

## 高维数据特征选择的多目标蚁群算法

### 3.1 引言

高维数据的特征选择是多目标无序组合优化问题,多目标优化问题的研究由来已久,它是指多个目标不分主次的优化问题<sup>[1]</sup>。多目标优化一般是 NP 难问题,即不能在多项式时间得出最优解,常用的解决多目标优化问题的一类算法称为多目标演化算法(Multiobjective Evolutionary Algorithm),对多目标演化算法的研究一直是学术界的热点问题<sup>[2-3]</sup>,如基于分解的多目标演化算法、非支配排序遗传算法Ⅲ、两档案算法以及多目标蚁群优化(Multiobjective Ant Colony Optimization, MOACO)算法等<sup>[2-7]</sup>。其中 MOACO 在解决多目标组合优化问题方面具有信息正反馈、内在分布式和易于与其他算法结合等特点,吸引了众多研究人员的关注。

多目标组合优化问题包含两种类型:一类是以多目标旅行商问题为代表的有序组合优化问题;另一类是以多目标特征选择问题为代表的无序组合优化问题(多目标子集问题)。由于多目标旅行商问题的解是有序的路径,符合蚂蚁顺序选择路径的过程,因此通过 MOACO 求解多目标旅行商问题是最早的成功案例,对 MOACO 的改进也常以多目标旅行商问题为背景<sup>[8-9]</sup>。然而在处理多目标子集问题方面,当前的 MOACO 算法存在两点局限性:一是多目标子集问题的解是元素的集合,与蚂蚁选择路径的次序无关,目前的 MOACO 算法是将信息素放在物品(元素)上,蚂蚁根据物品上的信息素浓度和启发式信息有序选择路径,从而导致解的无序性和蚂蚁选择物品的有序性之间存在矛盾,即更新信息素对蚂蚁的影响具有不确定性<sup>[10-11]</sup>;二是多数 MOACO 算法解决二目标优化问题,在扩展到三个以上目标的优化问题时其参数较难设置,导致 MOACO 算法难以处理具有较高目标维度的子集问题<sup>[12]</sup>。

本章提出基于等效路径更新的两档案多目标蚁群优化(Multiobjective Ant-colony-

optimization Based on Two Archives and Equivalent Routes Update, MATA&ERU)算法,使用等效路径信息素增强策略,消除解的无序性与蚂蚁求解有序性之间的矛盾,将原目标转换为收敛性目标和多样性目标并设置对应的帕累托档案,提升 MOACO 的扩展性。

## 3.2 理论方法

### 3.2.1 两档案设置

首先介绍多目标优化的相关概念,以最小化优化问题为例,多目标优化问题可由式(3-1)描述

$$\min \mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_m(\mathbf{x}))^T, \quad \mathbf{x} \in \Omega \quad (3-1)$$

式(3-1)中,决策向量  $\mathbf{x} = (x_1, x_2, \dots, x_n)$  属于非空决策空间  $\Omega$ , 目标函数向量  $\mathbf{F}: \Omega \rightarrow \Lambda$  由  $m (m \geq 2)$  个目标组成,  $\Lambda$  是目标空间<sup>[13]</sup>。

帕累托支配 (Pareto Dominance), 给定两个解  $\mathbf{x}, \mathbf{y} \in \Omega$  以及它们对应的目标向量  $\mathbf{F}(\mathbf{x}), \mathbf{F}(\mathbf{y}) \in R^m$ , 当且仅当  $\forall i \in \{1, 2, \dots, m\}, f_i(\mathbf{x}) \leq f_i(\mathbf{y})$  且  $\exists j \in \{1, 2, \dots, m\}, f_j(\mathbf{x}) < f_j(\mathbf{y})$  时,  $\mathbf{x}$  支配  $\mathbf{y}$  (表示为  $\mathbf{x} \succ \mathbf{y}$ )。

帕累托最优解 (Pareto Optimal Solution),  $\mathbf{x}^* \in \Omega$  是帕累托最优解当且仅当不存在另一个解  $\mathbf{x} \in \Omega$  支配它, 帕累托最优解又称非支配解 (Nondominated Solution)。

帕累托集 (Pareto Set), 所有帕累托最优解的并集称为帕累托集  $PS = \{\mathbf{x} \in \Omega \mid \nexists \mathbf{y} \in \Omega, \mathbf{y} \prec \mathbf{x}\}$ 。

帕累托前沿 (Pareto Front), 帕累托集对应的目标向量集合称为帕累托前沿。

近似集合 (Approximation Set), 设  $\mathbf{A} \subset \Lambda$  为目标向量的集合, 用  $\mathbf{A} = \{\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_{|\mathbf{A}|}\}$  表示, 如果  $\mathbf{A}$  中任意一个目标向量都不支配或等于  $\mathbf{A}$  中其他的目标向量, 则  $\mathbf{A}$  称为一个近似集合。

多目标演化算法包括两个性能目标: 收敛性目标, 近似集合尽可能趋近帕累托前沿; 多样性目标, 近似集合尽可能覆盖帕累托前沿<sup>[14]</sup>。

为了在多目标优化问题中获得较好的性能, 当前 MOACO 算法主要从三方面进行改进: 第一是根据目标个数设置对应的信息素矩阵, 使用权重和、权重积或随机法将多个信息素值聚合成单个值; 第二是设置多个启发式矩阵, 并通过权重和、权重积或随机法聚合启发式值; 第三是引入多群体, 每个群体搜索目标空间的一部分, 群体之间采用某种方式交互信息。以上三种改进方案均有不同程度的局限性。首先, 当目标个数较多时, 使用多个信息素矩阵会导致聚合参数难以设置; 其次, 相关研究表明设置多个启发式信息矩阵不能显著提升 MOACO 的性能<sup>[15]</sup>; 最后, 设置多群体需要通过权重向量分解目标空间, 该方法一般用于二目标优化问题, 当目标个数较多时, 该方法难以拓展使用。

通过多目标演化算法的性能目标可知, 无论存在几个原始的优化目标, 都可以将其转换为收敛性目标和多样性目标, 即算法获得的帕累托解尽可能趋近真实的帕累托前沿, 同时尽

可能覆盖真实帕累托前沿。有趣的是,这是个二目标“优化”问题,通过当前 MOACO 常用的方法对这两个目标进行优化求解即可,并且能够将算法灵活拓展到高维目标空间。具体而言,在全局范围内设置两个帕累托档案,即收敛性档案(Converge Archive)与多样性档案(Diversity Archive),收敛性档案提高算法的收敛能力,多样性档案提高算法获取解的分布多样性,与设置的两档案相对应,需要设置两个全局信息素矩阵,称为收敛性信息素矩阵与多样性信息素矩阵,信息素矩阵采用对应的档案解进行更新,即收敛性档案中的解更新收敛性信息素矩阵,多样性档案中的解更新多样性信息素矩阵,算法迭代结束后,输出收敛性档案作为最终帕累托解集。

### 3.2.2 等效路径信息素增强策略

在 MOACO 中,蚂蚁依次选择路径构建解,多目标子集问题是典型的无序组合优化问题,其解集内的元素无顺序关系,蚂蚁有序的选择与解的无序性之间存在矛盾,解决该问题是进一步提升算法求解性能的关键。引入基于图的蚂蚁系统<sup>[16]</sup>,构造包含一个有向图和一些从有向图到多目标子集问题映射的构造图,蚂蚁根据有向图边上的信息素值与静态启发式信息构建可行解。

