子种群,选出一部分优秀个体和较差个体分别进入发送列表和被替换列表。对各个子种群以环形的方式进行替换,用发送列表中优秀个体替换下一子种群的被替换列表中的个体,以达到整体进化的目的。整个群落的拓扑结构如图 5-34 所示,每个子种群将自己的优良信息传递给下一个子种群,相当于一种偏利共生型合作关系。最后,经过 SMCN×EN 次迭代,所有子种群融合为一个种群。

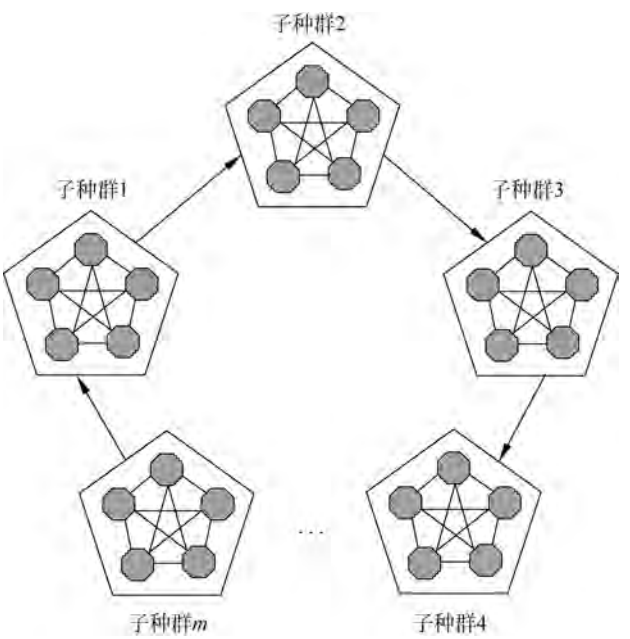


图 5-34　p-MSMOEA 群体拓扑结构

算法流程图如下：

p-MSNSGA-Ⅱ算法流程
1：设置子种群个数(m)，每个子种群大小(N)，p 值上下界(p-lb，p-ub)，最大迭代次数(MCN)，子种群交换迭代数(SMCN)，种群间交换次数(EN)
2：**循环**
3：　初始化子种群 S_1，S_2，…，S_m，为每个子种群分配 p_1，p_2，…，p_m
4：　　**循环**
5：　　初始化的个体按照非支配方法排序
6：　　对于每个子种群 S_j^t
　　应用基于 p-最优性准则的选择算子来选择优良个体
　　应用交叉和变异算子
　　生成后代子种群 $S_j^{t'}$
　　对 S_j^t 和 $S_j^{t'}$ 应用非支配排序和拥挤距离生成下一父代子种群 S_j^{t+1}
　　如果 mod(I，SMCN) = 0
　　m 个子种群进行种群交换
　　结束条件
7：　　结束循环
8：　　**迭代次数为** SMCN * EN **结束循环**
9：　子种群融合为一个种群并且应用非支配排序
10：**迭代次数为** MCN **结束循环**

5.6.2　算法的形式化描述

p-MSNSGA-Ⅱ算法的进化模型由个体层、群体层和群落层构成。定义如下：

p-MSNSGAII = (Individual，Population，Colony，Environment，Topology，Coevolution)

1. 个体层

$$\text{Individual} = \{\text{Individual}_1, \text{Individual}_2, \cdots, \text{Individual}_i, \cdots, \text{Individual}_{\text{INUM}}\}$$

第 i 个个体表示为 $\text{Individual}_i = < \text{IID}_i, \text{POSI}_i, \text{FITNESS}_i, \text{BHV}_i >$，它们分别表示个体 i 的序号、位置、适应度和行为集合。

2. 群体层

$$P = \{P_1, P_2, \cdots, P_M\}$$

群体层的任一个种群可表示为 $\text{Population} = < \text{OPT}, \text{PBHV}, N >$其中，OPT 为群内最优个体；PBHV 表示群内个体间行为，$\text{PBHV} = \{C^A\}$；N 为种群大小。

3. 群落层

$$\text{Col} = \{\text{Col}_i^1 \{\text{Col}_j^2 \{\text{Col}_k^3 \{\cdots \{\text{Col}_l^{\text{Num}}\}\}\}\}\}$$

4. 环境

Environment 由优化函数定义。

5. 拓扑结构

$$\text{Topology} = (\text{TS}, \{A_{ik}, P_{jk}^n\})$$

其中，TS 表示此算法静态拓扑结构；A_{ik}：表示第 k 个时间点个体 i 的邻域关系；P_{jk}^n 表示

第 k 个时间点第 n 层第 j 个群体的邻域关系。

6. 协同进化

$$\text{Coevolution}=\{\text{Sym}\},\quad \text{Sym}=\{\text{unitA}_i\}=(\{\text{signa}\},\{\text{unitB}\},\{\text{signb}\})_i$$

5.6.3　算法性能分析

1. 测试函数

算法在 SCH2、ZDT3、ZDT4、ZDT6、DTLZ2 和 DTLZ3 共 6 个基准函数上进行了测试。

1) SCH2

SCH2 是一个有 1 个变量的优化问题。函数描述如表达式(5-29)所示。

$$\text{SCH2}:\begin{cases}\text{Minimize } f_1(x)=\begin{cases}-x, & x\leqslant 1\\ x-2, & 1<x\leqslant 3\\ 4-x, & 3<x\leqslant 4\\ x-4, & x>4\end{cases}\\ \text{Minimize } f_2(x)=(x-5)^2\end{cases}\tag{5-29}$$

其中所有变量都在[−5,10]区间内。

2) ZDT3

ZDT3 是一个有 30 个变量的优化问题,但其帕累托前端是由一组不连续的曲线构成的。函数描述如表达式(5-30)所示。

$$\text{ZDT3}:\begin{cases}\text{Minimize} \quad f_1(x)=x_1\\ \text{Minimize} \quad f_2(x)=g(x)\left[1-\sqrt{x_1/g(x)}-\dfrac{x_1}{g(x)}\sin(10\pi x_1)\right]\\ g(x)=1+9\left(\sum\limits_{i=2}^{n}x_i\right)\Big/(n-1)\end{cases}\tag{5-30}$$

其中所有变量都在[0,1]区间内。帕累托最优解为:$0\leqslant x_1^*\leqslant 1$ 并且 $x_i^*=0(i=2,3,\cdots,30)$,但并非所有满足 $0\leqslant x_1^*\leqslant 1, x_i^*=0$ 的点都在帕累托前端上。

3) ZDT4

ZDT4 是一个有 10 个变量的优化问题。函数描述如表达式(5-31)所示。

$$\text{ZDT4}:\begin{cases}\text{Minimize} \quad f_1(x)=x_1\\ \text{Minimize} \quad f_2(x)=g(x)\left[1-\sqrt{\dfrac{f_1(x)}{g(x)}}\right]\\ g(x)=91+\sum\limits_{i=2}^{10}(x_i^2-10\cos(4\pi x_i))\end{cases}\tag{5-31}$$

其中第一个变量在[0,1]区间内,剩余变量在[−5,5]区间内。

4) ZDT6

ZDT6 是一个具有 10 个变量的多目标优化问题。然而该问题的帕累托最优前端的密度并非均匀分布。函数描述如表达式(5-32)所示。

$$\text{ZDT6}:\begin{cases}\text{Minimize} \quad f_1(x)=1-\exp(-4x_1)\sin^6(6\pi x_1)\\ \text{Minimize} \quad f_2(x)=g(x)[1-(f_1(x)/g(x))^2]\\ g(x)=1+9\left[\left(\sum\limits_{i=2}^{n}x_i\right)\Big/(n-1)\right]^{0.25}\end{cases}\tag{5-32}$$

