

一种基于多智能体进化的广义图染色算法^{*}

李瑾姝, 刘 静⁺, 焦李成, 胡 康, 王景润

(西安电子科技大学 智能信息处理研究所, 陕西 西安 710071)

T-Coloring Algorithm Based on Multiagent Evolution

LI Jin-Shu, LIU Jing⁺, JIAO Li-Cheng, HU Kang, WANG Jing-Run

(Institute of Intelligent Information Processing, Xidian University, Xi'an 710071 Shannxi, China)

+ Corresponding author: E-mail: neouma@163.com

Li JS, Liu J, Jiao LC, Hu K, Wang JR. T-Coloring algorithm based on multiagent evolution. *Journal of Software*, 2009,20(2):315–326. <http://www.jos.org.cn/1000-9825/3199.htm>

Abstract: Based on the study of T-coloring problem, multiagent systems and evolutionary algorithms are integrated to form a new algorithm, multiagent evolutionary algorithm for T-coloring problem (MAEA-TCP). Then, this method is used to deal with the realistic frequency assignment problem, and has achieved encouraging results. In this algorithm, each agent is fixed on a lattice point of agent lattice as a possible solution. In order to increase energies, they compete or cooperate with their neighbors. They can also use knowledge to achieve their aims. Three evolutionary operators are designed for simulating the intelligent behaviors of agent, such as competition, self-learning and so on. The evolutionary operators are controlled through evolution, so that the populations can evolve. Experiments on large random graph instances and Philadelphia instances show that MAEA-TCP is a more encouraging algorithm than other methods.

Key words: agent; evolutionary algorithm; T-coloring problem; frequency assignment problem

摘 要: 基于对广义图染色问题的研究,提出了一种求解广义图染色问题的多智能体进化算法(multiagent evolutionary algorithm for T-coloring problem,简称 MAEA-TCP),并将该算法应用到实际中的频率分配问题上,取得了良好的效果.该方法中每个智能体作为一个候选解被固定在智能体网格上,为了增加自身能量而与邻域当中的智能体展开竞争或者合作,同时智能体也可以利用自身的知识进行自学习来增加能量.根据广义图染色问题的特点,为智能体设计了3种算子:竞争算子、自学习算子和变异算子,以引导其进化,并用进化的方式来控制各算子,以协调智能体之间的相互作用.在实验中,分别使用大规模的随机图实例和费城实例来测试算法性能,同时给出参数测试结果和最佳取值区间.比较结果表明,该算法优于其他方法,具有良好的收敛性和实用价值.

关键词: 智能体;进化算法;广义图染色问题;频率分配问题

中图法分类号: TP18 **文献标识码:** A

* Supported by the National Natural Science Foundation of China under Grant Nos.60872135, 60502043, 60602064 (国家自然科学基金); the Program for New Century Excellent Talents in University of China under Grant No.NCET-06-0857 (新世纪优秀人才支持计划); the National High-Tech Research and Development Plan of China under Grant No.2006AA01Z107 (国家高技术研究发展计划(863))

Received 2007-07-12; Accepted 2007-09-30

图染色是源于图论的一个经典的数学原型问题,也是一个典型的 NP 难题,具有广泛的应用背景.广义图染色问题(T-coloring problem,简称 TCP)是一般图染色问题的一种更广泛意义上的延伸,也更难以求解,使用的方法主要有遗传算法、模拟退火算法、Tabu 搜索算法等^[1].这类问题引起众多学者的兴趣,不仅因为它在计算上难以求解,而且还因为它有众多的实际应用,如频道分配问题、时间表问题、代码编辑器的记录分配以及一些计算机辅助应用等^[2].

频率分配问题(frequency assignment problem,简称 FAP)作为广义图染色问题的一个主要应用方面,在现代化的信息社会中,其重要性越来越显著.随着移动通信的迅速发展,移动用户急剧增加.伴随着小区的缩小,相应地用户密度增大,最终导致用户数量无限增长与系统资源的有限扩容这二者之间的矛盾越来越突出,如何提高现有资源的利用率已成为移动通信发展的瓶颈^[3].目前求解该问题的传统方法主要有穷举搜索法、启发式算法^[4]、神经网络的方法等.

进化算法是模拟生物在自然界中的遗传和进化过程而形成的一种自适应全局优化概率算法.它广泛应用于计算机科学的诸多领域.尽管进化算法较之传统的方法有巨大的优越性,但是现有的进化算法仍存在着一些共同的不足,例如适应度函数是预先定义好的而不是局部的、只考虑到生物之间的竞争没有考虑协作的可能性、忽视了个体的学习能力等等^[5].多智能体系统的研究近年来引起了人们的广泛关注.一般来说,智能体具有知识、目标和能力,任何一个能够感知环境并反作用于环境的实体都可被看作智能体,而所有由多个自治成分构成的系统都可表达为一个智能体系统^[6].本文将多智能体与进化算法相结合,提出一种求解广义图染色问题的多智能体进化算法.基于智能体适应环境的能力,我们为其设计了 3 种行为:竞争行为、自学习行为和变异行为,并以进化迭代的方式控制各个算子,最终达到快速收敛的效果.