**定义 3-1** 定义多目标子集问题的构造图 $(G, \Gamma, \Gamma')$ ,有向图 $G=(V, E)$ , $\Gamma$ 和 $\Gamma'$ 是映射关系,并且满足以下几点:

(1) 备选解(路径)的集合 $E=\{e_{ij} \mid i=1,2,\dots,n, j=1,2,\dots,q\}$ ,蚂蚁访问路径的长度集合 $D=\{d_k \mid k=1,2,\dots,q+1\}$ ,其中 $n$ 为物品的规模数, $q$ 为物品可能的最大规模数且有 $q \leq n$ , $e_{ij}=\langle d_j, d_{j+1} \rangle, j=1,2,\dots,q$ 表示蚂蚁第 $j$ 步选择(访问)第 $i$ 个物品;

(2) 第 $k$ 步 $d_k, k=1,2,\dots,q+1$ 的入度为 $n$ ,第 $k$ 步 $d_k, k=1,2,\dots,q$ 的出度为 $n$ ,第1步 $d_1$ 的入度为0,第 $q+1$ 步 $d_{q+1}$ 的出度为0;

(3) 令 $W=\{\omega_l \mid l=1,2,\dots\}$ 为蚂蚁访问完有向图 $G$ 后搜索的路径解且 $\omega_l$ 满足:

① 从有向图的第1步 $d_1$ 开始搜索,可以在任意步终止访问;

② 搜索包含有向图的边 $e_{ij}, e_{i'j'}$ ,且满足 $i \neq i', j' \neq j$ ;

③  $W$ 为有向图边构成的路径解集且满足多目标子集问题的约束条件。

(4)  $\Gamma$ 将 $W$ 映射为多目标子集问题的可行解集, $\Gamma'$ 将 $W$ 映射为目标函数值。

根据定义 3-1,多目标子集问题构造图的有向图表示如图 3-1 所示。

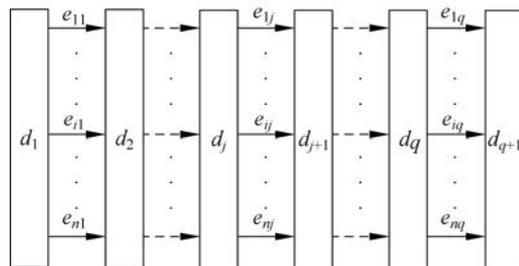


图 3-1 多目标子集问题构造图

文献[17]为了提高蚁群算法求解子集问题的性能,基于图的蚂蚁系统提出一种称为等效路径的信息素增强策略,同时更新有向图边的等效路径上的信息素值。但该方法仅用于

解决单目标优化问题,本章将其扩展到多目标优化问题中,解决蚂蚁构造解的有序性与多目标子集问题解的无序性之间的矛盾,下面给出等效路径的概念。

**定义 3-2** 若  $w_l, w_{l'} \in W$  且满足  $\Gamma'(w_l) = \Gamma'(w_{l'})$ , 则解  $\Gamma(w_l)$  和  $\Gamma(w_{l'})$  等价, 且称  $w_l, w_{l'}$  为等效路径。

**定理 3-1** 若有  $w_l \in W, |w_l| = L, w_l = (e_{i_1 1}, e_{i_2 2}, \dots, e_{i_L L}), w_l$  包含的边集为  $\Psi(w_l) = \{e_{ij} | i = i_1, i_2, \dots, i_L, j = 1, 2, \dots, L\}$ , 多目标子集问题的解为  $\Gamma(w_l)$  且目标函数值为  $\Gamma'(w_l)$ , 则在有向图  $G$  中至少有  $L!$  条等效路径。

**证明:** 根据定义 3-1 及集合的性质可知, 由于边集  $i_1, i_2, \dots, i_L$  相互之间无顺序关系, 因此它们的全排列组合有且仅有  $L!$  个, 边集  $\Psi(w_l)$  包含的边至多可以组成  $L!$  条长度为  $L$  路径, 则这些路径的解为  $\Gamma(w_l) = \{i_1, i_2, \dots, i_L\}$  且目标函数值均为  $\Gamma'(w_l)$ , 即这  $L!$  条边为等效路径。

下面举例说明传统信息素更新方式与等效路径信息素更新方式的差异, 假设在物品个数  $n = 10$ , 且物品可能被选的最大个数  $q = 5$  的前提下, 蚂蚁搜索的一个可行解为  $\{1, 3, 4, 5, 9\}$ , 在构造图中采用传统信息素更新方式与等效路径信息素更新方式的示例分别如图 3-2 和图 3-3 所示。

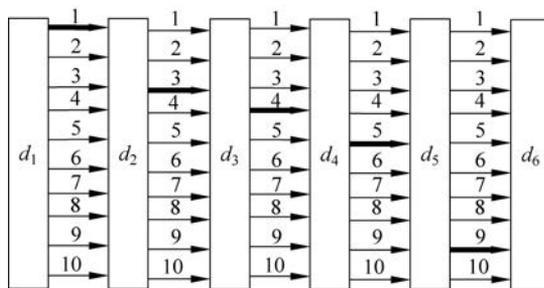


图 3-2 传统信息素更新方式示意图

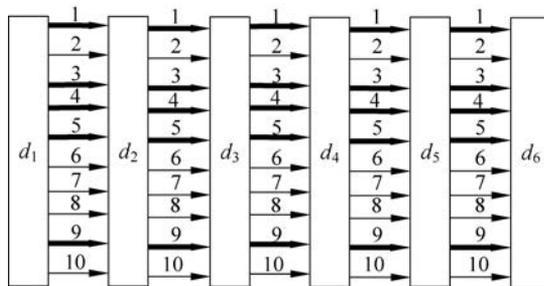


图 3-3 等效路径信息素更新方式示意图

在图 3-2 中, 通过传统方式更新信息素矩阵时, 由于可行解为  $\{1, 3, 4, 5, 9\}$ , 因此只增强对应边上的信息素值; 而在图 3-3 中, 通过等效路径方式更新信息素时, 需要增强与路径解  $\{1, 3, 4, 5, 9\}$  等价的路径上的信息素值 (有  $5! = 120$  条等效路径), 这将同时提高使蚂蚁选择  $5!$  条等效路径的概率, 即蚂蚁选择路径解  $\{1, 3, 4, 5, 9\}$  的概率得到了提升。因此, 等效路径信息素更新策略能够有效消除蚂蚁构造解的有序性与多目标子集问题解的无序性之间的矛盾。

### 3.2.3 多样性度量指标

由于在目标维数较高时,多目标演化算法的多样性恶化更为严重,因此多样性的提升是算法需要解决的重要方面<sup>[18]</sup>。

多目标演化算法有两种进化方式,一种是个体附带解的信息,通过迭代提高自身解的质量,如基于遗传操作的演化算法、粒子群算法、蝙蝠算法等<sup>[19,20]</sup>;另一种是采用类似“记忆库”的结构保存较好解的历史信息,并利用“记忆库”中的信息生成新的较好解,MOACO 以及和声搜索算法是这种方式的典型代表<sup>[21]</sup>。由于 MOACO 使用信息素矩阵保留较好解在源空间的历史信息,并使用信息素矩阵生成新解,因此若仅利用帕累托支配关系或距离函数等方法评估解在目标空间的多样性适应度值,可能会造成算法多次更新相同路径,从而导致多样性的恶化,下面通过例子说明该问题。