其中所有变量都在[0,1]区间内。帕累托最优解为：$0 \leqslant x_1^* \leqslant 1$ 并且 $x_i^* = 0 (i = 2,3,\cdots,10)$。

5) DTLZ2

当取 $M = 3$ 时，DTLZ2 测试问题的帕累托最优边界是第一象限内的单位球面，其描述如表达式(5-33)所示。

$$\text{DTLZ2}:\begin{cases}\text{Minimize} & f_1(x) = (1+g(x_M))\cos(x_1\pi/2)\cdots\cos(x_{M-1}\pi/2)\\ \text{Minimize} & f_2(x) = (1+g(x_M))\cos(x_1\pi/2)\cdots\sin(x_{M-1}\pi/2)\\ \vdots & \vdots\\ \text{Minimize} & f_M(x) = (1+g(x_M))\sin(x_1\pi/2)\\ \text{Subject to} & 0 \leqslant x_i \leqslant 1, \quad i = 1,2,\cdots,n\\ \text{Where} & g(x_M) = \sum_{x_i \in x_M} (x_i - 0.5)^2\end{cases} \tag{5-33}$$

该问题的帕累托最优解为 $x_i^* = 0.5$ $(x_i^* \in x_M)$ 并且所有目标函数值必须满足 $\sum_{m=1}^{M} (f_m^*)^2 = 1$。和 DTLZ1 一样，这里取 $k = |x_M| = 10$。总的决策变量 $n = M + k - 1$。

6) DTLZ3

DTLZ3 测试问题具有球形的帕累托最优边界，其描述如表达式(5-34)所示。

$$\text{DTLZ3}:\begin{cases}\text{Minimize} & f_1(x) = (1+g(x_M))\cos(x_1\pi/2)\cdots\cos(x_{M-1}\pi/2)\\ \text{Minimize} & f_2(x) = (1+g(x_M))\cos(x_1\pi/2)\cdots\sin(x_{M-1}\pi/2)\\ \vdots & \vdots\\ \text{Minimize} & f_M(x) = (1+g(x_M))\sin(x_1\pi/2)\\ \text{Subject to} & 0 \leqslant x_i \leqslant 1, i = 1,2,\cdots,n\\ \text{where} & g(x_M) = 100\left[|x_M| + \sum_{x_i \in x_M} (x_i - 0.5)^2 - \cos(20\pi(x_i - 0.5))\right]\end{cases} \tag{5-34}$$

其中 $k = |x_M|$，总决策变量数量为 $n = M + k - 1$。建议取 $k = 10$。

2. 评价指标

为了便于定量评估多目标优化算法的性能，这里给出了两种性能量度：

(1) 解集的收敛性量度 Υ_j。该量度测量的是 MOEA 算法所求出的最优解集向真实帕累托前端逼近的程度，计算式如式(5-35)所示：

$$\Upsilon = \frac{\sum_{i=1}^{N} d_i}{N} \tag{5-35}$$

其中，N 是一个 MOEA 算法获得的非支配解的数量；d_i 是每个非支配解到真实帕累托前端的欧式距离。为了计算该量度，设置 $H = 10\,000$ 个帕累托最优解，使这些解的目标函数值均匀分布在真实帕累托前端上。对于每个算法获得的每个解，计算该解到 H 的最近距离。每个算法所有解求得的平均距离被记作收敛性量度 Υ。图 5-35 所示显示了该量度的计算过程。

（2）解集的多样性分布量度 Δ。该量度测量的是 MOEA 算法所求出的最优解集的分布程度，即所获得的最优解覆盖真实帕累托前端的程度。计算式如下：

$$\Delta=\frac{d_f+d_l+\sum_{i=1}^{N-1}\left|d_i-\bar{d}\right|}{d_f+d_l+(N-1)\bar{d}} \tag{5-34}$$

其中，d_i 是所获得的非支配解集中连续两个解之间的欧氏距离；N 是一个 MOEA 算法所获得的非支配解集中解的个数。$\bar{d}$ 是所有 d_i 的平均值。d_f 和 d_l 是理论上的极限解与所获得的解集中的边界解之间的距离，描述如图 5-36 所示。

