

一种基于差分进化和编码修正的重叠联盟结构生成算法

桂海霞^{1,2}, 张国富^{1,3†}, 苏兆品^{1,3}, 蒋建国^{1,3}

(1. 合肥工业大学 计算机与信息学院, 安徽 合肥 230009; 2. 安徽理工大学 经济与管理学院, 安徽 淮南 232001;
3. 合肥工业大学 安全关键工业测控技术教育部工程研究中心, 安徽 合肥 230009)

摘要: 重叠联盟结构生成是人工智能和多智能体系统领域中的一个难点问题. 在重叠联盟中, 一个智能体可以同时参与多个不同的任务求解联盟. 然而, 由于每个智能体拥有的资源有限, 不同任务求解联盟之间的资源竞争会带来潜在的资源冲突. 针对这一问题, 本文将传统的差分进化扩充到三维整数编码, 编码中的每一个元素代表某智能体在某种资源上对某任务的实际贡献量. 此外, 设计了相应的编码修正策略以评估和解决编码中可能存在的资源冲突. 最后, 与相关已有方法进行了对比实验分析. 实验结果表明, 本文所提算法可以在较短的编码修正时间内获得更好的联盟结构值.

关键词: 多智能体系统; 重叠联盟; 资源冲突; 差分进化; 编码修正

引用格式: 桂海霞, 张国富, 苏兆品, 等. 一种基于差分进化和编码修正的重叠联盟结构生成算法. 控制理论与应用, 2018, 35(2): 215 – 223

中图分类号: TP18 文献标识码: A

An algorithm for overlapping coalition structure generation based on differential evolution and encoding repairs

GUI Hai-xia^{1,2}, ZHANG Guo-fu^{1,3†}, SU Zhao-pin^{1,3}, JIANG Jian-guo^{1,3}

(1. School of Computer and Information, Hefei University of Technology, Hefei Anhui 230009, China;
2. School of Economics and Management, Anhui University of Science and Technology, Huainan Anhui 232001, China;
3. Engineering Research Center of Safety Critical Industrial Measurement and Control Technology, Ministry of Education, Hefei University of Technology, Hefei Anhui 230009, China)

Abstract: Overlapping coalition structure generation (OCSG) is a hard topic in the fields of artificial intelligence and multi-agent systems. In overlapping coalitions, an agent may join in different task-oriented coalitions at the same time. Unfortunately, each agent's resources are limited, and thus there may exist potential resource conflicts over the scarce resources among different cooperative coalitions. To tackle such situations, first of all, this paper improves the traditional differential evolution with 3D integer encoding, in which each element represents the actual contribution amount of one type of resource of an agent to a task. Next, the corresponding encoding-repair strategy is presented to check and resolve the possible resource conflicts in each encoding. Finally, we compare the proposed algorithm with the existing approaches to OCSG. The empirical results show that the proposed algorithm can obtain a better coalition structure value within a shorter repair time.

Key words: multi-agent systems; overlapping coalitions; resource conflicts; differential evolution; encoding repairs

Citation: GUI Haixia, ZHANG Guofu, SU Zhaopin, et al. An algorithm for overlapping coalition structure generation based on differential evolution and encoding repairs. *Control Theory & Applications*, 2018, 35(2): 215 – 223

1 引言(Introduction)

近年来, 基于多智能体系统(multi-agent systems, MAS)的分布式智能控制正在蓬勃兴起, 使得智能体间的协调与合作研究越来越受到重视^[1-2]. 在MAS中,

当某个资源有限的智能体遇到其自身无法完成的任务时, 它可以选择与系统中的其他智能体进行交互和协作, 组成一个团队来共同承担这个任务, 这样的团队即为联盟(coalition)^[3-4]. 联盟生成是MAS中一切活

收稿日期: 2016-12-29; 录用日期: 2017-11-08.

†通信作者. E-mail: zgf@hfut.edu.cn; Tel.: +86 551-62901669.

本文责任编辑: 方浩.

国家自然科学基金项目(61573125, 61371155), 安徽省自然科学基金项目(1608085MF132, 1508085MF132, 1508085QF129), 安徽省科技攻关计划项目(1301b042023), 中央高校基本科研业务费专项资金(JZ2017YYPY0232)资助.

Supported by the National Natural Science Foundation of China (61573125, 61371155), the Anhui Provincial Natural Science Foundation (1608085MF132, 1508085MF132, 1508085QF129), the Key Projects of Science and Technology of Anhui Province (1301b042023) and the Fundamental Research Funds for the Central Universities (JZ2017YYPY0232).

动的基础,能够提高单个智能体求解任务的效率,已成为人工智能和MAS领域的一个热点课题,并已广泛应用于电力传输^[5]、虚拟企业^[6-7]、无线网络^[8]、多机器人协作^[9]和多媒体安全^[10]等。

然而,现有大多数的联盟生成研究均局限于非重叠联盟结构^[11-12],即在任意时刻一个智能体只能加入一个联盟。换句话说,当一个联盟形成后,对于加入到此联盟的每个智能体,即使它有足够多的剩余资源也不能加入其他任务联盟,势必造成智能体资源的极大浪费。反言之,为了提高智能体资源的利用率和任务的求解效率,一个资源充沛的智能体完全可以同时参加多个不同的联盟,以获取最大的系统利益,这一联盟重叠性已在虚拟企业^[6,13]和无线网络等应用领域中得到有效验证^[14-15]。

相比于非重叠联盟理论的硕果累累^[11-12],重叠联盟形成研究起步较晚,目前还处于探索阶段。由于重叠联盟允许智能体自由参加任务求解联盟,其解空间要远比非重叠联盟庞大的多。因此,就目前的现状来说,重叠联盟形成仍有许多问题需要深入研究。

在重叠联盟数学建模和效用分配方面,Chalkiadakis等^[16-17]从对策论中引入核的概念来对重叠联盟进行建模和分析,但其模型要求每个智能体只能拥有单一资源。Zick等^[18-20]提出了一种有仲裁者参与的重叠联盟模型,并对其进行了计算复杂性分析。张国富等^[21]采用讨价还价策略对多任务进行串行分派,然后基于效用非减原则划分重叠联盟效用。随后,本文在先前的工作中^[22]基于能者多劳的思想采取按比例分配,对多个并发任务进行并行分派,并根据任务分派情况划分重叠联盟的效用。然而,上述已有工作并没有分析重叠联盟的解空间,也没有指出如何搜索重叠联盟结构。