假设对于一个最小化二目标优化问题,算法搜索到 3 个解  $x, y, z$  (其中  $x$  为本次已加入帕累托档案中的新解),其在目标空间的适应度值分别为  $F(x) = (5, 4), F(y) = (4, 7), f(z) = (5, 5)$ ,它们在源空间中对应的路径分别为  $x = \{1, 3, 4, 6\}, y = \{1, 3, 4, 9\}, z = \{1, 5, 7, 9\}$ 。通过帕累托关系可以看出,解  $x$  与  $y$  互不支配,而  $x$  支配  $z$ ,因此更新帕累托档案时应当保留解  $y$  而放弃解  $z$ 。但是通过观察可以发现, $x$  与  $y$  在源空间的路径具有较高的相似度,而  $x$  与  $z$  在源空间的路径相似度较低,若保留  $y$  而放弃  $z$  会造成 MOACO 使用帕累托档案中的解更新信息素矩阵时多次更新路径  $\{1, 3, 4\}$ ,从而增加蚂蚁选择该路径的概率,降低算法的多样性。

为此,提出一种结合目标空间帕累托支配关系与源空间距离信息的多样性度量指标,解决 MOACO 在源空间可能多次更新相同路径的问题。

令  $P_t$  表示第  $t$  次迭代时算法生成的解集, $\bar{P}_t$  表示当前迭代需要更新的帕累托档案,为  $P_t$  和  $\bar{P}_t$  中的个体分配强度值  $S$ ,定义个体解  $i$  的强度值  $S(i)$  表示个体  $i$  支配其他解的数量,如式(3-2)

$$S(i) = |\{j \mid j \in P_t + \bar{P}_t \wedge i \succ j\}| \quad (3-2)$$

式(3-2)中, $|\cdot|$  表示集合的基数, $+$  表示集合的并操作,符号  $\succ$  表示帕累托支配。基于式(3-2),定义个体  $i$  的粗糙适应度  $R(i)$ ,如式(3-3)

$$R(i) = \sum_{j \in P_t + \bar{P}_t, j \succ i} S(j) \quad (3-3)$$

式(3-3)中,个体  $i$  的粗糙适应度  $R(i)$  由在  $P_t$  和  $\bar{P}_t$  中支配  $i$  的个体决定。因此对个体  $i$ ,粗糙适应度  $R(i)$  越小表明  $i$  的适应度越好,当  $R(i) = 0$  时,表示没有任何解支配  $i$ ,即  $i$  是帕累托解。

为个体分配粗糙适应度是一种基于帕累托关系的小生境方法,在算法迭代初期,帕累托解较少,随着迭代的进行,帕累托解的数量随之增多,算法选择压力减小,使用帕累托关系就无法有效选择帕累托解,因此需要结合解在源空间的位置信息来选择更为合适的帕累托解。如前文所述,MOACO 的蚂蚁在迭代完成后死亡,解的历史信息通过其在源空间的位置(路

径)更新信息素矩阵进行存储,因此采用个体解在源空间的位置信息选择帕累托解更符合 MOACO 的自然属性。具体而言,由于在多目标子集问题中,解的元素无顺序关系,因此可以使用谷元距离(Tominato Distance)度量解在源空间的位置信息<sup>[22]</sup>。令  $w_i, w_j$  分别为多目标子集问题的 2 个可行解  $i, j$  在源空间的路径解,它们的谷元距离如式(3-4)所示。

$$D_{ij} = 1 - \frac{|w_i| + |w_j| - 2|w_i \cap w_j|}{|w_i| + |w_j| - |w_i \cap w_j|} \quad (3-4)$$

式(3-4)中, $w_i \cap w_j$  为  $w_i$  和  $w_j$  的交集,且  $D_{ij} \in [0, 1]$ ,当  $w_i$  与  $w_j$  完全相同时, $D_{ij} = 1$ ,当  $w_i$  与  $w_j$  完全不同时, $D_{ij} = 0$ ,因此  $D_{ij}$  越小表明两个解之间的位置距离越大,在源空间的多样性越好。

综上,结合帕累托支配关系度量指标与距离度量指标,定义个体解  $i$  的多样性度量指标  $\gamma$ ,如式(3-5)所示。

$$\gamma(i) = R(i) + \Xi_i^h \quad (3-5)$$

式(3-5)中, $\Xi_i^h$  表示个体  $i$  对其他解的距离  $D_{ij}$  按照升序排列后的第  $h$  个值,其中  $h$  为算法运行参数,论文设为档案中个体数量的均方根  $\sqrt{|P_t| + |\bar{P}_t|}$ 。

### 3.3 算法描述

本节描述 MATA&ERU 算法及其组成部分,包括路径选择概率的计算、档案的更新以及信息素矩阵的更新。路径选择概率决定蚂蚁以较大的可能性选择较好路径;档案的更新决定解在目标空间的分布,档案中的候选解通过更新信息素矩阵影响算法的求解性能;信息素矩阵更新的关键在于如何将档案中的解转换成源空间的路径信息,并通过更新信息素矩阵保存该信息。

#### 3.3.1 路径选择概率公式

在 MATA&ERU 中,蚂蚁通过计算转移概率选择路径逐步构建可行解,在  $t$  时刻,蚂蚁  $k$  从  $d_j$  经过边  $e_{ij}$  到达  $d_{j+1}$  的路径选择概率如式(3-6)所示。

$$p_{ij}^k(t) = \begin{cases} \frac{[\tau_{ij}(t-1)]^\alpha \cdot [\eta_i]^\beta}{\sum_{e_{ij} \notin \text{visit}_k} [\tau_{ij}(t-1)]^\alpha \cdot [\eta_b]^\beta}, & e_{ij} \notin \text{visit}_k \\ 0, & \text{其他} \end{cases} \quad (3-6)$$

式(3-6)中, $\tau_{ij}(t-1)$ 是边  $e_{ij}$  在  $(t-1)$ 时刻的信息素值, $\eta_i$  是静态启发式信息,表示选择物品  $i$  的期望, $\alpha$  是信息素值重要程度因子, $\beta$  是启发式信息重要程度因子,visit <sub>$k$</sub>  表示蚂蚁  $k$  访问过的边集。由于 MATA&ERU 在全局范围内设置两个信息素矩阵,因此需要将两个矩阵中的信息素值聚合成单个值计算路径选择概率,聚合方式采用式(3-7)实现。

$$\tau_{ij}(t-1) = [\tau_{ij}^{\text{ca}}(t-1)]^{(1-\lambda)} \cdot [\tau_{ij}^{\text{da}}(t-1)]^\lambda \quad (3-7)$$