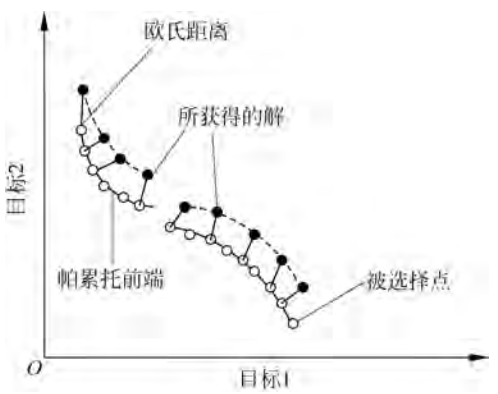


图 5-35　收敛量度 Υ

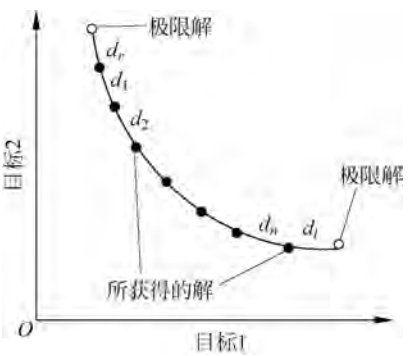


图 5-36　多样性量度 Δ

3. 测试结果

多目标测试结果如表 5-12 所示。

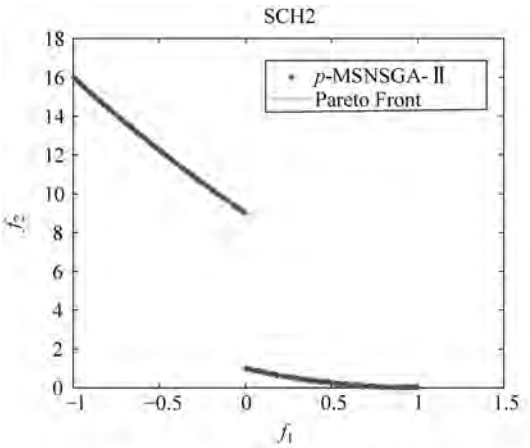
表 5-12　多目标优化结果

测试函数	标准		p-MSNSGA-Ⅱ	p-MSMODE	p-MSMOEA/D	NSGA-Ⅱ	MODE	MOEA/D
SCH2 $D=1$	收敛性越小越好	平均值	5.01e−06	8.73e−05	7.59e−04	4.01e−04	3.54e−03	8.19e−03
		最优值	3.93e−06	1.20e−06	7.05e−04	3.78e−04	3.35e−03	7.29e−03
		最差值	2.04e−05	1.72e−04	8.03e−04	7.80e−04	3.83e−03	9.96e−03
		标准差	2.10e−06	1.61e−05	3.99e−05	1.10e−05	1.98e−04	5.29e−04
	多样性越小越好	平均值	6.81e−02	5.19e−02	7.73e−02	6.48e−02	7.21e−02	8.59e−02
		最优值	3.06e−02	4.93e−02	7.01e−02	4.09e−02	6.62e−02	7.25e−02
		最差值	8.62e−02	6.48e−02	1.13e−01	1.16e−01	1.19e−01	1.31e−01
		标准差	5.59e−03	4.95e−03	2.57e−03	1.27e−03	5.50e−03	7.60e−03
ZDT3 $D=30$	收敛性越小越好	平均值	8.59e−05	2.51e−05	4.15e−04	2.03e−02	1.76e−02	8.62e−03
		最优值	5.37e−05	2.04e−05	3.96e−04	1.09e−02	5.51e−03	7.40e−03
		最差值	1.28e−04	2.86e−04	4.34e−04	2.26e−02	2.05e−02	3.05e−02
		标准差	3.00e−05	4.92e−04	1.59e−05	5.71e−02	7.85e−03	6.30e−02
	多样性越小越好	平均值	4.62e−02	6.24e−02	7.69e−02	5.21e−02	7.73e−02	8.45e−02
		最优值	3.79e−02	5.65e−02	5.65e−02	4.63e−02	5.85e−02	6.55e−02
		最差值	6.40e−02	6.52e−02	1.28e−01	6.59e−02	9.82e−02	1.32e−01
		标准差	2.49e−03	2.31e−03	1.42e−02	3.85e−03	7.42e−03	1.80e−02

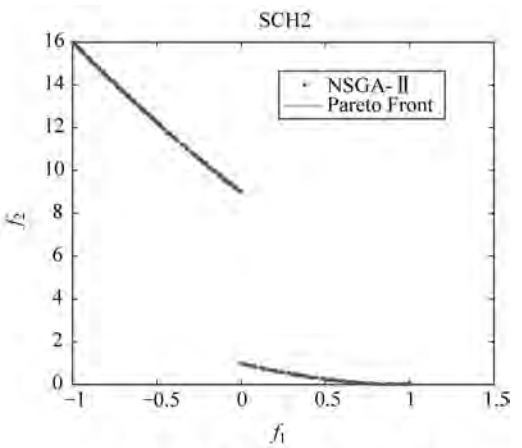
续表

测试函数	标准		p-MSNSGA-Ⅱ	p-MSMODE	p-MSMOEA/D	NSGA-Ⅱ	MODE	MOEA/D
ZDT4 $D=30$	收敛性越小越好	平均值	1.04e−02	2.61e−03	4.77e−03	3.69e−02	1.64e−02	4.78e−02
		最优值	1.01e−02	1.12e−03	1.62e−03	2.40e−02	1.97e−03	3.01e−02
		最差值	1.14e−02	7.51e−03	9.41e−03	1.21e−01	7.68e−02	9.93e−02
		标准差	5.78e−04	1.14e−02	1.56e−02	1.44e−02	2.05e−02	2.41e−02
	多样性越小越好	平均值	6.63e−02	7.02e−02	9.81e−02	8.93e−02	9.08e−02	1.00e−01
		最优值	6.13e−02	4.90e−02	7.05e−02	6.32e−02	9.05e−02	8.24e−02
		最差值	7.91e−02	9.87e−02	1.19e−01	1.09e−01	1.68e−01	1.18e−01
		标准差	7.36e−03	8.87e−03	6.78e−03	1.12e−02	1.64e−01	6.89e−03
ZDT6 $D=10$	收敛性越小越好	平均值	2.95e−05	2.89e−04	4.30e−04	6.98e−02	3.71e−02	5.48e−02
		最优值	2.39e−05	2.71e−05	3.21e−04	5.31e−02	2.72e−03	5.09e−02
		最差值	3.87e−05	6.45e−04	6.74e−04	7.72e−02	7.37e−02	7.01e−02
		标准差	6.08e−06	1.10e−05	7.63e−05	6.52e−03	2.17e−02	6.45e−03
	多样性越小越好	平均值	6.37e−02	5.68e−02	8.14e−02	8.35e−01	9.99e−01	8.96e−01
		最优值	5.87e−02	4.08e−02	6.76e−02	6.12e−01	9.31e−01	6.19e−01
		最差值	7.51e−02	7.32e−02	1.01e−01	1.12e+00	1.24e+00	1.16e+00
		标准差	1.26e−03	4.03e−03	2.06e−03	1.08e−01	1.91e−02	1.87e−01
DTLZ2	收敛性越小越好	平均值	2.96e−03	1.96e−04	4.06e−04	4.15e−02	1.79e−03	5.51e−03
		最优值	2.88e−03	2.30e−05	6.80e−05	3.05e−02	7.50e−04	1.58e−03
		最差值	3.02e−03	7.13e−04	5.67e−03	6.51e−02	7.97e−02	6.27e−02
		标准差	6.30e−04	8.48e−05	5.87e−04	8.10e−03	2.11e−02	1.01e−02
	多样性越小越好	平均值	4.46e−02	4.21e−02	4.76e−02	5.07e−01	4.77e−01	5.39e−01
		最优值	3.37e−02	3.75e−02	2.16e−02	3.85e−01	4.23e−01	2.88e−01
		最差值	4.60e−02	4.64e−02	7.61e−02	6.73e−01	4.92e−01	9.98e−01
		标准差	1.97e−03	1.88e−03	7.12e−03	5.32e−02	7.81e−02	1.29e−01
DTLZ3	收敛性越小越好	平均值	3.92e−02	4.70e−02	1.84e−01	1.21e−01	3.72e−01	3.60e−01
		最优值	3.48e−02	1.86e−03	6.73e−02	5.08e−02	1.53e−01	1.78e−01
		最差值	4.10e−02	1.05e−01	8.26e−01	6.56e−01	1.08e+00	1.02e+00
		标准差	2.55e−01	1.23e−01	1.24+00	8.65e−01	2.38e+00	2.6+00
	多样性越小越好	平均值	5.87e−02	3.75e−02	8.00e−02	9.42e−02	4.16e−02	8.02e−02
		最优值	4.21e−02	3.49e−02	2.70e−02	4.54e−02	3.61e−02	3.45e−02
		最差值	6.97e−02	4.67e−02	1.21e−01	1.24e−01	4.81e−02	1.27e−01
		标准差	7.35e−03	2.62e−03	1.52e−02	2.07e−02	2.81e−03	2.06e−02

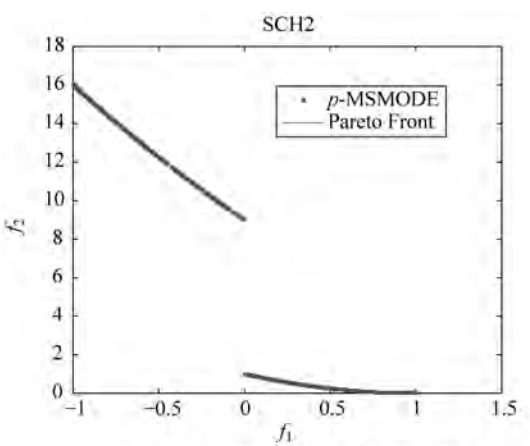
可以看出，在收敛性和分布性指标上，p-MSNSGA-Ⅱ、p-MSMODE 和 p-MSMOEA/D 在 6 个测试函数上均分别优于 NSGA-Ⅱ、MODE 和 MOEA/D 算法，如图 5-37～图 5-43 所示。