本文第 1 节给出广义图染色问题和频率分配问题的数学模型.第 2 节详细描述智能体的定义、行为以及求解 TCP 的多智能体进化算法.第 3 节介绍我们如何将该算法应用于 FAP 上.第 4 节分别给出用随机图实例和费城实例对本文算法性能进行的全面测试结果.第 5 节总结全文.

1 问题描述

1.1 广义图染色问题

根据文献[2],对于一般的图染色问题,我们可以这样来描述:给定一个无向图 $G=(V,E)$,其中, $V=\{v_1,v_2,\dots,v_n\}$ 是所有节点的集合, $E=\{e_{ij}|i,j=1,2,\dots,n\}$ 是所有边的集合.给该图的节点染色,我们要使用最少的颜色数并保证相邻节点颜色不同.设 $c(v_i)$ 表示指派给节点 v_i 的颜色,那么正确的染色必须满足这样的约束:

$$\forall e_{ij} \in E, |c(v_i) - c(v_j)| \neq 0 \quad (1)$$

即相邻节点使用的颜色数值之差不能为 0.将其推广,就有了广义图染色问题的定义.首先定义一个集合 $T = \{T_{ij} \in \mathbb{N} \mid \text{对每一 } e_{ij} \in E\}$,表示相邻节点使用的颜色限制,其中 T_{ij} 是类似 $\{0,1,5,8\}$ 这样的无符号整数集合,指派给相邻节点的颜色必须满足如下约束:

$$\forall e_{ij} \in E, |c(v_i) - c(v_j)| \notin T_{ij} \quad (2)$$

相邻节点使用的颜色数值之差不能等于 T_{ij} 中的任何一个值.颜色数目是指在染色过程中使用不同颜色的最少数量,用 $\chi_T(G)$ 表示;颜色跨度表示在染色完成后使用的最大颜色数值与最小颜色数值之差,用 $span$ 表示. T-coloring 问题的目标就是寻找对一个无向图进行正确染色后的最小跨度,用 $sp_T(G)$ 表示.

如果每一个 $T_{ij} \in T$ 都是一个连续整数的集合 $T_{ij} = \{0,1,2,\dots,t_{ij}-1\}$,则有受限 T-coloring 的定义,约束(2)转化为

$$\forall e_{ij} \in E, |c(v_i) - c(v_j)| \geq t_{ij} \quad (3)$$

其中,每一个 t_{ij} 都是预先定义好的正整数值,被赋给每一条边.在本文中,我们将主要解决受限 T-coloring 问题.可以看出,一般图染色问题是广义图染色问题在 $T_{ij} = \{0\}$ 时的一个特例.

1.2 频率分配问题

频率分配问题最早源于无线电发报机,给一个区域内的不同发报机分配不同频率,以保证工作时相互之间不产生干扰^[7].目前研究的主要对象则是移动蜂窝网络系统.网络的服务区分成很多蜂窝(cell),每个蜂窝中都有大量的用户,要求很多不同的频率来满足用户间的通信要求.系统必须通过给每个用户发出的通话请求(call)指派适当的频率,保证满足一定的通信质量,调和用户间的通信以避免产生干扰.为了表示信道频率的相互约束,根据文献[4]中所使用的方法,我们定义相容性矩阵 $C=[c_{ij}]$,矩阵中的每一个元素 c_{ij} 用来表示指派给一对通话上的两个频率不产生干扰时所允许的最小频率差值,这样的一对通话分别来自蜂窝 i 和蜂窝 j .

频率分配问题主要分为两种:一种是固定频率分配,是指对频率点进行合理分配时尽可能多地满足约束条件,以获得干扰因素最小化;另一种是最小频率跨度分配,是指在保证满足所有约束条件的情况下,使得占用的频率跨度最小.基于我们前面的工作与 FAP 的相关性,现在我们讨论的是最小跨度频率分配问题(minimum span frequency assignment problem,简称 MSFAP).

Hale 在文献[8]中已经用图论的语言描述了该问题.给定一个无向图 $G=(V,E)$,其中的节点集合 $V=\{v_1,v_2,\dots,v_n\}$ 可以表示用户发出的所有通话的集合 $V=\{x_1,x_2,\dots,x_n\}$,边的集合可以表示通话之间的约束.设 f_{ik} 表示指派给第 i 个蜂窝中第 k 个通话的频率,取值为非负整数,则频率分配问题的数学模型可表示如下:

$$\begin{cases} \text{Minimize} & \max_{i,k} f_{ik} \\ \text{subject to} & |f_{ik} - f_{jl}| \geq c_{ij}, \text{ for all } i, j, k, l (k \neq l, \text{ if } i = j) \end{cases} \quad (4)$$

频率分配问题的跨度 $span$ 是指在对所有通话分配了频率之后所使用过的最大和最小频率间的差值.我们的目标就是找到这样一个最小的跨度 $sp_T(G)$.