式(3-7)中,  $\tau_{ij}^{ca}(t-1)$  表示边  $e_{ij}$  在  $(t-1)$  时刻的收敛性信息素值,  $\tau_{ij}^{da}(t-1)$  表示边  $e_{ij}$  在  $(t-1)$  时刻的多样性信息素值,  $\lambda$  是权重参数且有  $0 \leq \lambda \leq 1$ , 这里设置  $\lambda = 0.5$ , 表示收敛性信息与多样性信息具有相同的重要程度。

### 3.3.2 变异机制

在单目标优化问题中, 演化算法的目标是获得一个较好的次优解, 因此收敛性是算法主要考虑的方面。而在多目标优化问题中, 多目标演化算法的目标是获得尽量靠近帕累托前沿且分布较好的近似集合, 因此多样性也是多目标演化算法需要考虑的重要方面。多目标演化算法的多样性维持主要来自两方面, 一是算法能够生成具有较多样性的后代解, 二是在迭代过程中算法对帕累托档案的更新, 即选择具有较多样性的解。在 MOACO 中, 由于蚂蚁根据信息素值与启发式信息计算路径选择概率构建新解, 因此蚂蚁会以较高的概率选择较好路径, 导致算法生成后代解的多样性较弱。为了进一步提高蚂蚁生成后代解的多样性, MATA&ERU 引入遗传算法的变异机制, 在新生成的解上进行变异操作, 生成具有较多样性的后代解, MATA&ERU 的变异机制描述如表 3-1 所示。

表 3-1 变异机制伪代码

输入: 新生成的路径解, 变异阈值 $\sigma$ , 备选路径长度 $n$ ;	
输出: 变异解	
1.	将路径解转换成长度为 $n$ 的染色体, 选中的路径对应位值为 1, 否则为 0;
2.	<b>for</b> 位置索引 $g = 1 : n$ <b>do</b>
3.	<b>if</b> 随机数 $\text{rand} > \sigma$ <b>do</b>
4.	当前位置值取反;
5.	<b>end if</b>
6.	<b>end for</b>
7.	若变异结果为不可行解, 则调整为可行解并转换为路径解

表 3-1 中第 1 行是将蚂蚁的路径解转换为染色体, 第 2~6 行是逐位访问染色体并进行变异操作, 第 3、4 行是变异操作, 即若当前随机值大于设定阈值, 则对当前位的值取反(1 变为 0, 0 变为 1), 第 7 行是将变异解调整为可行解并转换为蚂蚁的路径解进行输出, 可行解的调整过程采用 Li K 等使用的方法<sup>[23]</sup>。

### 3.3.3 两档案更新

每轮迭代后, 需要更新全局收敛性档案与多样性档案中的帕累托解, MATA&ERU 固定档案规模, 档案中解的个数超过设定规模时, 需要删除多余解。当前存在一类能够综合反映解集质量的指标函数, 其中最具代表性的是 hypervolume 指标和 epsilon 指标, 这些指标与帕累托关系存在严格的单调性, 即指标数值越高, 帕累托解在收敛性与多样性方面的质量越好<sup>[24-25]</sup>。但是由于 hypervolume 指标计算复杂度较高, 因此在实际中常用于评估解集的质量, 而 epsilon 指标的计算复杂度较低, 因此 MATA&ERU 将 epsilon 指标作为收敛性目标函数, 评

估收敛性档案中解的适应度,保留具有较好收敛性的解。epsilon 指标的定义如式(3-8)

$$I_{\epsilon+}(x_1, x_2) = \min_{\epsilon} (f_i(x_1) - \epsilon \leq f_i(x_2)), \quad i \in \{1, \dots, m\} \quad (3-8)$$

式(3-8)表示解  $x_1$  弱支配解  $x_2$  需要移动最短距离值  $\epsilon$ 。根据 epsilon 指标定义,为个体解  $x_1$  分配适应度的方法如式(3-9)

$$\Theta(x_1) = \sum_{x_2 \in P \setminus \{x_1\}} -e^{-I_{\epsilon+}(x_2, x_1)/k} \quad (3-9)$$

式(3-9)中,  $k$  是算法运行参数,一般设置为 0.05。

综上,收敛性档案中的帕累托解更新伪代码如表 3-2 所示。

表 3-2 收敛性档案更新伪代码

输入: 待更新的收敛性档案 $A_c$ , 收敛性档案规模 $N_{ca}$
输出: 经过更新的收敛性档案 $A_c$
<pre> 1. while <math> A_c  &gt; N_{ca}</math> do 2.     选择 <math>\Theta(x^*)</math> 值最小的解 <math>x^*</math>; 3.     将个体解 <math>x^*</math> 从收敛性档案 <math>A_c</math> 中删除; 4.     通过式 <math>\Theta(x) = \Theta(x) + e^{-I_{\epsilon+}(x^*, x)/k}</math> 更新档案中剩余个体解的适应度值; 5. end while </pre>

表 3-2 中,第 1 行循环判断当前档案中个体解的规模是否超过设定阈值,第 2~4 行是通过计算个体适应度值逐一删除档案中的个体解。

多样性档案使用本章提出的多样性度量指标  $\gamma$  进行更新,与收敛性档案逐个删除多余解的方式不同,多样性档案通过逐一选择个体解的方式进行更新,其伪代码描述如表 3-3 所示。

表 3-3 多样性档案更新伪代码

输入: 待更新的多样性档案 $A_d$ , 多样性档案规模 $N_{da}$
输出: 经过更新的多样性档案 $A_d$
<pre> 1. if <math> A_d  &gt; N_{da}</math> do 2.     计算档案 <math>A_d</math> 中每个解在多样性指标 <math>\gamma</math> 上的值; 3.     按照 <math>\gamma</math> 值对 <math>A_d</math> 中的每个解进行降序排列, 并选择前 <math>N_{da}</math> 个解; 4.     清空档案 <math>A_d</math> 并使用选择的 <math>N_{da}</math> 个解组成新的多样性档案 <math>A_d</math>; 5. end if </pre>

表 3-3 中,第 1 行判断当前档案中的个体解规模是否超过设定阈值,第 2~4 行通过计算个体解的多样性评估值,选择满足档案规模的个体解组成新的多样性档案。

### 3.3.4 信息素更新方式

收敛性档案与多样性档案更新完成后,需要更新全局收敛性信息素矩阵和多样性信息素矩阵, MATA&ERU 使用收敛性档案中的解更新收敛性信息素矩阵,多样性档案中的解更新多样性信息素矩阵。

档案中的路径解按照 3.2.2 节等效路径信息素增强策略更新信息素矩阵,若在  $l$  时刻拟对路径  $\text{tabu}^l$  上的信息素值进行更新,更新方法如式(3-10)

$$\tau_{ij}(t) = \begin{cases} (1 - \rho)\tau_{ij}(t - 1) + \Delta'(\text{tabu}^l), & e_{ij} \in \Psi(\text{tabu}^l) \\ (1 - \rho)\tau_{ij}(t - 1), & \text{其他} \end{cases} \quad (3-10)$$

式(3-10)中, $\rho$  是信息素挥发系数, $\Delta'(\text{tabu}^l)$  是信息素更新增量, $\Psi(\text{tabu}^l)$  是  $\text{tabu}^l$  的等效路径, $\Delta'(\text{tabu}^l)$  的计算通过式(3-11)实现

$$\Delta'(\text{tabu}^l) = Q \times Sg(\text{tabu}^l) / NA \quad (3-11)$$

式(3-11)中, $NA$  表示档案中解的个数(为  $N_{ca}$  或  $N_{da}$ ), $Sg$  为路径  $\text{tabu}^l$  的评估值(收敛性评估值或多样性评估值), $Q$  是常数,根据  $\rho$  值确定。