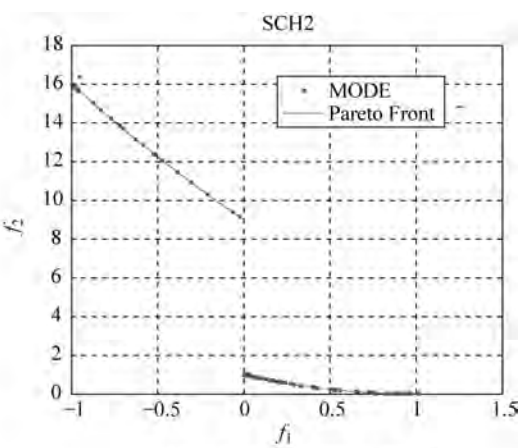
(a) *p*-MSNSGA-Ⅱ在SCH2函数上的PF值



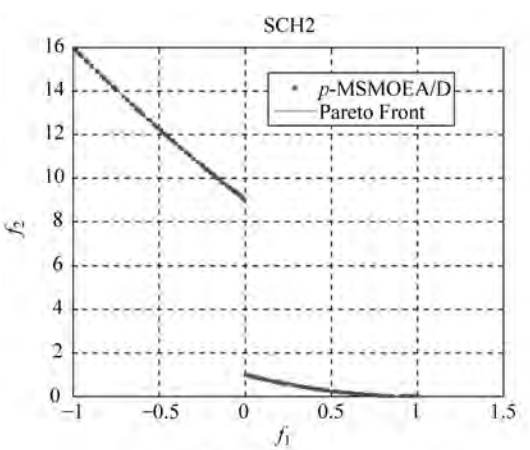
(b) NSGA-Ⅱ在SCH2函数上的PF值



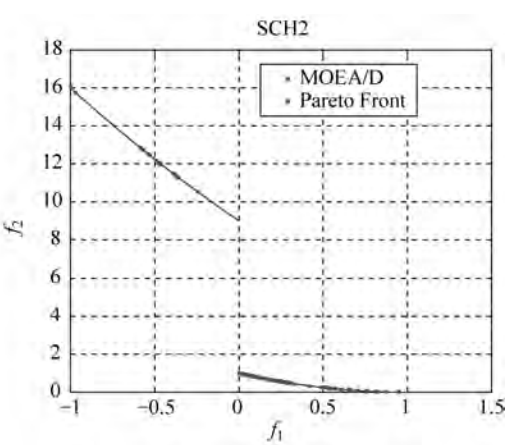
(c) *p*-MSMODE在SCH2函数上的PF值



(d) MODE在SCH2函数上的PF值

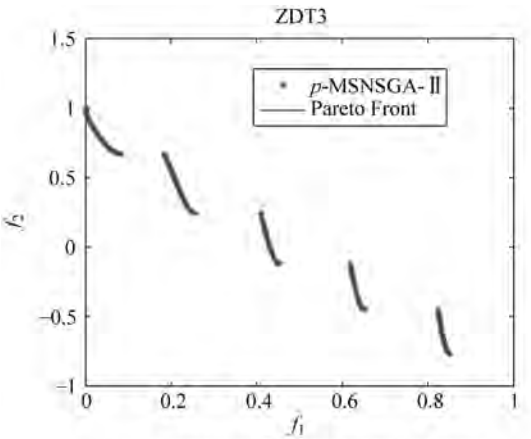


(e) *p*-MSMOEA/D在SCH2函数上的PF值

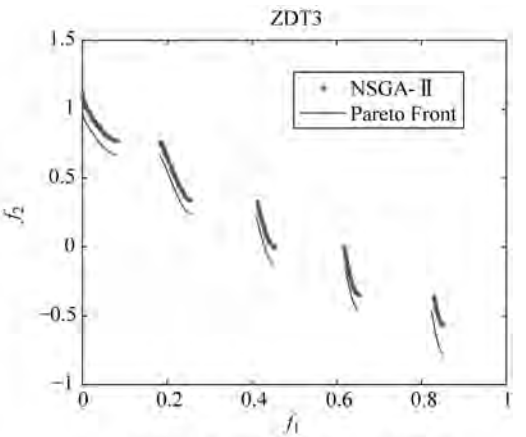


(f) MOEA/D在SCH2函数上的PF值

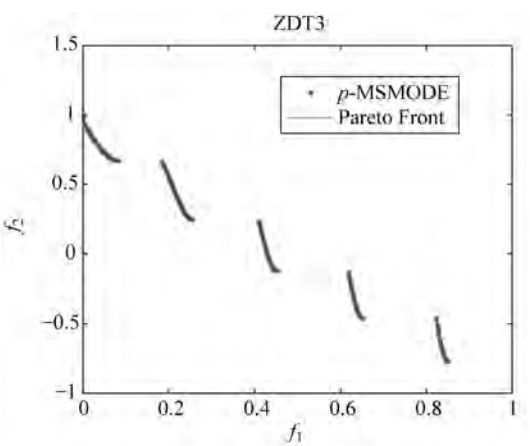
图 5-37　SCH2 优化结果



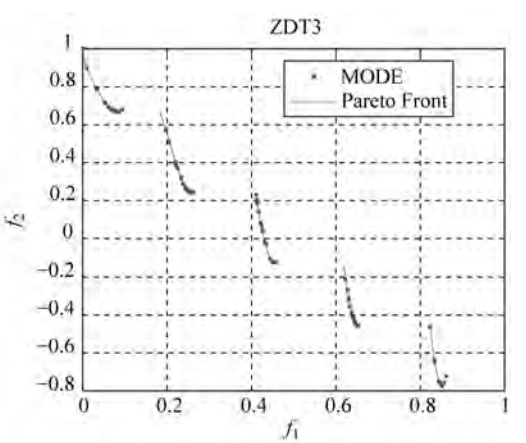
(a) *p*-MSNSGA-Ⅱ在ZDT3函数上的PF值



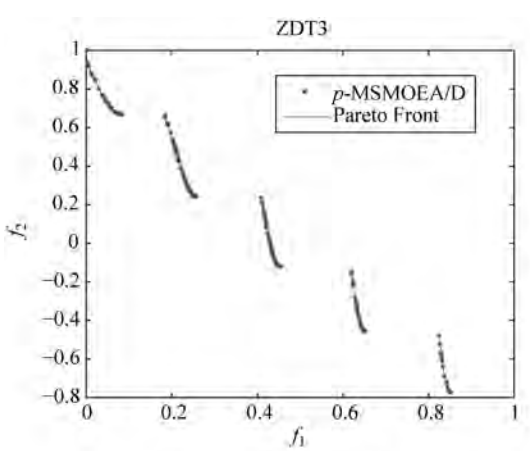
(b) NSGA-Ⅱ在ZDT3函数上的PF值



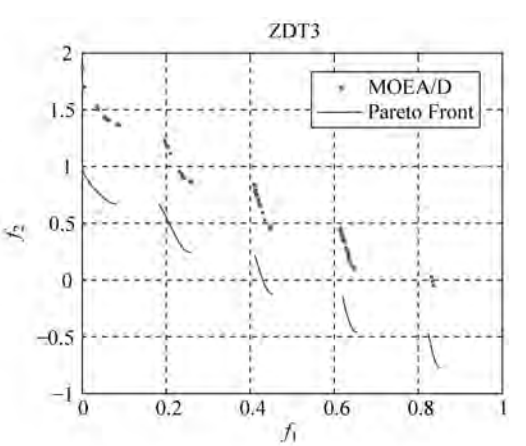
(c) *p*-MSMODE在ZDT3函数上的PF值



(d) MODE在ZDT3函数上的PF值

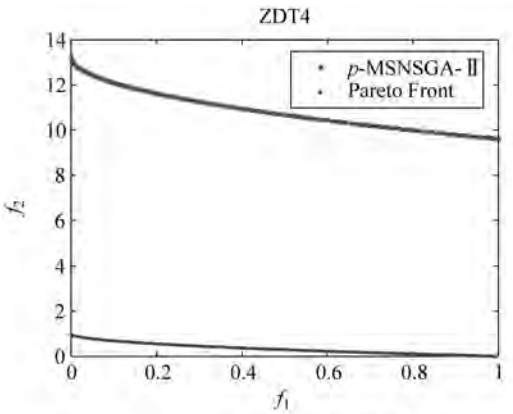


(e) *p*-MSMOEA/D在ZDT3函数上的PF值

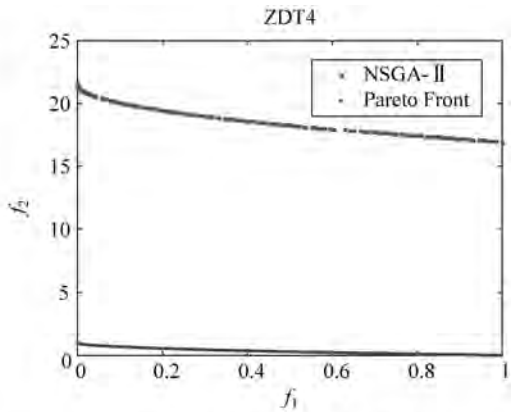


(f) MOEA/D在ZDT3函数上的PF值