在重叠联盟结构生成(overlapping coalition structure generation, OCSG)方面,Shehory和Kraus^[23]假设所有任务的执行是串行的,且每个智能体在完成任务后其资源不会损耗,并设计了一种启发式算法来搜索每个任务的求解联盟。但是,为了降低算法复杂度,其算法对联盟规模进行了严格限制,导致其算法很难得到最优解,甚至近似最优解。Zhan等^[24]基于动态规划和贪婪策略搜索重叠联盟,但其算法对多个任务仍然采用串行搜索,下一个任务的求解联盟只能在前一个任务的联盟形成后才能进行搜索,一旦中间某个任务的联盟无法形成,算法将不再搜索后续任务的求解联盟。而且,上述这种串行搜索方式必然会影响后续任务的联盟优化结果,因为某些智能体的资源已被前面的联盟占据。

为了重叠联盟结构的并行生成,Lin和Hu^[25]以及Zhang等^[26]将传统的离散粒子群算法^[27]扩充到二维二进制编码来搜索重叠联盟。但是其设计的编码修正策略过于复杂,需要耗费大量的计算开销,导致算法

效率极其低下。为此,张国富等^[28]提出将有效联盟的剩余能力转移给一个虚拟联盟,然后让这个虚拟联盟去帮助和解决其他无效联盟。这种启发式操作虽然大大降低了编码修正的计算成本,但随着虚拟联盟的规模越来越大,会导致后续无效联盟经过修正后通信成本增大,降低了一些联盟的优越性。针对该问题,杜继永等^[29]在连续粒子群算法^[30]中采用二阶段初始化,首先对每个粒子随机生成二维二进制编码,再根据二进制编码在智能体资源空间内随机生成二维整数编码,作为智能体在联盟中的资源贡献量。在编码修正时,如果有效联盟的剩余能力超过了某个成员的贡献量,则剔除该成员,对联盟进行裁减。否则的话,将联盟的剩余能力平摊到联盟成员上,最后利用智能体的剩余能力来解决其他无效联盟。这种启发式操作可以在一定程度上提高重叠联盟的优越性,但其修正策略过于繁琐,增加了编码修正的计算开销,而且,这种把联盟的剩余能力平摊到成员上的操作势必改变智能体持有的原始资源量,导致联盟解空间的动态变化,降低了算法的性能,所以其最后给出的解是不可靠的。

基于上述背景,本文将差分进化^[31-32]扩充到三维整数编码,编码中的每一个元素直观地代表某一智能体在某一种资源上对某一任务的实际贡献量,并提出相应的编码修正策略完成智能体资源的快速分配,同时有效避免潜在的资源冲突,试图以一种更加形象和有效的方式实现一个智能体可以同时参与多个不同的联盟。

2 问题描述(Problem formulations)

MAS中现有: $n \in \mathbb{N}$ 个智能体, $A = \{a_1, \dots, a_n\}$; $m \in \mathbb{N}$ 个待求解的任务, $T = \{t_1, \dots, t_m\}$ 。

对于 $\forall t_j \in T (j=1, \dots, m)$ 都有一个 $r \in \mathbb{N}$ 种资源需求向量, $\mathbf{D}_j = [d_1^j \dots d_r^j]$, 其中 $d_k^j (k=1, \dots, r)$ 为一个非负整数, 表示任务 t_j 在第 k 种资源上的需求量。

对于 $\forall a_i \in A (i=1, \dots, n)$ 具有 r 种初始资源, $\mathbf{B}_i = [b_1^i \dots b_r^i]$, 其中 b_k^i 为一个非负整数, 表示 a_i 拥有的第 k 种资源的量。

联盟 $C_j \subseteq A, C_j \neq \emptyset$, 为任务 t_j 对应的求解联盟。在重叠联盟中, 每个智能体可以同时参与多个不同的联盟并贡献自己的资源。为了清晰的刻画这一现象, 假设 $\forall a_i \in C_j$ 对于任务 t_j 所需要的第 k 种资源有一个实际贡献量 $w(i, j, k)$ 。显然, $w(i, j, k)$ 为一个非负整数, 且满足 $0 \leq w(i, j, k) \leq b_k^i$ 。注意, 如果 $a_i \notin C_j$, 则 $w(i, j, k) = 0$ 。此时, 联盟 C_j 也具有 r 种资源, $\mathbf{B}_{C_j} = [b_1^{C_j} \dots b_r^{C_j}]$ 为 C_j 中所有成员为了满足任务 t_j 所贡献的资源量总和, 也就是任务 t_j 的资源需求量, 即对 $\forall k \in \{1, \dots, r\}$, 有

$$b_k^{C_j} = \sum_{i=1}^n w(i, j, k) = d_k^j.$$

还需要注意的是, a_i 对所有任务的实际资源贡献之和不应该超过其资源初始总量, 否则就会产生资源冲突,

即要想避免资源冲突, 就必须满足对 $\forall k \in \{1, \dots, r\}$, 有

$$\sum_{j=1}^m w(i, j, k) \leq b_k^i.$$

另外, 对于 $\forall a_i \in A$ 具有 r 种剩余资源, $\mathbf{P}_i = [p_1^i \dots p_r^i]$, 其中: p_k^i 为一个非负整数, 满足 $0 \leq p_k^i \leq b_k^i$, 表示 a_i 参与某些联盟后剩余的 k 种资源量, 即对 $\forall k \in \{1, \dots, r\}$, 有

$$p_k^i = b_k^i - \sum_{j=1}^m w(i, j, k).$$

显然, 如果 a_i 没有参与任何联盟, 则 $\mathbf{P}_i = \mathbf{B}_i$.

与惯例一样^[28-29], C_j 的值由特征函数 $v(C_j) \geq 0$ 给出:

$$v(C_j) = \pi(t_j) - \theta(C_j) - \varepsilon(C_j), \quad (1)$$

其中: $\pi(t_j)$ 为完成任务 t_j 所获得的报酬; $\theta(C_j)$ 为 C_j 中所有成员贡献的资源所对应的成本; $\varepsilon(C_j)$ 为 C_j 中的各智能体成员协作求解任务 t_j 过程中的通信成本.

一般来说, 对于给定的 m 个任务 t_1, \dots, t_m , 最多需要 m 个求解联盟 C_1, \dots, C_m . 重叠联盟结构生成问题就是在给定的约束条件下, 尽可能的划分出 m 个任务求解联盟 C_1, \dots, C_m , 使得联盟结构值达到最大, 这是一个典型的 NP 完全问题^[21].

$$\max \sum_{j=1}^m v(C_j), \quad (2)$$

$$\sum_{i=1}^n b_k^i \geq \sum_{j=1}^m d_k^j, \quad k = 1, \dots, r. \quad (3)$$

3 OCSG 算法(Algorithm for OCSG)

3.1 差分进化(Differential evolution)

差分进化(differential evolution, DE)^[31-32]是一种模拟生物进化的随机搜索算法. DE 保留了基于种群的全局搜索策略, 采用实数编码, 并基于差分的简单变异操作和一对一的竞争生存策略, 通过反复迭代, 使得那些适应环境的个体被保存下来. 这种特有的记忆能力使得 DE 可以动态跟踪当前的搜索情况, 并及时调整其搜索策略, 因而 DE 具有较强的全局收敛能力和鲁棒性, 特别适合求解一些复杂优化问题^[31-32].