### 3.3.5 算法伪代码及时间复杂度分析

综上所述,MATA&ERU 算法的伪代码描述如表 3-4 所示。

表 3-4 MATA&ERU 伪代码

输入: 收敛性档案 $A_c$ , 收敛性档案规模 $N_{ca}$ , 多样性档案 $A_d$ , 多样性档案规模 $N_{da}$ , 蚂蚁数量 $N^a$ , 变异阈值 $\sigma$ , 最大迭代次数 $NC$	
输出: 多样性档案 $A_d$	
1.	<b>while</b> 迭代次数 $\text{iter} < NC$ <b>do</b>
2.	<b>for</b> 蚂蚁 $a = 1 : N^a$ <b>do</b>
3.	蚂蚁根据式(3-6)构建个体解,在个体解上运行变异机制;
4.	<b>end for</b>
5.	更新收敛性档案 $A_c$ , 更新多样性档案 $A_d$ ;
6.	分配候选解,并采用式(3-10)更新信息素矩阵;
7.	<b>end while</b>

表 3-4 中第 1 行是算法迭代循环次数判断,第 2~4 行是每只蚂蚁构建解的过程,第 5 行更新两个帕累托档案,第 6 行是利用档案中的解更新对应的信息素矩阵。

下面分析 MATA&ERU 算法的时间复杂度。假设问题解的候选基数为  $n$ ,则初始化全局两个信息素矩阵的时间复杂度为  $O(2 \times n^2)$ ,所有蚂蚁构建可行解的时间复杂度为  $O(N^a \times n^2)$ ,变异机制的时间复杂度为  $O(N^a \times n)$ ,更新收敛性档案的时间复杂度为  $O(N_{ca}^2)$ ,更新多样性档案的时间复杂度为  $O(N_{da}^2)$ ,更新两个信息素矩阵的时间复杂度为  $O(2 \times n^2)$ ,算法总体的时间复杂度为  $O(\max(NC \times N^a \times n^2, NC \times N_{ca}^2, NC \times N_{da}^2))$ 。

## 3.4 实验与分析

为了验证 MATA&ERU 的有效性和优越性,选用基于指标函数蚁群优化 IBACO、多准则网站优化 MCWSO、多目标蚁群优化  $mACO-4$ 、非支配遗传算法 NSGA II 和基于分解

的多目标演化算法 MOEA/D 等 5 种算法进行对比<sup>[6,12,15,26-27]</sup>。IBACO 是基于指标函数的多目标蚁群算法,它利用 epsilon 指标评估解的适应度,具有良好的收敛性;MCWSO 和 mACO-4 是具有代表性的多目标蚁群算法,它们在解的收敛性和分布性上都具有优异的表现;NSGA II 和 MOEA/D 是基于遗传操作的典型连续域优化算法。

采用 ETH 网站中的 12 个标准多目标高维背包问题(<http://www.tik.ee.ethz.ch/sop/download/supplementary/testProblemSuite/>)作为多目标子集问题(高维数据的多目标特征选择问题)的应用实例进行测试。IBACO、MCWSO 和 mACO-4 的参数设置与原文一致,NSGA II 和 MOEA/D 的交叉率为 0.7,变异率为 0.01,它们的迭代次数设置与 MATA&ERU 一致。MATA&ERU 的相关参数如下: $\alpha=1, \beta=8, \rho=0.01, Q=500$ ,变异阈值  $\sigma=0.8$ ,信息素初始化浓度  $\tau_{ij}(0)=100$ ,收敛性档案与多样性档案的规模设置与蚂蚁个数相同,实验独立运行 10 次,测试实例的相关信息和 MATA&ERU 中蚂蚁个数及迭代次数设置如表 3-5 所示。

表 3-5 多目标背包问题测试实例及参数设置

序号	物品个数	目标(背包)个数	蚂蚁个数	迭代次数	实例表示
1	100	2	120	1000	100-2
2	250	2	150	1500	250-2
3	500	2	200	1700	500-2
4	750	2	250	2000	750-2
5	100	3	250	1500	100-3
6	250	3	350	1500	250-3
7	500	3	350	1500	500-3
8	750	3	350	2000	750-3
9	100	4	350	1700	100-4
10	250	4	450	1700	250-4
11	500	4	450	1700	500-4
12	750	4	450	2000	750-4

针对多目标背包问题的特点,MATA&ERU 的启发式信息  $\eta_i$  由式(3-12)给出

$$\eta_i = \sum_{b=1}^m (P_{bi}/W_{bi}/V_b), \quad i=1,2,\dots,n. \quad (3-12)$$

式(3-12)中, $P_{bi}$  表示第  $b$  个背包(目标)中第  $i$  个元素(物品)的价值, $W_{bi}$  表示第  $b$  个背包中第  $i$  个元素的重量, $V_b$  表示第  $b$  个背包的容量。

为了综合评价算法的收敛性与多样性,使用 3 种指标对算法进行评估,即覆盖 C 指标、分散指标(Spacing,SP)和反向世代距离(Inverted Generational Distance,IGD),下面分别介绍这 3 种度量指标。

$C$  指标用来度量解的收敛性, 设对比算法的解集为  $X'$  和  $X''$ ,  $C$  指标的计算方式如式(3-13)

$$C = (X', X'') = \frac{|\{a'' \in X'' : \exists a' \in X', a' \succ a''\}|}{|X''|} \quad (3-13)$$

当  $C(X', X'') = 1$  时,  $X''$  中的所有点等于  $X'$  中的点或被  $X'$  中的点支配, 当  $C(X', X'') = 0$  时,  $X''$  中没有点被  $X'$  中的点支配。值得注意的是,  $C(X', X'')$  和  $C(X'', X')$  需要同时考虑, 因为  $C(X', X'') \neq 1 - C(X'', X')$ 。

分散指标 SP 用来度量解的多样性, 其定义如式(3-14)

$$SP = \sqrt{\frac{1}{g-1} \sum_{i=1}^{g-1} (d_i - \bar{d})^2} \quad (3-14)$$

式(3-14)中,  $g$  是算法获得解的个数,  $d_i$  是第  $i$  个解到解集中最近点的欧氏距离,  $\bar{d}$  是  $d_i$  的均值, 因此 SP 为  $d_i$  的均方差, 它反映了解集的分布性, 且 SP 值越小, 表明解集的多样性越好。

设  $S$  是算法获得的解集, 其 IGD 值的计算方法如式(3-15)

$$IGD(S, P^*) = \frac{\sum_{x \in P^*} \text{dist}(x, S)}{|P^*|} \quad (3-15)$$

式(3-15)中,  $P^*$  是帕累托前沿的均匀抽样解集,  $|P^*|$  是  $P^*$  的基数,  $\text{dist}(x, S)$  是解  $x$  与解集  $S$  中最近点之间的欧氏距离, IGD 是一种综合度量指标, 能同时度量收敛性与多样性。