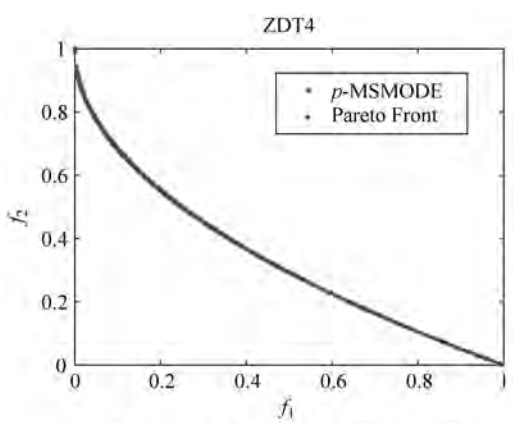
图 5-38　ZDT3 优化结果



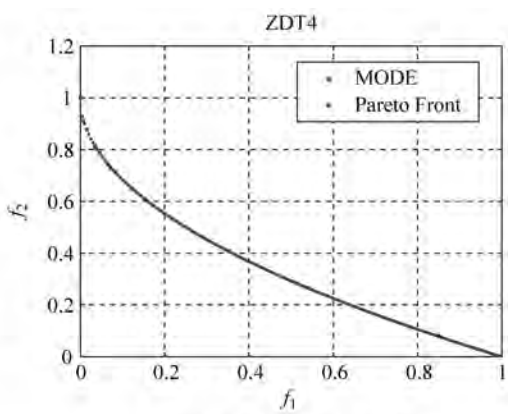
(a) p-MSNSGA-Ⅱ在ZDT4函数上的PF值



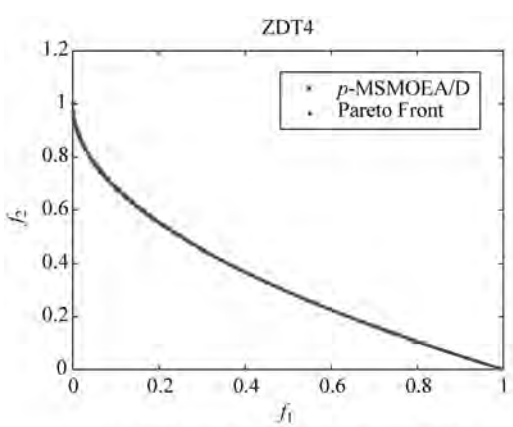
(b) NSGA-Ⅱ在ZDT4函数上的PF值



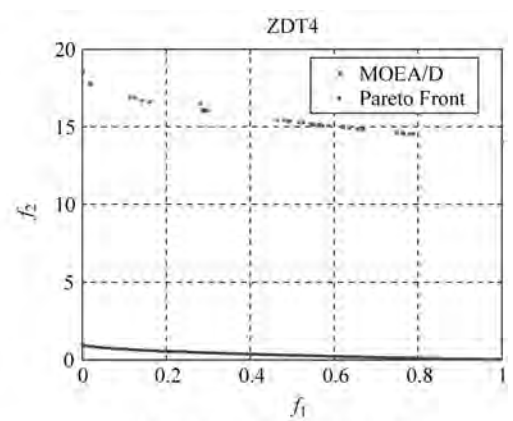
(c) p-MSMODE在ZDT4函数上的PF值



(d) MODE在ZDT4函数上的PF值

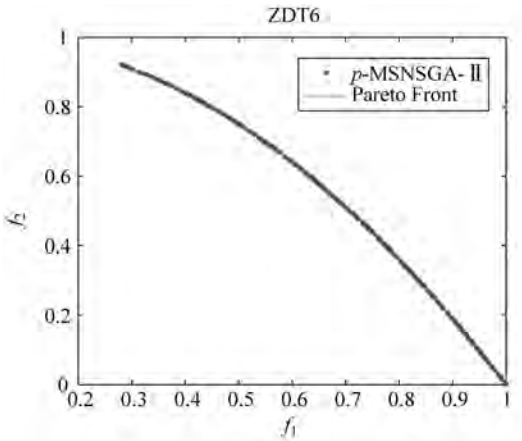


(e) p-MSMOEA/D在ZDT4函数上的PF值

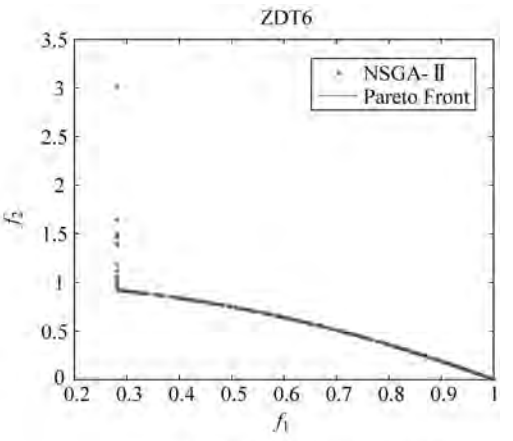


(f) MOEA/D在ZDT4函数上的PF值

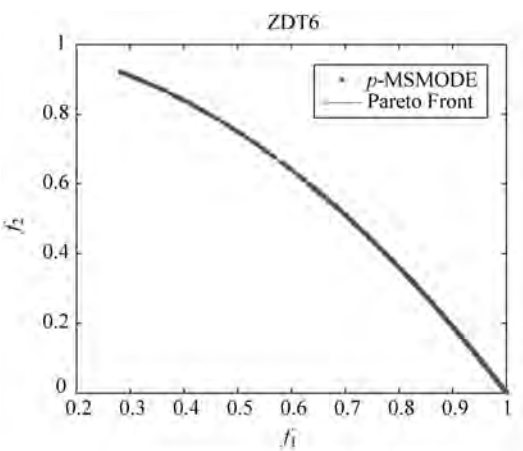
图 5-39　ZDT4 优化结果



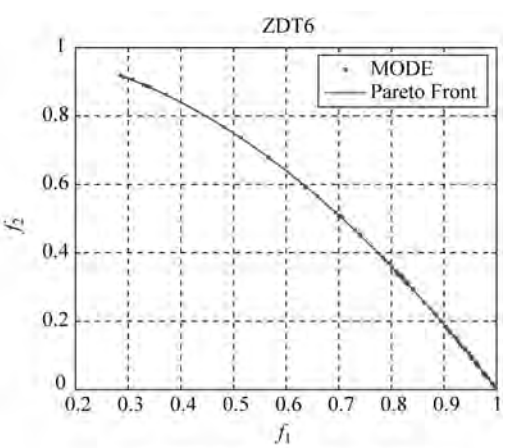
(a) *p*-MSNSGA-Ⅱ在ZDT6函数上的PF值



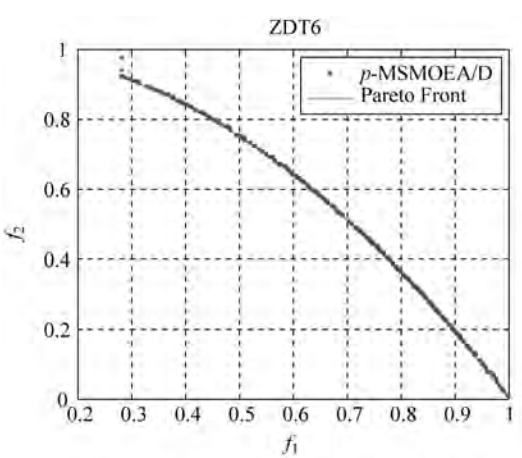
(b) NSGA-Ⅱ在ZDT6函数上的PF值



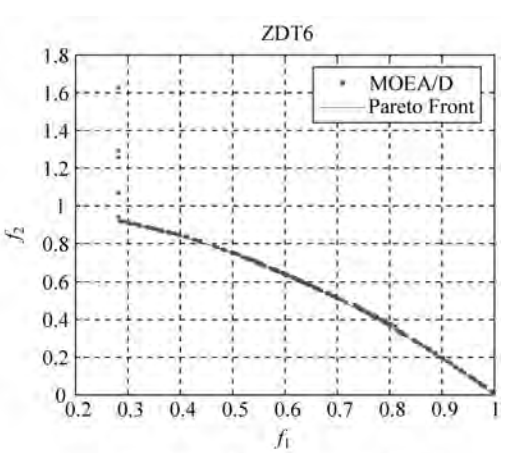
(c) *p*-MSMODE在ZDT6函数上的PF值



(d) MODE在ZDT6函数上的PF值

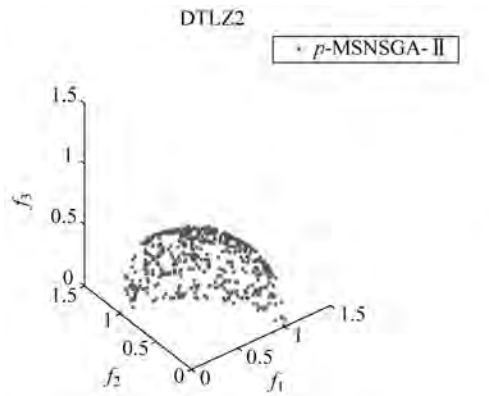


(e) *p*-MSMOEA/D在ZDT6函数上的PF值

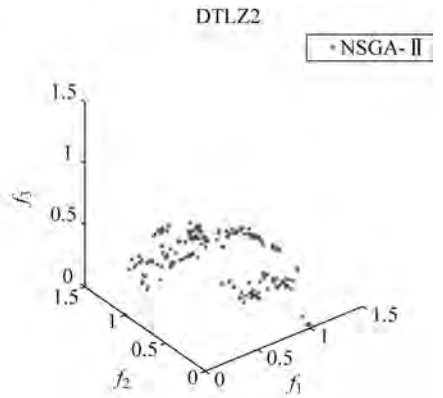