DE 算法主要包括变异(mutation)、交叉(crossover)和选择(selection)3 种操作. 变异操作主要利用从种群中随机选取的两个个体的差向量作为第 3 个个体的随机变化源, 将差向量加权后按照一定的规则与第 3 个个体求和而产生变异个体. 设 X_s^t (s 为当前个体在种群中的序号, t 为当前迭代数) 为当前种群的一个个体, $X_{s_1}^t, X_{s_2}^t$ 和 $X_{s_3}^t$ ($s_1 \neq s_2 \neq s_3 \neq s$) 是从当前种群中随机选取的另外 3 个不同个体, 则变异操作执行如下:

$$V_s^t = X_{s_1}^t + F \times (X_{s_2}^t - X_{s_3}^t), \quad (4)$$

其中 $F \in [0.4, 1]$ 为缩放因子, 用于控制差异向量的缩放程度. 交叉操作是将上述变异个体 V_s^t 与其对应的原

始目标个体 X_s^t 进行参数混合, 生成试验个体 U_s^t :

$$U_s^t = \begin{cases} V_s^t, & \text{rand}(0, 1) \leq Cr, \\ X_s^t, & \text{rand}(0, 1) > Cr, \end{cases} \quad (5)$$

其中: $\text{rand}(0, 1)$ 为 $(0, 1)$ 间均匀分布的随机数, $Cr \in [0, 1]$ 为交叉概率. 如果试验个体 U_s^t 的适应度值优于目标个体 X_s^t 的适应度值, 则在下一代中试验个体取代目标个体, 否则目标个体仍保存下来, 该操作即为选择:

$$X_s^{t+1} = \begin{cases} U_s^t, & f(U_s^t) \leq f(X_s^t), \\ X_s^t, & f(U_s^t) > f(X_s^t). \end{cases} \quad (6)$$

在每一代的进化过程中, 每一个体矢量作为目标个体一次, 算法通过上述的变异、交叉和选择 3 个操作不断地迭代计算, 保留优良个体, 淘汰劣质个体, 引导搜索过程向全局最优解逼近.

3.2 三维整数编码(3D integer encoding)

本文没有采用已有的二维二进制编码^[28]和二维二进制和整数混合编码^[29], 这是因为二维二进制编码只能表现哪些智能体参与了联盟, 而不能展示各成员到底贡献了多少资源, 二维二进制和整数混合编码虽然解决了上述问题, 但表达过于复杂和累赘. 因此, 如图 1 所示, 为了更加直观地表示哪些智能体参与了联盟和每个智能体在每种资源上对每个联盟的实际贡献量, 本文将 DE 扩充到三维整数编码, 编码的每一位即表示每个智能体在每个任务中每种资源的实际贡献量 $w(i, j, k)$. 其中: i 轴表示智能体, j 轴表示任务, k 轴表示资源.

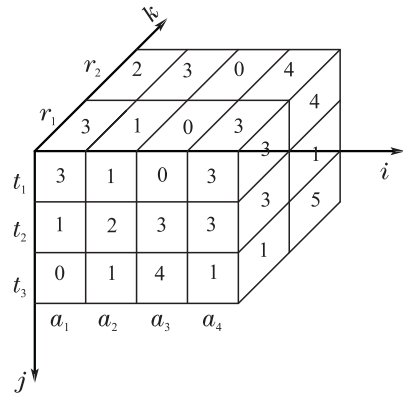


图 1 三维整数编码

Fig. 1 3D integer encoding

为了组成初始种群, 编码中的每一位根据给定的条件进行随机初始化, 对 $\forall i \in \{1, \dots, n\}, \forall j \in \{1, \dots, m\}$ 和 $\forall k \in \{1, \dots, r\}$, 执行

$$w(i, j, k) \leftarrow \text{rand}[0, \min(b_k^i, d_k^j)], \quad (7)$$

其中 $\text{rand}[0, \min(b_k^i, d_k^j)]$ 表示 $[0, \min(b_k^i, d_k^j)]$ 之间的随机数. 如果 $\exists k \in \{1, \dots, r\}$, 满足 $w(i, j, k) \neq 0$, 则表示 a_i 在任务 t_j 的求解联盟 C_j 中.

在上述编码中, 对于每个任务求解联盟来说, 只要在 k 轴的某一位上出现联盟中所有成员贡献之和小于

任务的需求,则该联盟即是一个无效联盟;即使该联盟是有效的,一旦联盟拥有的资源超过了任务的需求,势必造成智能体资源的浪费;而且,对于每个智能体来说,只要在 k 轴的某一位上出现智能体的资源贡献总量超过了其自身的拥有量,这时就会产生资源冲突.为此,我们需要对编码进行适当的修正,以确保每个任务求解联盟恰好可以满足任务需求,同时不会出现资源冲突.

3.3 编码修正(Encoding repairs)

为了有效避免资源冲突,当一个智能体被选中加入一个联盟,它能向该联盟贡献的资源量不应该大于其当前的资源拥有量.同时,我们需要计算该智能体在这个联盟中的实际贡献量以及它的剩余资源量,以便该智能体通过剩余资源来参与其他联盟.一旦该智能体没有任何剩余资源,它将不再响应其他任务的请求.基于上述思想,编码修正策略主要步骤描述如下:

步骤1 初始化: $\forall i \in \{1, \dots, n\}, P_i \leftarrow B_i; \forall j \in \{1, \dots, m\}, C_j \leftarrow \emptyset, B_{C_j} \leftarrow 0$.

步骤2 在 j 轴上随机选择一个未曾检查过(即没有评估过的)的基因位(即联盟 C_j).依次遍历 i 轴和 k 轴上的每一位,如果 $d_k^j = 0$,则任一 $w(i, j, k) \leftarrow 0$.如果 $d_k^j > 0$ 且 $w(i, j, k) > 0$,则将 a_i 加入联盟 C_j ,同时更新 B_{C_j} 和 P_i 的值,即

$$\begin{cases} C_j \leftarrow C_j \cup \{a_i\}, \\ b_k^{C_j} \leftarrow b_k^{C_j} + w(i, j, k), \\ p_k^i \leftarrow p_k^i - w(i, j, k). \end{cases} \quad (8)$$

步骤3 再次遍历 k 轴上的每一位:

1) 如果 $b_k^{C_j} < d_k^j$,则联盟 C_j 不能满足 t_j 在第 k 种资源上的需求.此时,依次选择 C_j 成员中最大的 $p_k^{i^*}$,执行

$$\begin{cases} \bar{w}(i^*, j, k) \leftarrow \min(p_k^{i^*}, d_k^j - b_k^{C_j}) + w(i^*, j, k), \\ p_k^{i^*} \leftarrow p_k^{i^*} - \min(p_k^{i^*}, d_k^j - b_k^{C_j}), \\ b_k^{C_j} \leftarrow b_k^{C_j} + \min(p_k^{i^*}, d_k^j - b_k^{C_j}), \end{cases} \quad (9)$$