MATA&ERU 及 5 种对比方法在 4 个二目标测试实例上的结果如图 3-4 所示, 其中  $f_1$  和  $f_2$  表示两个目标的评估值。

通过图 3-4 中 6 种方法的测试结果可以看出, 尽管在 100-2 测试数据上结果较好, 但是在其他 3 个二目标测试实例上, NSGA II 和 MOEA/D 在收敛性和多样性上具有较弱的性能表现。继续观察 IBACO、MCWSO 和  $m$ ACO-4 的结果, 可以看出, 在 100-2 测试实例上, 它们能够搜索到较好的帕累托前沿, 但是在多样性分布上效果较弱; 在 250-2、500-2 和 750-2 测试实例上, 这 3 种方法只能搜索到一部分帕累托前沿, 说明随着测试实例复杂性的增加, 算法的搜索性能有所下降, 特别是在 750-2 的测试实例上, MCWSO 和  $m$ ACO-4 的多样性能力下降最为显著。最后观察 MATA&ERU 的表现, 可以发现, 在所有二目标的测试实例中, MATA&ERU 的性能最好, 特别是具有较强的收敛能力, 即在所有对比算法中最接近帕累托前沿。但是在 250-2、500-2 和 750-2 测试实例上, MATA&ERU 的多样性能力也有所下降, 这表明测试实例复杂性的增加对算法多样性的影响更为显著。

下面比较算法的收敛能力, 为了方便表示, 使用数字代表各个算法, 即 1 表示 MATA&ERU、2 表示 IBACO、3 表示 MCWSO、4 表示  $m$ ACO-4、5 表示 NSGA II、6 表示 MOEA/D。C 指标中,  $C(1, 2)$  表示算法 1 对算法 2 的收敛能力,  $C(2, 1)$  表示算法 2 对算法 1 的收敛能力, C 指标值越大, 表明算法的相对收敛能力越强。MATA&ERU 与 5 种对比方法在收敛能力上的比较结果如表 3-6 所列。

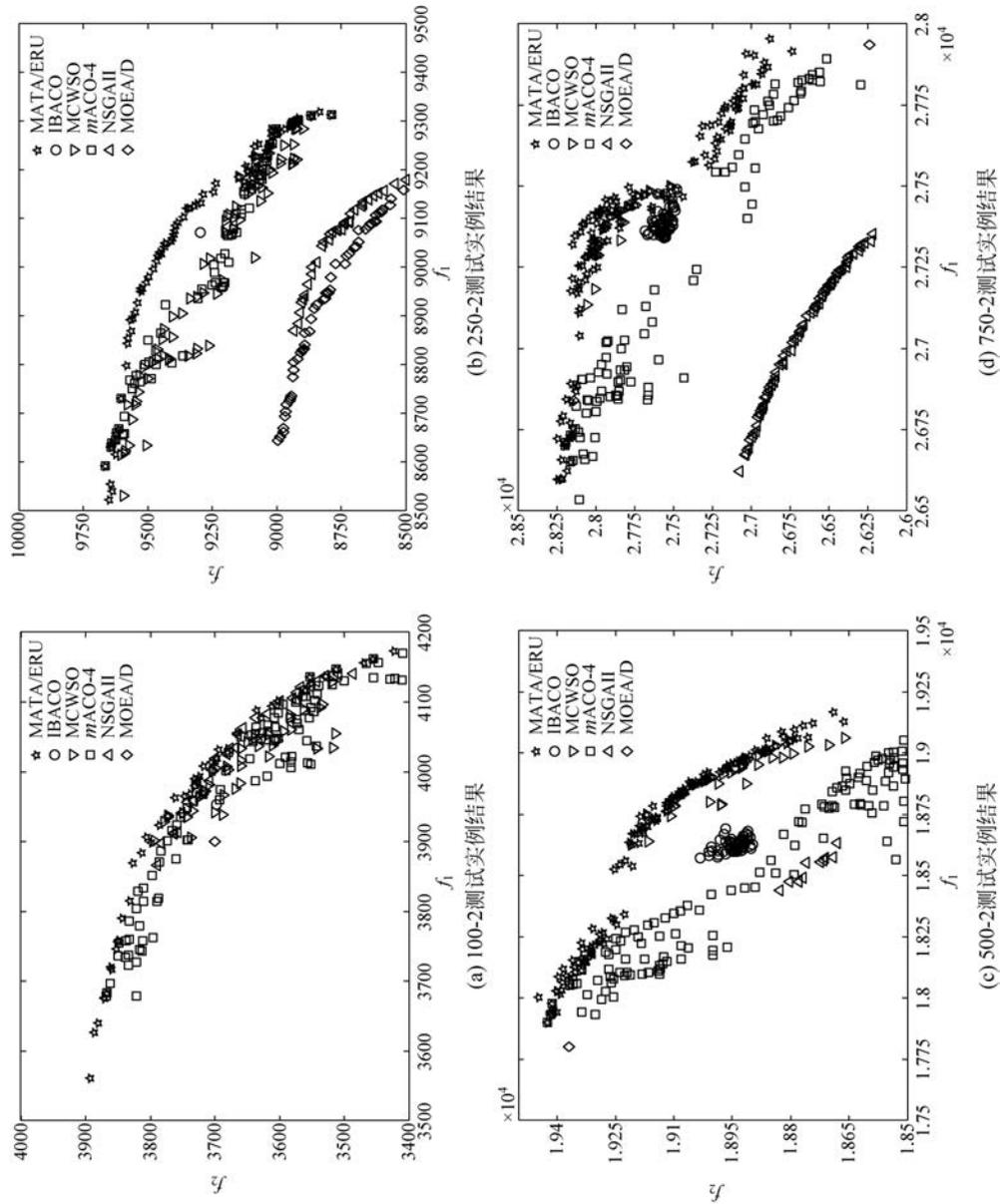


图 3-4 二目标测试实例结果

表 3-6 对比算法 C 指标均值

测试实例	$C(1,2)$ $C(2,1)$	$C(1,3)$ $C(3,1)$	$C(1,4)$ $C(4,1)$	$C(1,5)$ $C(5,1)$	$C(1,6)$ $C(6,1)$
100-2	<b>0.1500</b>	<b>0.3388</b>	<b>0.2162</b>	<b>0.8864</b>	0
	0	0	0.0167	0	0
250-2	<b>0</b>	<b>0.9222</b>	<b>0.9333</b>	<b>1</b>	0
	0	0.0067	0.0467	0	0
500-2	0	0	0	1	1
	0	0	0	0	0
750-2	<b>0.1000</b>	<b>0.6506</b>	<b>0.6627</b>	<b>1</b>	<b>1</b>
	0.0120	0.0040	0.0120	0	0
100-3	<b>0.2125</b>	<b>0.1401</b>	<b>0.3867</b>	0.0381	0
	0	0	0	<b>0.1960</b>	0
250-3	<b>0.8250</b>	<b>0.0492</b>	<b>0.1773</b>	<b>1</b>	<b>1</b>
	0	0	0	0	0
500-3	0	<b>0.3055</b>	<b>0.1851</b>	<b>1</b>	<b>1</b>
	0	0.0029	0.0143	0	0
750-3	<b>0.7750</b>	<b>0.7879</b>	<b>0.4660</b>	<b>1</b>	<b>1</b>
	0	0	0	0	0
100-4	<b>0.1900</b>	<b>0.9057</b>	<b>0.1611</b>	<b>0.6239</b>	<b>0.0875</b>
	0	0	0.0057	0.0057	0
250-4	<b>0.0200</b>	<b>0.0938</b>	<b>0.0686</b>	<b>0.8235</b>	<b>1</b>
	0	0	0	0	0
500-4	<b>0.4300</b>	<b>0.0565</b>	<b>0.0875</b>	<b>1</b>	<b>1</b>
	0	0	0	0	0
750-4	0	<b>0.0206</b>	<b>0.0654</b>	<b>1</b>	<b>1</b>
	0	0	0.0178	0	0