(f) MOEA/D在ZDT6函数上的PF值

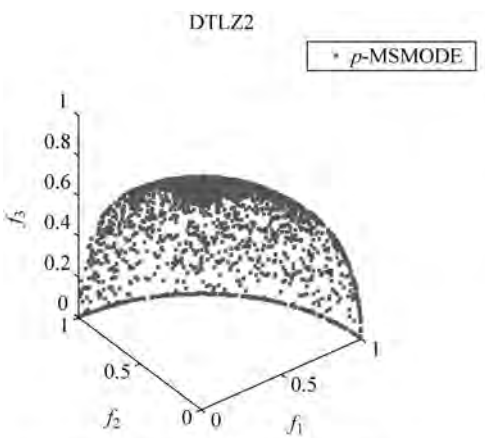
图 5-40　ZDT6 优化结果



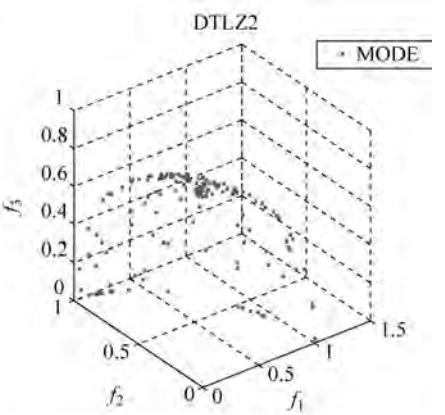
(a) *p*-MSNSGA-Ⅱ在DTLZ2函数上的PF值



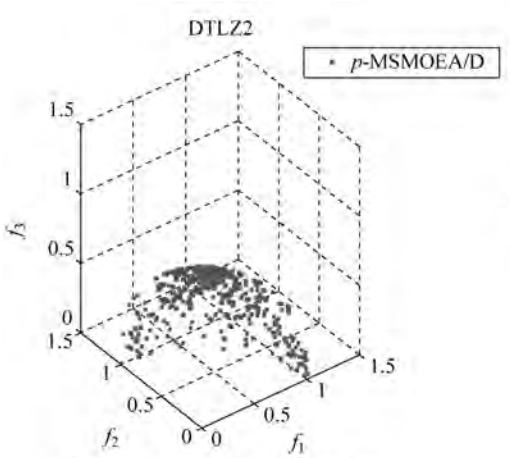
(b) NSGA-Ⅱ在DTLZ2函数上的PF值



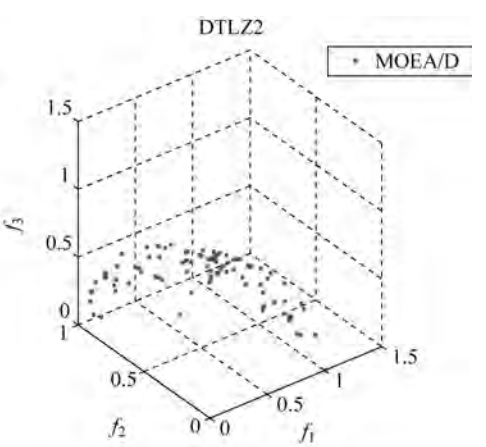
(c) *p*-MSMODE在DTLZ2函数上的PF值



(d) MODE在DTLZ2函数上的PF值

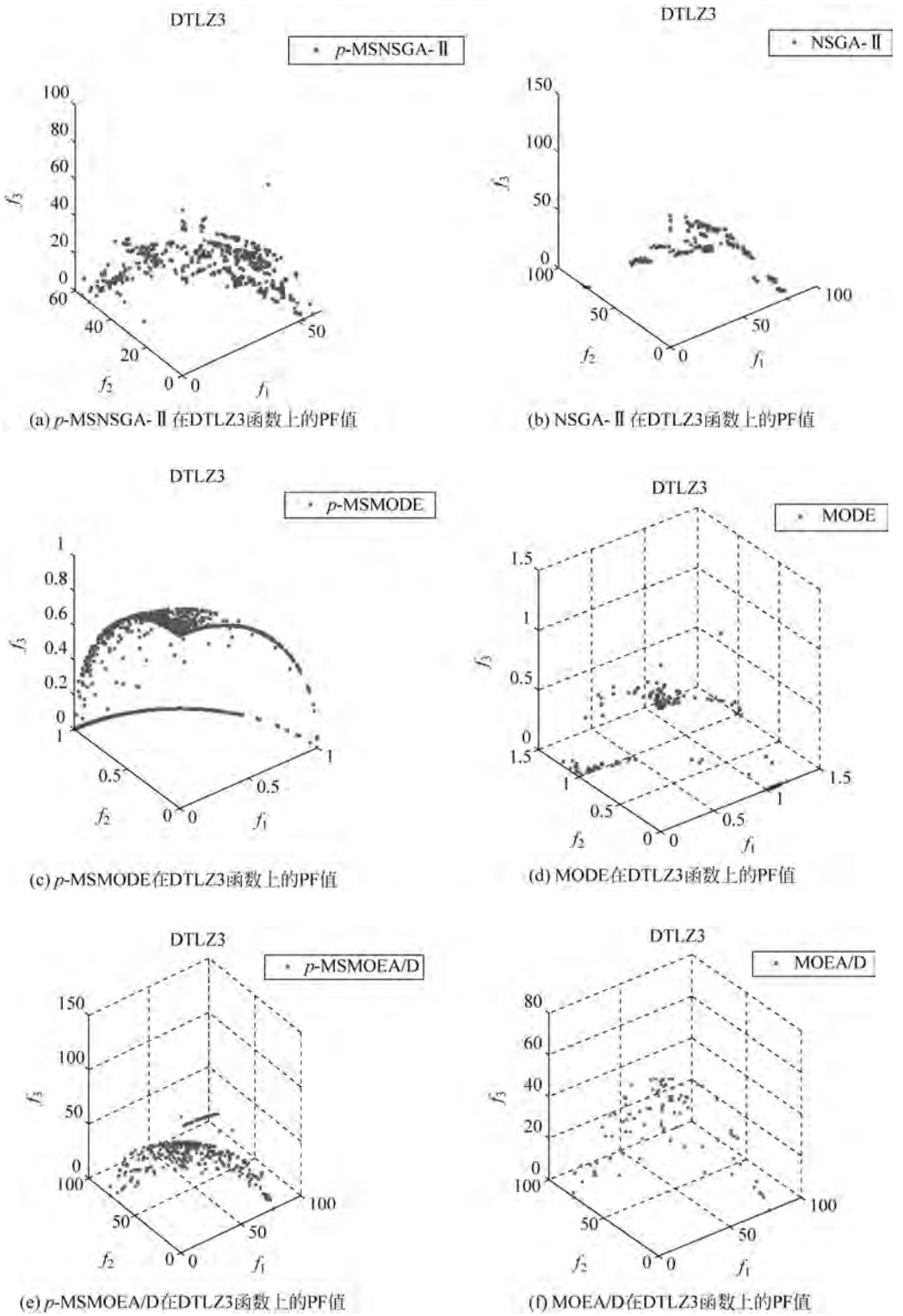


(e) *p*-MSMOEA/D在DTLZ2函数上的PF值



(f) MOEA/D在DTLZ2函数上的PF值

图 5-41　DTLZ2 优化结果



(a) *p*-MSNSGA-Ⅱ在DTLZ3函数上的PF值

(b) NSGA-Ⅱ在DTLZ3函数上的PF值

(c) *p*-MSMODE在DTLZ3函数上的PF值

(d) MODE在DTLZ3函数上的PF值

(e) *p*-MSMOEA/D在DTLZ3函数上的PF值

(f) MOEA/D在DTLZ3函数上的PF值

图 5-42　DTLZ3 优化结果

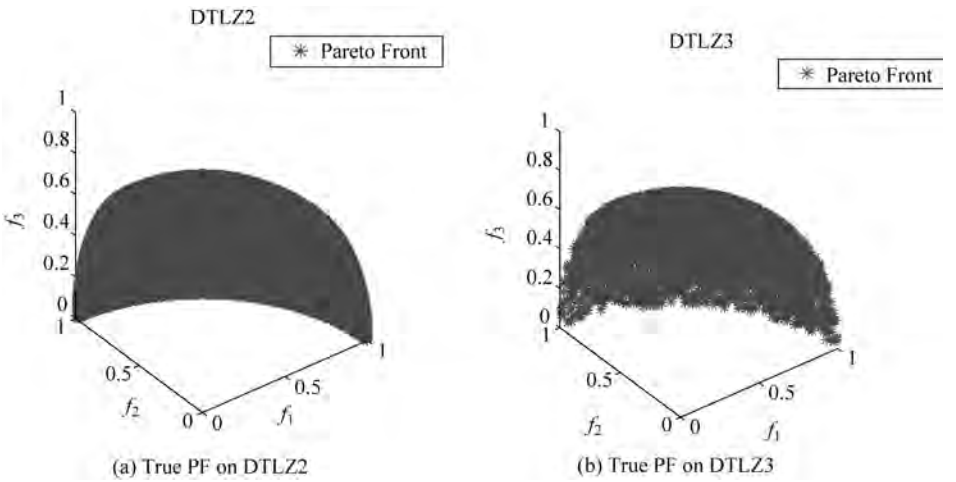


图 5-43　DTLZ2、DTLZ3 真实帕累托前端

参 考 文 献

[1] 李振基. 生态学[M]. 北京：科学出版社，2000.

[2] Janzen D H. When is it coevolution? [J]. Evolution，1980，34：611-612.

[3] 尚玉昌. 行为生态学[M]. 北京：北京大学出版社，1998.

[4] Douglas A E. Symbiotic Interactions[M]. Oxford：Oxford University Press，1994.

[5] 陈瀚宁. RFID 系统优化模型与智能算法研究[D]. 沈阳：中国科学院沈阳自动化研究所，2009.

[6] 牛奔. 基于生物行为的群体智能优化方法研究[D]. 沈阳：中国科学院沈阳自动化研究所，2008.

[7] Hillis W D. Co-evolving parasites improve simulated evolution as an optimization procedure[J]. Physica D，1990，42(1-3)：228-234.

[8] Angeline P J，Pollack J B. Competitive Environments Evolve Better Solutions for Complex Tasks[C]// Proceedings of the 5th International Conference on Genetic Algorithms. Illinois：Springer，1993.

[9] Paredis J. Coevolutionary computation[J]. Artificial life，1995，2(4)：355-375.

[10] Paredis J. Coevolutionary life-time learning [C]//International Conference on Parallel Problem Solving from Nature. Berlin：Springer，1996：72-80.