直到满足 $b_k^{C_j} = d_k^j$ 或者没有可用的 $p_k^{i^*}$.如果 C_j 中所有成员遍历完仍然存在 $b_k^{C_j} < d_k^j$,则在 i 轴上遍历所有的 $w(i, j, k) = 0$ 但 $p_k^i > 0$ 的基因位,依次从中选择最大的 p_k^i ,执行

$$\begin{cases} \bar{w}(i, j, k) \leftarrow \min(p_k^i, d_k^j - b_k^{C_j}), \\ p_k^i \leftarrow p_k^i - \bar{w}(i, j, k), \\ b_k^{C_j} \leftarrow b_k^{C_j} + \bar{w}(i, j, k), \\ C_j \leftarrow C_j \cup \{a_i\}, \end{cases} \quad (10)$$

直到满足 $b_k^{C_j} = d_k^j$.

2) 如果 $b_k^{C_j} > d_k^j$,则联盟 C_j 能够满足 t_j 在第 k 种资源上的需求,但存在资源浪费.此时,依次选取 C_j 成员

中最小的 $w(i^*, j, k) > 0$,执行

$$\begin{cases} \bar{w}(i^*, j, k) \leftarrow \max[0, d_k^j + w(i^*, j, k) - b_k^{C_j}], \\ p_k^{i^*} \leftarrow p_k^{i^*} + \min[b_k^{C_j} - d_k^j, w(i^*, j, k)], \\ b_k^{C_j} \leftarrow b_k^{C_j} - \min[b_k^{C_j} - d_k^j, w(i^*, j, k)], \end{cases} \quad (11)$$

直到满足 $b_k^{C_j} = d_k^j$.

步骤4 依次遍历 i 轴和 k 轴上的每一位,如果 $\sum_{k=1}^r w(i, j, k) = 0$ 且 $a_i \in C_j$,则让 a_i 退出联盟 C_j ,即 $C_j \leftarrow C_j - \{a_i\}$.

步骤5 如果 j 轴上所有的基因位都已遍历过,则退出编码修正,否则转步骤2.

需要指出的是:在步骤2中,首先遍历和统计所有 $w(i, j, k) > 0$ 的基因位,这是为了尽量保留编码原有的进化趋势和特征,确保算法的收敛;在步骤3中,当联盟 C_j 不能满足 t_j 在第 k 种资源上的需求时,首先依次评估 C_j 成员中最大的可用 $p_k^{i^*}$,如果 C_j 仍然无效,再依次挑选 C_j 外最大的可用 p_k^i ,这样操作的目的是让资源充沛的智能体尽可能的贡献资源,从而尽量减少其他智能体加入联盟的概率,限制联盟规模的增加,控制联盟的通信成本,防止联盟值快速下降,同时又能在一定程度上减少修正次数,节省计算开销;当联盟 C_j 的资源拥有量超过 t_j 在第 k 种资源上的需求时,依次评估 C_j 成员中最小的非零 $w(i^*, j, k)$,这是因为最小的 $w(i^*, j, k)$ 退出任务的概率最大,从而可以在一定程度上促使智能体退出联盟,减小联盟规模和通信成本,继而增加联盟值;而且,在步骤2和步骤3中所有基因位的选取均是采用随机抽样的方式,以保证种群的多样性.

此外,上述编码修正策略还具有如下性质:

定理1 编码修正策略的时间复杂度至多为 $O(m \times n \times r)$.

证 在上述修正策略中,首先,很容易看出步骤2的时间复杂度至多为 $O(m \times n \times r)$.其次,在步骤3中,每个联盟上的每一维资源都需要检查,如果 $b_k^{C_j} < d_k^j$,最恶劣的情形是需要把 C_j 外所有的智能体拉入才能满足 t_j 的需求,即至多需要遍历 n 个智能体,因此,其时间复杂度至多为 $O(m \times n \times r)$.如果 $b_k^{C_j} > d_k^j$,最恶劣的情形是 $C_j = A$ 且修正后只需保留一个智能体仍然能满足 t_j 的需求,即至多需要遍历 $n - 1$ 个智能体,所以其时间复杂度也是 $O(m \times n \times r)$.最后,步骤4需要判断智能体在所有联盟中的实际贡献量,其时间复杂度至多为 $O(m \times n \times r)$.综上,编码修正策略的时间复杂度至多为 $O(m \times n \times r)$.

定理2 在满足式(3)的前提下,任意一个三维整数编码都能被本文的修正策略修正为一个合法编码.

证 在本文修正策略中, 根据步骤3的计算方法可以保证修正后的 C_{j^*} 恰好可以满足 t_{j^*} 的需求, 即 $B_{C_{j^*}} = D_{j^*}$. 因此, 在修正下一个联盟时, 满足

$$\sum_{i=1}^n b_k^i - B_{C_{j^*}} \geq \sum_{j=1}^m d_k^j - D_{j^*}, k = 1, \dots, r.$$

由上述条件可知, 系统中智能体的剩余资源总量仍然能够满足其他任务的需求, 从而可以确保下一个联盟也能被修正为一个有效联盟, 依此类推, 所有联盟都能被修正为有效而不会有资源冲突, 因此, 本文修正策略在式(3)条件下可以确保每个编码都能修正为一个合法编码.

3.4 算法描述(Algorithm description)

基于差分进化和编码修正的OCSG算法如图2所示, 具体描述如下: 1) 按照第4.2节的方法随机产生第1代种群; 2) 按照第4.3节的方法对初始种群进行编码修正, 并根据式(2)计算每个个体的适应度值; 3) 对修正过的初始种群进行变异和交叉操作产生试验种群; 4) 按照第4.3节的方法对试验种群进行编码修正, 并根据式(2)计算每个个体的适应度值; 5) 根据适应度值对试验种群和初始种群进行选择, 生成下一代初始种群; 6) 如果已经达到最大迭代次数, 则结束算法并输出结果, 否则转3).

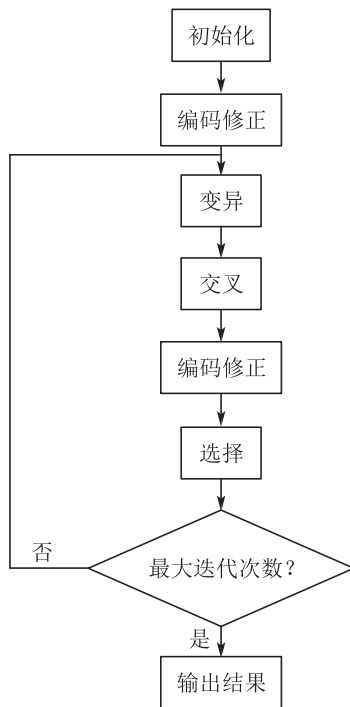


图 2 OCSG算法流程图

Fig. 2 Flowchart of algorithm for OCSG

在上述算法中, 在变异和交叉前必须对初始种群进行编码修正, 以确保每个个体都是有效的, 因为无效个体的进化会大大降低算法的效率^[28]; 同样, 对产生的试验种群在选择前也必须进行编码修正, 以保证选择操作的有效性.