从表 3-6 中可以看出,除了在 250-2、500-2、500-3 和 750-4 测试实例上, MATA&ERU 的收敛能力都优于 IBACO; 与 MCWSO 和  $mACO-4$  相比,除了在 500-2 数据集上, MATA&ERU 都具有更好的收敛性能; 最后对比 NSGA II 和 MOEA/D 的收敛能力,除了 100-3 测试实例,在多数测试条件下, MATA&ERU 的收敛性能都显著优于 NSGA II 和 MOEA/D。

下面分析 MATA&ERU 及 5 种对比算法的多样性,它们的  $SP$  值测试结果如表 3-7 所列。

表 3-7 对比算法 SP 均值

测试实例	MATA&ERU	IBACO	MCWSO	$mACO-4$	NSGA II	MOEA/D
100-2	29.2911	<b>3.3882</b>	15.7200	32.0441	35.8174	118.2776
250-2	44.1053	<b>44.0115</b>	96.9568	58.4064	53.6652	357.5405
500-2	40.9948	<b>10.0474</b>	26.4808	73.6211	1689.2770	589.1191
750-2	46.1012	<b>16.3309</b>	115.2252	109.7494	35.5253	818.0353
100-3	<b>13.5832</b>	41.9443	14.0613	30.7693	48.7180	165.9507

续表

测试实例	MATA&ERU	IBACO	MCWSO	<i>m</i> ACO-4	NSGA II	MOEA/D
250-3	<b>9. 8495</b>	75. 7193	19. 6290	36. 5164	55. 9837	483. 9021
500-3	<b>26. 9408</b>	89. 9247	28. 3948	59. 1194	2472. 0619	696. 7934
750-3	<b>26. 3993</b>	59. 1731	76. 5131	76. 3955	2340. 5325	917. 5688
100-4	<b>19. 8563</b>	53. 8995	27. 3030	42. 3193	64. 9534	171. 8453
250-4	<b>27. 3680</b>	67. 0404	42. 8691	37. 0444	75. 5752	373. 0973
500-4	<b>33. 9944</b>	89. 8625	16. 4992	57. 4618	1015. 3691	848. 4129
750-4	<b>29. 8117</b>	111. 7552	37. 9865	65. 8623	116. 0403	1099. 9936

从表 3-7 可以看出, MATA&ERU 在多数测试条件下都具有较好的多样性, 但是在二目标的测试实例上, IBACO 的多样性更为显著, 造成这种情况的可能原因是在目标维度较低时, 采用 epsilon 指标选择帕累托解能够保持较好的多样性, 然而当目标维度较高时, 这种方式对维持解的多样性是不足的, 这也从另一方面反映出 MATA&ERU 使用的多样性度量指标是有效的。此外从 SP 指标可以看出, 在测试实例上, NSGA II 和 MOEA/D 同样不具备理想的多样性。

现对 MATA&ERU 算法作性能的综合比较分析, 其 IGD 指标测试结果如表 3-8 所列。

表 3-8 对比算法 IGD 均值

测试实例	MATA&ERU	IBACO	MCWSO	<i>m</i> ACO-4	NSGA II	MOEA/D
100-2	<b>807. 8505</b>	835. 3761	811. 3835	819. 2451	813. 7452	937. 0332
250-2	<b>547. 2077</b>	582. 7111	605. 2128	589. 5517	893. 4603	1248. 9469
500-2	<b>729. 1222</b>	833. 3596	745. 7748	861. 2893	2775. 8448	1695. 6052
750-2	<b>1620. 0748</b>	1739. 7270	1796. 4048	1801. 8531	2713. 9296	3219. 9071
100-3	521. 7582	526. 8646	535. 1890	531. 5882	<b>501. 2955</b>	656. 3155
250-3	<b>684. 1746</b>	743. 1379	690. 6468	738. 3040	775. 0745	1073. 1932
500-3	<b>1212. 7188</b>	1367. 4735	1232. 1343	1375. 07785	2214. 8503	2296. 9213
750-3	<b>1533. 2902</b>	1816. 7118	1572. 5043	1821. 2348	2914. 8789	3223. 9616
100-4	<b>882. 1408</b>	891. 1976	942. 7249	889. 9326	897. 8372	990. 2010
250-4	<b>762. 5629</b>	807. 5082	837. 6126	831. 3502	917. 3267	1084. 9778
500-4	<b>1424. 1599</b>	1580. 6191	1435. 3153	1617. 8696	1911. 0176	2483. 0695
750-4	<b>2052. 2790</b>	2259. 4768	2107. 6578	2281. 6926	2793. 8403	3649. 1339

从表 3-8 中可以看出, 除了在 100-3 测试实例中, NSGA II 表现更为优异, 在多数测试条件下, MATA&ERU 都具有较好的综合性能, 这与采用 *C* 指标和 SP 指标分析得出的结论基本一致。此外, 还可以发现, 在同一个测试算法中, 当目标个数相同时, IGD 值随着背包维度的增加而增加, 同样地, 在背包维度相同时, IGD 值随着目标个数的增加而增加。这说明, 背包维度和目标个数的增加都会提高测试实例复杂性, 从而造成算法性能的下降。

最后, 验证等效路径信息素更新策略的有效性。由于等效路径信息素更新策略解决了

蚂蚁构建解的有序性和问题解的无序性之间的矛盾,即提高了算法的收敛性能,因此可以通过  $C$  指标进行度量。保持 MATA&ERU 的参数设置不变,仅对包含等效路径信息素更新策略和不包含该策略的 MATA&ERU 进行比较,设包含该策略的算法为 1,不包含该策略的算法为 2,算法仍然独立运行 10 次,它们的  $C$  指标结果如表 3-9 所列。

表 3-9 等效路径信息素更新策略  $C$  指标比较

测试实例	$C(1,2)$	$C(2,1)$
100-2	<b>0.5212</b>	0.3267
250-2	<b>0.7938</b>	0.3058
500-2	<b>0.7589</b>	0.4371
750-2	0.4578	<b>0.5661</b>
100-3	0.6491	<b>0.6984</b>
250-3	<b>0.7620</b>	0.5014
500-3	<b>0.7544</b>	0.4937
750-3	<b>0.8029</b>	0.3874
100-4	<b>0.6464</b>	0.4559
250-4	0.6505	<b>0.6506</b>
500-4	<b>0.8330</b>	0.4906
750-4	0.7079	<b>0.7998</b>