[11] Rosin C D，Belew R K. New methods for competitive coevolution[J]. Evolutionary computation，1997，5(1)：1-29.

[12] Potter M A，De Jong K A. A cooperative coevolutionary approach to function optimization[C]// International Conference on Parallel Problem Solving from Nature. Berlin：Springer，1994：249-257.

[13] Potter M A，Jong K A D. Cooperative coevolution：An architecture for evolving coadapted subcomponents[J]. Evolutionary computation，2000，8(1)：1-29.

[14] 刘静. 协同进化算法及其应用研究[D]. 西安：西安电子科技大学，2004.

[15] 郭圆平. 动态种群规模的协同进化算法模型，理论与应用[D]. 合肥：中国科学技术大学，2008.

[16] 郑浩然，基于生态特征的进化与协同研究[D]. 合肥：中国科学技术大学，2000.

[17] Cao X B，Li J L，Wang X F. Research on coevolutionary optimization based on ecological cooperation [J]. Journal of software. 2001，12(4)，521-528.

[18] 曹先彬，高隽，王煦法. 基于生态竞争模型的遗传强化学习[J]. 软件学报，1999(06)：99-103.

[19] 曹先彬，罗文坚，王煦法. 基于生态种群竞争模型的协同进化[J]. 软件学报，2001(04)：84-90.

[20] Chen H, Zhu Y, Hu K, et al. Dynamic RFID network optimization using a self-adaptive bacterial foraging algorithm[J]. International Journal of Artificial Intelligence, 2011, 7(11): 219-231.

[21] Chen H, Zhu Y, Hu K, et al. RFID network planning using a multi-swarm optimizer[J]. Journal of Network and Computer Applications, 2011, 34(3): 888-901.

[22] Chen H, Zhu Y, Hu K. Discrete and continuous optimization based on multi-swarm coevolution[J]. Natural Computing, 2010, 9(3): 659-682.

[23] 金莉, 朱云龙, 申海. 三级物流网络选址-路径问题建模与求解算法研究[J]. 控制与决策, 2010 (8): 1195-1200.

[24] 常春光, 朱云龙, 胡琨元, 等. 基于人工免疫算法的精铜板带加工配料优化[J]. 控制与决策, 2010 (7): 1093-1097.

[25] Shen H, Jin L, Zhu Y, et al. Hybridization of particle swarm optimization with the K-Means algorithm for clustering analysis[C]//2010 IEEE Fifth International Conference on Bio-Inspired Computing: Theories and Applications. IEEE, 2010: 531-535.

[26] Shen H, Zhu Y, Jin L, et al. Two-phase heuristic for capacitated vehicle routing problem[C]//2010 Second World Congress on Nature and Biologically Inspired Computing. IEEE, 2010: 534-539.

[27] Ma J, Zhu Y, Shi G. Immune genetic algorithm for flexible job-shop scheduling problem[C]//2010 IEEE International Conference on Automation and Logistics. IEEE, 2010: 486-489.

[28] Chunguang C, Yunlong Z, Kunyuan H, et al. Selective Purchasing Optimization of Raw Materials for Refined Copper Strip by BFO Algorithm [C]//2010 International Conference on Digital Manufacturing & Automation. IEEE, 2010, 2: 337-340.

[29] Chunguang C, Yunlong Z, Kunyuan H, et al. Research on smelting ingredient diluting for refined copper strip by bacteria foraging optimization algorithm [C]//2010 International Conference on Digital Manufacturing & Automation. IEEE, 2010, 2: 275-278.

[30] Chang C, Zhu Y, Hu K, et al. Optimization of grouping batch and sorting order for smelting charges in refined copper strip producing by AIA[C]//2010 IEEE International Conference on Progress in Informatics and Computing. IEEE, 2010, 1: 40-43.

[31] Yan X, Zhu Y, Sui X, et al. Research of manufacturing enterprise information system integration based on extended ESB[C]//International Conference on Applied Informatics and Communication. Springer, Berlin, Heidelberg, 2011: 41-48.

[32] Zou W, Zhu Y, Chen H, et al. Artificial bee colony algorithm based on von neumann topology structure[C]//The Third International Conference on Computer and Electrical Engineering. 2012, 53.

[33] Zou W, Zhu Y, Chen H, et al. Cooperative approaches to artificial bee colony algorithm[C]//2010 International Conference on Computer Application and System Modeling (ICCASM 2010). IEEE, 2010, 9: 944-948.

[34] Ma J, Zhu Y L Shi G, et al. An Adaptive Immune Genetic Algorithm and Its Application for FJSP [C]//Proceedings of the 4th International Conference on Intelligent Information Technology Application.

[35] Ma J, Zhu Y, Shi G. Immune genetic algorithm for flexible job-shop scheduling problem [C]// International Technology & Innovation Conference. IEEE, 2010.

[36] Ma J, Zhu Y, Shi G. Solving the flexible job-shop scheduling problem by immune genetic algorithm [C]//2010 3rd International Conference on Advanced Computer Theory and Engineering. IEEE, 2010, 4: 4215-4218.

[37] Kunyun Hu, Yunlong Zhu, Compound Particle Optimization Using Speciation for Multimodal Function Optimization, 2009 Fifth International Conference on Natural Computation, Tianjin, China,

pp. 182-186.

[38] Chen H N, Zhu Y L, Hu K Y. Compound Particle Optimization Using Speciation for Multimodal Function Optimization[C]//2009 Fifth International Conference on Natural Computation. Tianjin, 182-186.

[39] Xu J W, Jiang B, Zhu Y L, et al. Operation Model for a Class of Manufacturing/Remanufacturing System with Uncertain Demands[C]//Proceedings of International Conference of management engineering and information technology. 2009: 490-494.

[40] Shen H, Zhu Y, Liu T, et al. Particle swarm optimization in solving vehicle routing problem[C]// 2009 Second International Conference on Intelligent Computation Technology and Automation. IEEE, 2009, 1: 287-291.

[41] Zhou X, Zhu Y, Guo H. Research on retailer-driven revenue-sharing contracts model under manufacturers competition [C]//2008 International Seminar on Business and Information Management. IEEE, 2008, 2: 79-83.

[42] Niu B, Zhu Y, He X, et al. A multi-swarm optimizer based fuzzy modeling approach for dynamic systems processing[J]. Neurocomputing, 2008, 71(7-9): 1436-1448.

[43] Chen H, Zhu Y. Optimization based on symbiotic multi-species coevolution[J]. Applied Mathematics and Computation, 2008, 205(1): 47-60.

[44] Chen H N, Zhu Y L, Hu K Y, et al. Global optimization based on hierarchical coevolution model [C]//2008 IEEE Congress on Evolutionary Computation. Hongkong: IEEE, 2008: 1497-1504.

[45] Jin P, Zhu Y. Mining customer change model based on swarm intelligence[C]//International Conference on Intelligent Computing. Springer, Berlin, Heidelberg, 2007: 456-464.

[46] Chen H N, Zhu Y L, Hu K Y, et al. Global optimization based on hierarchical coevolution model [C]//2008 IEEE Congress on Evolutionary Computation. Hongkong: IEEE, 2008: 1497-1504.

[47] Hu Y, Chen H, He M, et al. Multi-Swarm Multi-Objective Optimizer Based on-Optimality Criteria for Multi-Objective Portfolio Management[J]. Mathematical Problems in Engineering, 2019.