4 对比实验与分析(Comparative experiment and analysis)

本文工作可以看作是对最近的文献[28]和文献[29]所提工作的改进, 因此, 为了验证本文算法的性能, 本节将本文所提算法与文献[28]和文献[29]的算法进行对比实验. 本文算法参数为: $F = 0.6$, $Cr = 0.9$. 为了对比的公平性, 本文沿用了上述两个文献的算法参数和实验环境: 在实验环境1中, 所有智能体拥有的资源总量恰好等于所有任务的资源需求总量, 即 $\sum_{i=1}^n b_k^i = \sum_{j=1}^m d_k^j$, $k = 1, \dots, r$; 在实验环境2中, 所有智能体拥有的资源总量大于所有任务的资源需求总量, 即

$$\sum_{i=1}^n b_k^i > \sum_{j=1}^m d_k^j, k = 1, \dots, r.$$

在每种实验环境中, 各测试案例均根据预先设定的阈值随机生成, 并在AMD A8 CPU、内存4 GB、操作系统Windows 7的个人计算机上独立运行30次.

4.1 智能体个数的影响(Impact of n)

为了检查3种算法对智能体个数的敏感性, 在第1个实验中, $m = 4$, $r = 2$, n 从16依次增加到30.

表1给出了3种算法分别在两种实验环境中不同 n 时的联盟结构值. 为了进行统计分析, 笔者给出了联盟结构值的均值和标准差, 并基于Wilcoxon Rank Sum检验(0.05的显著性水平)和Bonferroni校正^[33]进行显著性分析, 用黑体标出差异显著的最佳均值. 总的来说, 本文算法在两种环境中得到的联盟结构值均要明显优于文献[28–29]的算法. 在实验环境1中, 随着 n 的增加, 3种算法的联盟结构值呈快速下降趋势. 这是因为在该环境中所有智能体都必须参与任务才能满足任务的需求, 随着智能体数的增加, 每个任务的求解联盟规模越来越大, 导致通信成本越来越高. 值得注意的是, 当 $n \leq 20$ 时, 文献[29]的算法得到的系统总收益要好于文献[28]的算法, 但当 $n \geq 22$ 时, 文献[29]的算法性能下降较快. 正如前面所分析的那样, 文献[29]将有效联盟的剩余资源转移给联盟成员, 改变了智能体的初始资源向量, 导致解空间发生变化, 而环境的动态变化会降低优化算法的性能^[34], 因此, 文献[29]的算法很不稳定. 在实验环境2中, 随着智能体数的增加, 系统资源越来越充沛, 本文算法和文献[28]算法所得的联盟结构值相对比较稳定, 而文献[29]算法会改变解空间, 所得联盟结构值缓慢下降.

不同的演化算法具有不同的进化特征和操作, 因此直接对比本文算法和文献[28–29]算法的运行时间有失公允, 不能正确反应各算法的效率, 为此, 表2给出了3种算法分别在两种实验环境中不同 n 时的平均编码修正时间及其95%置信区(s). 总体来看, 3种算法

在两种实验环境中的编码修正时间都会随着 n 的增加而增多,但文献[29]算法明显增加的更快.而且,本文算法和文献[28]算法耗费时间较为接近但明显快于文

献[29]算法.可见,文献[29]算法对 n 更加敏感,这是因为文献[29]算法的编码修正包含很多冗余的裁减操作来判断联盟中哪些成员可以剔除出联盟.

表 1 3种算法在不同 n 时的联盟结构值(均值±标准差)

Table 1 Coalition structure values (mean and standard deviation) of three algorithms with different n

n	实验环境1			实验环境2		
	本文算法	文献[28]算法	文献[29]算法	本文算法	文献[28]算法	文献[29]算法
16	1026±27	857±12	966±26	1265±4	1253±5	1193±11
18	984±29	832±9	916±29	1266±4	1253±5	1192±10
20	932±30	812±6	858±33	1267±4	1255±5	1183±9
22	860±23	790±7	742±18	1265±3	1255±3	1183±14
24	789±35	749±7	658±32	1271±5	1257±5	1170±13
26	751±39	698±15	554±30	1269±4	1256±4	1165±13
28	684±44	638±11	474±35	1268±3	1257±4	1161±9
30	649±36	561±12	372±35	1270±5	1258±6	1153±10

表 2 3种算法在不同 n 时的编码修正时间(均值±95%置信区间,秒)

Table 2 Encoding repair time (mean and 95% confidence interval of the mean, in seconds) of three algorithms with different n

n	实验环境1			实验环境2		
	本文算法	文献[28]算法	文献[29]算法	本文算法	文献[28]算法	文献[29]算法
16	0.131±0.004	0.137±0.005	0.584±0.015	0.122±0.005	0.135±0.005	0.543±0.015
18	0.149±0.006	0.153±0.007	0.716±0.022	0.138±0.008	0.146±0.010	0.638±0.020
20	0.165±0.008	0.172±0.008	0.808±0.017	0.154±0.009	0.167±0.011	0.731±0.023
22	0.177±0.008	0.184±0.007	0.907±0.025	0.167±0.010	0.176±0.009	0.805±0.020
24	0.189±0.006	0.201±0.009	0.971±0.017	0.181±0.012	0.191±0.014	0.888±0.023
26	0.211±0.011	0.223±0.012	1.117±0.019	0.201±0.013	0.205±0.013	0.954±0.025
28	0.225±0.008	0.230±0.008	1.174±0.023	0.208±0.008	0.218±0.009	1.047±0.022
30	0.241±0.008	0.249±0.006	1.361±0.024	0.221±0.008	0.243±0.009	1.081±0.024

4.2 任务个数的影响(Impact of m)

在第2个实验中,测试任务个数的变化对3种算法的性能影响.其中: $n = 10$, $r = 2$, m 从6增加到20.表3为3种算法在不同 m 时分别在两种实验环境下的联盟结构值(均值和标准差).