从表 3-9 中可以看出,采用等效路径信息素更新策略的算法在多数测试条件下都具有较好的收敛性能,尽管在 750-2、100-3、250-4 和 750-4 等测试实例中,算法 1 的效果弱于算法 2,但是差距并不显著。综上所述,等效路径信息素更新策略在解决蚂蚁构建解的有序性和多目标子集问题解的无序性方面是有效的。

## 本章小结

为了解决当前多目标蚁群算法在高维数据特征选择方面的不足,提出基于等效路径更新的两档案多目标蚁群优化 MATA&ERU。基于图的蚂蚁系统将问题转化成带权图,信息素放在带权图的边上,提出使用等效路径信息素增强策略,将问题的无序信息转换为等效路径上的信息素,消除了解的无序性与蚂蚁求解有序性之间的矛盾;将原目标转换为收敛性目标和多样性目标并设置对应的帕累托档案和信息素矩阵,采用 epsilon 指标函数更新收敛性档案,提出结合目标空间帕累托支配关系与源空间的距离信息的新指标度量解的多样性,并更新多样性档案,提高了多目标蚁群算法的可扩展性。以 2~4 个目标的标准多目标背包问题为例进行测试,验证了等效路径信息素增强策略的有效性以及所提算法在收敛性以及多样性上的优越性。本章为第 4~6 章的高维数据实体分辨的进一步研究提供算法准备。

## 本章参考文献

- [1] Li B D, Li J L, Tang K, et al. Many-Objective Evolutionary Algorithms: a Survey [J]. *ACM Computing Surveys*, 2015, 48(1): 1-35.
- [2] Mukhopadhyay A, Maulik U, Bandyopadhyay S, et al. Survey of Multiobjective Evolutionary Algorithms for Data Mining: Part II [J]. *IEEE Transactions on Evolutionary Computation*, 2014, 18(1): 20-35.
- [3] Mukhopadhyay A, Maulik U, Bandyopadhyay S, et al. A Survey of Multiobjective Evolutionary Algorithms for Data Mining: Part I [J]. *IEEE Transactions on Evolutionary Computation*, 2014, 18(1): 4-19.
- [4] Wang H D, Jiao L C, Yao X. Two\_Arch2: An Improved Two-Archive Algorithm for Many-Objective Optimization [J]. *IEEE Transactions on Evolutionary Computation*, 2015, 19(4): 524-541.
- [5] 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]. *IEEE Transactions on Evolutionary Computation*, 2014, 18(4): 577-601.
- [6] Zhang Q, Li H. MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition [J]. *IEEE Transactions on Evolutionary Computation*, 2007, 11(6): 712-731.
- [7] Lopez-Ibanez M, Stutzle T. The Automatic Design of Multiobjective Ant Colony Optimization Algorithms [J]. *IEEE Transactions on Evolutionary Computation*, 2012, 16(6): 861-875.
- [8] Wang Z T, Guo J S, Zheng M F, et al. Uncertain Multiobjective Traveling Salesman Problem [J]. *European Journal of Operational Research*, 2015, 241(2): 478-489.
- [9] Ariyasingha I D I D, Fernando T G I. Performance Analysis of the Multi-Objective Ant Colony Optimization Algorithms for the Traveling Salesman Problem [J]. *Swarm and Evolutionary Computation*, 2015, 23(Supplement C): 11-26.
- [10] Phonrattanasak P, Leeprechanon N. Multiobjective Ant Colony Optimization for Fast Charging Stations Planning in Residential Area [C]. *Proceedings of the 2014 IEEE Innovative Smart Grid Technologies - Asia*, Kuala Lumpur, Malaysia, May 20-23, 2014: 290-295.
- [11] Malekloo M, Kara N. Multi-objective ACO Virtual Machine Placement in Cloud Computing Environments [C]. *Proceedings of the 2014 IEEE Globecom Workshops*, Austin, TX, USA, Dec. 8-12, 2014: 112-116.
- [12] Dilip K, Kumar T V V. Multi-criteria Website Optimization Using Multi-Objective ACO [C]. *Proceedings of the 4th International Conference on Reliability, Infocom Technologies and Optimization*, Noida, India, Sept. 2-4, 2015: 1-6.
- [13] Zitzler E, Thiele L, Laumanns M, et al. Performance Assessment of Multiobjective Optimizers: An Analysis and Review [J]. *IEEE Transactions on Evolutionary Computation*, 2003, 7(2): 117-132.
- [14] Ishibuchi H, Akedo N, Nojima Y. Behavior of Multiobjective Evolutionary Algorithms on Many-Objective Knapsack Problems [J]. *IEEE Transactions on Evolutionary Computation*, 2015, 19(2): 264-283.
- [15] Lopez-Ibanez M, Stutzle T. The Impact of Design Choices of Multiobjective Antcolony Optimization Algorithms on Performance: an Experimental Study on the Biobjective TSP [C]. *Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation*, Portland, Oregon, USA, Jul. 7-11, 2010: 71-78.
- [16] Gutjahr W J. A Graph-Based Ant System and Its Convergence [J]. *Future Generation Computer*

- Systems, 2000, 16(9): 873-888.
- [17] 曹建军, 张培林, 王艳霞, 等. 一种求解子集问题的基于图的蚂蚁系统[J]. 系统仿真学报, 2008, 20(22): 6146-6150.
- [18] Asafuddoula M, Ray T, Sarker R. A Decomposition-Based Evolutionary Algorithm for Many Objective Optimization[J]. IEEE Transactions on Evolutionary Computation, 2015, 19(3): 445-460.
- [19] Wang G G, Gandomi A H, Alavi A H, et al. A Hybrid Method Based on Krill Herd and Quantum-Behaved Particle Swarm Optimization [J]. Neural Computing and Applications, 2016, 27(4): 989-1006.
- [20] Gandomi A H, Yang X S, Alavi A H, et al. Bat Algorithm for Constrained Optimization Tasks[J]. Neural Computing and Applications, 2013, 22(6): 1239-1255.
- [21] Wang G G, Gandomi A H, Zhao X, et al. Hybridizing Harmony Search Algorithm with Cuckoo Search for Global Numerical Optimization[J]. Soft Computing, 2016, 20(1): 273-285.
- [22] Richard O D, Perer E H. Pattern Classification and Scene Analysis[M]. New York: John Willey and Sons, 2001: 131-132.
- [23] Ke L, Zhang Q, Battiti R. MOEA/D-ACO: a Multiobjective Evolutionary Algorithm Using Decomposition and AntColony[J]. IEEE Transactions on Cybernetics, 2013, 43(6): 1845-1859.
- [24] Jiang S W, Zhang J, Ong Y S, et al. A Simple and Fast Hypervolume Indicator-Based Multiobjective Evolutionary Algorithm[J]. IEEE Transactions on Cybernetics, 2015, 45(10): 2202-2213.
- [25] Bringmann K, Friedrich T, Klitzke P. Two-Dimensional Subset Selection for Hypervolume and Epsilon-Indicator[M]. New York: ACM, 2014: 589-596.
- [26] Bader J, Zitzler E. HypE: an Algorithm for Fast Hypervolume-Based Many-Objective Optimization [J]. Evolutionary Computation, 2011, 19(1): 45-76.
- [27] Deb K, Pratap A, Agarwal S, et al. A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II [J]. IEEE Transactions on Evolutionary Computation, 2002, 6(2): 182-197.