表 3 3种算法在不同 m 时的联盟结构值(均值±标准差)

Table 3 Coalition structure values (mean and standard deviation) of three algorithms with different m

m	实验环境1			实验环境2		
	本文算法	文献[28]算法	文献[29]算法	本文算法	文献[28]算法	文献[29]算法
6	1943±8	1527±158	1899±15	1992±8	1836±152	1974±11
8	2593±8	1716±3	2524±17	2645±5	2596±134	2617±18
10	3345±8	2324±159	3253±14	3387±4	3352±134	3364±16
12	3984±6	2760±104	3868±15	4030±5	3767±25	3990±14
14	4670±7	2800±108	4553±17	4717±8	4214±20	4700±24
16	5323±8	3406±154	5185±13	5358±6	4712±32	5347±18
18	6220±6	3637±103	6076±16	6258±5	5131±34	6244±32
20	6978±8	3675±34	6813±17	7009±6	5554±24	6991±23

如表3所示, 随着 m 的增加, 3种算法得到的联盟结构值呈快速上升趋势, 这是因为任务的增加带来了更多的报酬, 而智能体个数不变, 所以总的通信成本有限. 在实验环境1中, 本文算法与文献[29]算法差距不大, 但明显好于文献[28]算法. 这是因为随着 m 的增加, 文献[28]算法中虚拟联盟的规模越来越大, 导致后续联盟的通信成本越来越高, 影响了联盟的优越性, 而本文算法和文献[29]算法都可以有效裁减联盟规模, 控制通信成本的增加. 在实验环境2中, 当 $m \leq 10$ 时, 3种算法的联盟结构值很接近, 当 $m \geq 12$ 时, 文献[28]算法的结果越来越不如本文算法和文献[29]算法. 这是因为智能体个数不变, 系统总资源始终保持不变, m 较小时, 系统资源

相对比较充分, 文献[28]算法中虚拟联盟被使用的概率较小, 随着 m 的增加, 系统资源越来越紧张, 需要引入虚拟联盟才能满足后续任务的需求, 导致后续联盟的收益较小.

表4为3种算法在不同 m 时分别在两种实验环境下的编码修正时间(均值 $\pm 95\%$ 置信区间, s). 随着 m 的增加, 3种算法的平均编码修正时间都会增加, 但无论是在实验环境1还是在实验环境2, 本文算法和文献[28]算法差距不大, 且明显快于文献[29]算法. 因此, 文献[29]算法对 m 比较敏感. 这是因为文献[29]算法需要对每个任务的求解联盟进行反复裁减, m 越大, 用于裁减操作的计算开销越大.

表 4 3种算法在不同 m 时的编码修正时间(均值 $\pm 95\%$ 置信区间, 秒)

Table 4 Encoding repair time (mean and 95% confidence interval of the mean, in seconds) of three algorithms with different m

m	实验环境1			实验环境2		
	本文算法	文献[28]算法	文献[29]算法	本文算法	文献[28]算法	文献[29]算法
6	0.121 \pm 0.008	0.135 \pm 0.009	0.685 \pm 0.016	0.116 \pm 0.006	0.132 \pm 0.008	0.619 \pm 0.011
8	0.164 \pm 0.009	0.178 \pm 0.009	1.012 \pm 0.022	0.149 \pm 0.008	0.166 \pm 0.008	0.940 \pm 0.022
10	0.209 \pm 0.011	0.236 \pm 0.011	1.406 \pm 0.021	0.174 \pm 0.010	0.209 \pm 0.009	1.310 \pm 0.009
12	0.232 \pm 0.015	0.269 \pm 0.014	1.839 \pm 0.036	0.209 \pm 0.012	0.246 \pm 0.013	1.727 \pm 0.016
14	0.253 \pm 0.014	0.323 \pm 0.014	2.386 \pm 0.064	0.239 \pm 0.014	0.303 \pm 0.014	2.199 \pm 0.008
16	0.275 \pm 0.015	0.360 \pm 0.015	2.821 \pm 0.027	0.271 \pm 0.015	0.351 \pm 0.015	2.703 \pm 0.032
18	0.301 \pm 0.009	0.413 \pm 0.008	3.482 \pm 0.037	0.296 \pm 0.011	0.371 \pm 0.011	3.134 \pm 0.719
20	0.338 \pm 0.018	0.448 \pm 0.018	4.079 \pm 0.132	0.308 \pm 0.007	0.414 \pm 0.008	3.906 \pm 0.037

4.3 资源种数的影响(Impact of r)

在最后的实验中, 测试资源种数的变化对3种算法的性能影响. 其中: $n = 10, m = 4, r$ 从2增加到12.

表5给出了3种算法在不同 r 时分别在两种实验环境下的联盟结构值(均值 \pm 标准差). 如表5所示, 本文算法得到的联盟结构值要略好于文献[28]和文献[29]算法, 但随着 r 的增加, 3种算法在两种环境下

的联盟结构值均呈显著下降趋势. 这是因为随着 r 的增加, 资源成本不断加大, 而任务报酬不变, 智能体间总的通信成本也有限, 导致联盟收益越来越小, 最后趋于0. 此外, 可以看出, 3种算法得到的联盟结构值在实验环境1中的下降趋势明显要比在实验环境2中快, 这是因为在实验环境1中所有智能体都必须参与任务, 其对应的联盟规模较大, 从而带来额外的通信成本.

表 5 3种算法在不同 r 时的联盟结构值(均值 \pm 标准差)

Table 5 Coalition structure values (mean and standard deviation) of three algorithms with different r

r	实验环境1			实验环境2		
	本文算法	文献[28]算法	文献[29]算法	本文算法	文献[28]算法	文献[29]算法
2	1181 \pm 18	925 \pm 18	1176 \pm 11	1232\pm9	1194 \pm 124	1223 \pm 5
4	831\pm25	647 \pm 119	775 \pm 15	939\pm20	728 \pm 1	871 \pm 21
6	557\pm29	468 \pm 23	391 \pm 23	680\pm19	505 \pm 2	498 \pm 22
8	293\pm37	75 \pm 114	43 \pm 32	421\pm30	282 \pm 2	154 \pm 16
10	33 \pm 26	0	0	164 \pm 22	59 \pm 2	0
12	0	0	0	0	0	0

表6给出了3种算法在不同 r 时分别在两种实验环境下的编码修正时间(均值 $\pm 95\%$ 置信区间, s). 由表6可见, 随着 r 的增加, 3种算法在两种环境下的平均编码修正时间都会增加, 且总体来看, 本文算法与文献[28]算法差距不大, 但要明显快于文献[29]算法. 因此, 文献[29]算法对 r 比较敏感. 这是因为文献[29]算法中的联盟裁减操作需要遍历每一维资源.

表 6 3种算法在不同 r 时的编码修正时间(均值 $\pm 95\%$ 置信区间, 秒)

Table 6 Encoding repair time (mean and 95% confidence interval of the mean, in seconds) of three algorithms with different r

r	实验环境1			实验环境2		
	本文算法	文献[28]算法	文献[29]算法	本文算法	文献[28]算法	文献[29]算法
2	0.084 \pm 0.008	0.086 \pm 0.012	0.367 \pm 0.009	0.083 \pm 0.009	0.085 \pm 0.007	0.352 \pm 0.014
4	0.159 \pm 0.006	0.163 \pm 0.006	0.664 \pm 0.014	0.155 \pm 0.006	0.160 \pm 0.008	0.618 \pm 0.008
6	0.249 \pm 0.011	0.257 \pm 0.009	0.938 \pm 0.012	0.244 \pm 0.011	0.254 \pm 0.009	0.892 \pm 0.007
8	0.306 \pm 0.009	0.316 \pm 0.009	1.184 \pm 0.011	0.298 \pm 0.012	0.308 \pm 0.010	1.128 \pm 0.029
10	0.399 \pm 0.009	0.415 \pm 0.009	1.452 \pm 0.007	0.382 \pm 0.007	0.410 \pm 0.006	1.386 \pm 0.039
12	0.471 \pm 0.007	0.491 \pm 0.007	1.914 \pm 0.023	0.453 \pm 0.013	0.482 \pm 0.011	1.832 \pm 0.041

综上, 文献[29]算法对 m, n, r 三个参数均比较敏感, 且所得联盟结构值不稳定. 因此, 文献[29]算法只适合于小样本下的联盟形成问题. 本文算法和文献[28]算法对 m, n, r 三个参数均不敏感, 但文献[28]算法在实验环境1中得到的联盟结构值明显不如本文算法, 说明文献[28]算法只适用于资源非常充分的情景, 而本文算法更加鲁棒, 在两种实验环境中均有较佳的表现.

5 结论(Conclusions)

为了更加有效的挖掘重叠联盟, 本文提出一种基于差分进化和编码修正的OCSG算法. 通过对相关工作的总结和分析, 基于差分进化设计了一种新的三维整数编码, 以便更加直观地表示每个智能体对每个任务在每种资源上的实际贡献量, 并设计了相应的编码修正策略, 以确保任何一个无效编码都能被快速修正为一个有效编码, 解决了多联盟竞争同一资源时可能存在的资源冲突问题. 通过与最近已有工作的对比实验和分析, 本文算法在两种实验环境中, 在编码修正耗时和联盟结构值上的表现均要优于已有算法. 但是, 本文还有如下不足需要在以后的工作中加以考虑: 首先, 和已有工作一样, 本文的修正策略必须在式(3)条件下才能发挥功效, 当智能体的资源总量不能满足所有任务的资源需求时, 如何对编码进行修正是一个值得深入研究的问题; 其次, 在未来的研究工作中, 笔者还需要对OCSG问题进行理论分析.

参考文献(References):

- [1] FANG Yi, XIONG Zhang, WANG Jiankun. Multi-agent system in intelligent control [J]. *Control Theory & Applications*, 2006, 23(5): 810 – 814.
(方义, 熊璋, 王剑昆. 智能控制中的多Agent系统 [J]. *控制理论与应用*, 2006, 23(5): 810 – 814.)
- [2] LUO Delin, ZHANG Haiyang, XIE Rongzeng, et al. Unmanned aerial vehicles swarm conflict based on multi-agent system [J]. *Control Theory & Applications*, 2015, 32(11): 1498 – 1504.
(罗德林, 张海洋, 谢荣增, 等. 基于多agent系统的大规模无人机集群对抗 [J]. *控制理论与应用*, 2015, 32(11): 1498 – 1504.)
- [3] JIANG Jianguo, SU Zhaopin, ZHANG Guofu, et al. Agent-behavior strategy in serial multi-task coalition formation [J]. *Control Theory & Applications*, 2008, 25(5): 853 – 856.
(蒋建国, 苏兆品, 张国富, 等. 多任务联盟形成中的Agent行为策略研究 [J]. *控制理论与应用*, 2008, 25(5): 853 – 856.)
- [4] SERVICE T C, ADAMS J A. Coalition formation for task allocation: theory and algorithms [J]. *Autonomous Agents and Multi-Agent Systems*, 2011, 22(2): 225 – 248.
- [5] LIU Bin, CHEN Laijun, WANG Yuchen, et al. Allocating reserve cost for hedging against wind generation uncertainty: a coalitional-game-theoretic approach [J]. *Control Theory & Applications*, 2016, 33(4): 437 – 445.
(刘斌, 陈来军, 汪雨辰, 等. 应对风电出力不确定性的备用成本分摊: 联盟博弈方法 [J]. *控制理论与应用*, 2016, 33(4): 437 – 445.)
- [6] ZHAO Qiang, XIAO Renbin. The multi-agent-based model for task scheduling in virtual enterprise and its optimization [J]. *Control Theory & Applications*, 2009, 26(4): 459 – 462.
(赵强, 肖人彬. 基于多Agent的虚拟企业任务调度模型及优化 [J]. *控制理论与应用*, 2009, 26(4): 459 – 462.)
- [7] ZHANG G, JIANG J, SU Z, et al. Searching for overlapping coalitions in multiple virtual organizations [J]. *Information Sciences*, 2010, 180(17): 3140 – 3156.
- [8] WU D, CAI Y, ZHOU L, et al. A cooperative communication scheme based on coalition formation game in clustered wireless sensor networks [J]. *IEEE Transactions on Wireless Communications*, 2012, 11(3): 1190 – 1200.
- [9] SEN S D, ADAMS J A. An influence diagram based multi-criteria decision making framework for multirobot coalition formation [J]. *Autonomous Agents and Multi-Agent Systems*, 2015, 29(6): 1061 – 1090.
- [10] ZHAO H V, LIN W S, LIU K J R. Cooperation and coalition in multimedia fingerprinting colluder social networks [J]. *IEEE Transactions on Multimedia*, 2012, 14(3): 717 – 733.

- [11] HU Shanli, SHI Chunyi, LI Shaofang. Cardinality structure grouping and coalition structure generation with given required bound [J]. *Chinese Journal of Computers*, 2012, 35(12): 2618 – 2624.
(胡山立, 石纯一, 李少芳. 给定限界的势结构分组与联盟结构生成 [J]. 计算机学报, 2012, 35(12): 2618 – 2624.)
- [12] RAHWAN T, MICHALAK T P, WOOLDRIDGE M, et al. Coalition structure generation: a survey [J]. *Artificial Intelligence*, 2015, 229: 139 – 174.
- [13] AFSARMANESH H, ANALIDE C. Virtual enterprises-methods and approaches for coalition formation [J]. *International Journal of Production Research*, 2009, 47(17): 4655 – 4659.
- [14] ZHANG Z, SONG L, HAN Z, et al. Coalitional games with overlapping coalitions for interference management in small cell networks [J]. *IEEE Transactions on Wireless Communications*, 2014, 13(5): 2659 – 2669.
- [15] XIAO Y, CHEN K C, YUEN C, et al. A Bayesian overlapping coalition formation game for device-to-device spectrum sharing in cellular networks [J]. *IEEE Transactions on Wireless Communications*, 2015, 14(7): 4034 – 4051.
- [16] CHALKIADAKIS G, ELKIND E, MARKAKIS E, et al. Overlapping coalition formation [C] // *Proceedings of the 4th International Workshop on Internet and Network Economics*. Shanghai: Springer, 2008, 12: 307 – 321.
- [17] CHALKIADAKIS G, ELKIND E, MARKAKIS E, et al. Cooperative games with overlapping coalitions [J]. *Journal of Artificial Intelligence Research*, 2010, 39(1): 179 – 216.
- [18] ZICK Y, ELKIND E. Arbitrators in overlapping coalition formation games [C] // *Proceedings of the 10th International Conference on Autonomous Agent and Multi-Agent Systems*. Taipei: IFAAMAS, 2011, 5: 55 – 62.
- [19] ZICK Y, CHALKIADAKIS G, ELKIND E. Overlapping coalition formation games: charting the tractability frontier [C] // *Proceedings of the 11th International Conference on Autonomous Agent and Multi-Agent Systems*. Valencia: IFAAMAS, 2012, 6: 787 – 794.
- [20] ZICK Y, ELKIND E, MARKAKIS E. Arbitration and stability in cooperative games with overlapping coalitions [J]. *Journal of Artificial Intelligence Research*, 2014, 50(1): 847 – 884.
- [21] ZHANG Guofu, ZHOU Peng, SU Zhaopin, et al. A payoff distribution strategy for overlapping coalitions based on bargaining [J]. *Pattern Recognition and Artificial Intelligence*, 2014, 27(10): 930 – 938.
(张国富, 周鹏, 苏兆品, 等. 一种基于讨价还价的重叠联盟效用划分策略 [J]. 模式识别与人工智能, 2014, 27(10): 930 – 938.)
- [22] GUI Haixia, JIANG Jianguo, ZHANG Guofu. Payoff distribution strategy of overlapping coalitions for concurrent multiple tasks [J]. *Pattern Recognition and Artificial Intelligence*, 2016, 29(4): 332 – 340.
(桂海霞, 蒋建国, 张国富. 面向并发多任务的重叠联盟效用分配策略 [J]. 模式识别与人工智能, 2016, 29(4): 332 – 340.)
- [23] SHEHORY O, KRAUS S. Formation of overlapping coalitions for precedence-ordered task-execution [C] // *Proceedings of the 2nd International Conference on Multi-agent Systems*. Kyoto: AAAI, 1996, 12: 330 – 337.
- [24] ZHAN Y, WU J, WANG C, et al. On the complexity and algorithms of coalition structure generation in overlapping coalition formation games [C] // *Proceedings of the 24th IEEE International Conference on Tools with Artificial Intelligence*. Athens: IEEE, 2012, 11: 868 – 873.
- [25] LIN C, HU S. Multi-task overlapping coalition parallel formation algorithm [C] // *Proceedings of the 6th International Joint Conference on Autonomous Agent and Multi-Agent Systems*. Honolulu: IFAAMAS, 2007, 5: 1260 – 1262.
- [26] ZHANG G, JIANG J, LU C, et al. A revision algorithm for invalid encodings in concurrent formation of overlapping coalitions [J]. *Applied Soft Computing*, 2011, 11(2): 2164 – 2172.
- [27] LI Junjun, GAN Shihong, XU Bogui. Discrete particle swarm optimization algorithm based on pseudo power function and its applications [J]. *Control Theory & Applications*, 2011, 28(6): 834 – 838.
(李建军, 甘世红, 许波桅. 基于伪幂函数的离散粒子群算法及其应用 [J]. 控制理论与应用, 2011, 28(6): 834 – 838.)
- [28] ZHANG Guofu, ZHOU Peng, JIANG Jianguo, et al. An algorithm for overlapping coalition formation based on virtual coalition [J]. *Acta Electronica Sinica*, 2012, 40(1): 121 – 127.
(张国富, 周鹏, 蒋建国, 等. 基于虚拟联盟的重叠联盟形成算法 [J]. 电子学报, 2012, 40(1): 121 – 127.)
- [29] DU Jiyong, ZHANG Fengming, HUI Xiaobin, et al. Overlapping coalition generation using improved continuous particle swarm optimization [J]. *Journal of Shanghai Jiao Tong University (Science)*, 2013, 47(12): 1918 – 1923.
(杜继永, 张凤鸣, 惠晓滨, 等. 改进型连续粒子群算法求解重叠联盟生成问题 [J]. 上海交通大学学报(自然版), 2013, 47(12): 1918 – 1923.)
- [30] LI Junjun, HUANG Youfang, YANG Bin. Alternately preying particle swarm optimization algorithm and convergence analysis of its particle trajectories [J]. *Control Theory & Applications*, 2013, 30(7): 850 – 855.
(李建军, 黄有方, 杨斌. 交替捕食的粒子群优化算法及其粒子轨迹收敛性分析 [J]. 控制理论与应用, 2013, 30(7): 850 – 855.)
- [31] DAS S, SUGANTHAN P N. Differential evolution: a survey of the state-of-the-art [J]. *IEEE Transactions on Evolutionary Computation*, 2011, 15(1): 4 – 31.
- [32] ZHOU Xiaogen, ZHANG Guijun, MEI Shan, et al. Differential evolution algorithm based on abstract convex underestimate selection strategy [J]. *Control Theory & Applications*, 2015, 32(3): 388 – 397.
(周晓根, 张贵军, 梅珊, 等. 基于抽象凸估计选择策略的差分进化算法 [J]. 控制理论与应用, 2015, 32(3): 388 – 397.)
- [33] BENJAMINI Y, YEKUTIELI D. The control of the false discovery rate in multiple testing under dependency [J]. *The Annals of Statistics*, 2001, 29(4): 1165 – 1188.
- [34] NGUYEN T T, YAO X. Continuous dynamic constrained optimization—the challenges [J]. *IEEE Transactions on Evolutionary Computation*, 2012, 16(6): 769 – 786.
- [35] ALTAMN D G, MACHIN D, BRYANT T N, et al. *Statistics With Confidence: Confidence Intervals and Statistical Guidelines* [M]. London: BMJ Books, 2000.

作者简介:

桂海霞 (1978–), 女, 博士研究生, 主要研究方向为多agent系统和进化计算, E-mail: guihaixia18@sohu.com;

张国富 (1979–), 男, 博士, 副教授, 英国伯明翰大学计算机学院访问学者(2015~2016), 中国自动化学会、IEEE会员, 主要研究方向为计算智能、多agent系统、基于搜索的软件工程, E-mail: zgf@hfut.edu.cn;

苏兆品 (1983–), 女, 博士, 副教授, IEEE会员, 主要研究方向为演化计算、灾害应急决策、多媒体安全, E-mail: szp@hfut.edu.cn;

蒋建国 (1955–), 男, 教授, 中国计算机学会高级会员, 硕士, 主要研究方向为分布式智能系统和数字图像处理与分析, E-mail: jgjiang@hfut.edu.cn.