2 求解广义图染色的多智能体进化算法

2.1 TCP中智能体的定义

智能体是一个物理的或虚拟的实体,主要具有以下特性^[9,10]:(1) 生存在一个特定的环境中;(2) 能够感知所处的局部环境;(3) 具有一些自治的行为;(4) 其行为是由某种目的驱动的.因此,智能体的含义非常广泛.针对不同的问题,智能体的具体含义和表达也有所区别.下面我们给出在求解广义图染色问题中使用的智能体的定义.

定义 1. 在广义图染色中,智能体就是解空间 \mathbf{S} 中的一个候选解.我们的算法中存在两种相互关联的智能体.一种智能体称为排列智能体,表示对无向图各节点进行染色时的顺序,因此排列智能体表达为一个非重复的自然数序列:

$$agent = (n_1, n_2, \dots, n_N) \in \mathbf{S}, n_i \in \mathbf{IN} \quad (5)$$

其中 N 表示节点个数, \mathbf{IN} 表示自然数集合.

另一种智能体称为解智能体,表示根据某一个排列智能体进行染色之后各节点的染色结果,因此解智能体表达为一个可重复的自然数序列:

$$agent = (a_1, a_2, \dots, a_N) \in \mathbf{S}, a_i \in \mathbf{IN} \quad (6)$$

每一个排列智能体都对应一个解智能体,但并非一一对应,不同的排列智能体可能得到相同的解智能体.

定义 2. 由于我们关心的是染色之后得到的最小跨度 $sp_T(G)$,因此智能体的能量被定义为跨度的相反数:

$$\forall agent \in \mathbf{S}, Energy(agent) = -sp_T(G) \quad (7)$$

每个智能体的目标是最大化自身的能量,可以看出,能量越大,跨度就越小,也越接近最优值.

由于每个智能体必须携带一些信息,我们将智能体的信息结构表示如下:

- P : 表示一种排列, $P \in \mathbf{S}$.
- $E(agent)$: 根据排列智能体进行染色后得到的解智能体的能量, $E(agent) = Energy(P)$.
- SL : 自学习标志.在自学习行为中,当 SL 值为真时,才可以进行自学习行为.

定义 3. 所有的智能体都生存在一个环形网格状的环境中,称为智能体网格,记为 L .设智能体网格的规模为

$(L_{size} \times L_{size})$, 其组织形式如图 1 所示, 其中每个圆圈代表一个智能体, 圆圈当中的数字表示该智能体所处的位置, 以下记处在第 i 行第 j 列的智能体为 $L_{i,j}$. 每个智能体占据一个格点位置且不能移动, 但由于它有感知局部环境的能力, 因此可以与周围的智能体发生相互作用. 与之发生作用的智能体称为该智能体的邻域 $Neighbors_{i,j}$:

$$Neighbors_{i,j} = \{L_{i',j}, L_{i,j'}, L_{i'',j}, L_{i,j''}\} \tag{8}$$

$$\text{其中, } i' = \begin{cases} i-1, & i \neq 1 \\ L_{size}, & i = 1 \end{cases}, j' = \begin{cases} j-1, & j \neq 1 \\ L_{size}, & j = 1 \end{cases}, i'' = \begin{cases} i+1, & i \neq L_{size} \\ 1, & i = L_{size} \end{cases}, j'' = \begin{cases} j+1, & j \neq L_{size} \\ 1, & j = L_{size} \end{cases}$$

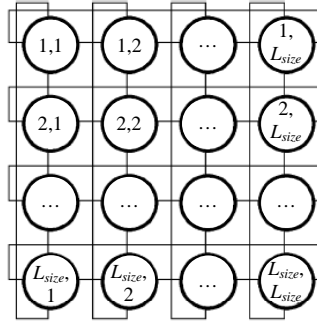


Fig.1 Agent lattice

图 1 智能体网格

2.2 智能体的行为

在求解问题时, 我们希望以尽可能少的计算量和时间求得高质量的解, 因此计算代价可以被看作是智能体生存环境里的资源. 每个智能体的行为都由目的驱动, 而资源又是有限的, 因此, 智能体需要通过相互竞争以获取更多的生存资源. 基于这种思想, 我们设计了 3 种行为来实现智能体的目的, 它们分别是竞争行为、自学习行为和变异行为. 其中竞争行为利用邻域的信息来实现智能体之间的竞争与合作, 自学习行为和变异行为利用智能体自身的知识来增加能量.

A. 竞争行为

在这种行为中, 每个智能体与邻域中的其他智能体进行能量比较, 若其能量不小于周围其他所有智能体的能量, 则可以存活下来; 否则, 它将死亡, 由邻域内能量最大的一个智能体产生的子代来替换该智能体的格点位置. 我们将这一过程描述如下:

设竞争行为作用在格点位置为 (i,j) 的智能体 $L_{i,j}$ 上, $Energy$ 为能量网格, 其中存放 L 中所有智能体当前的能量值, 智能体 $L_{i,j}$ 的能量用 $E(L_{i,j})$ 表示, E 表示任意某个智能体的能量. $Max_{i,j}$ 表示该智能体邻域内能量最大的一个智能体, 即 $Max_{i,j} \in Neighbors_{i,j}$, 对任意 $agent \in Neighbors_{i,j}$ 有 $E(agent) < E(Max_{i,j})$. 如果 $E(L_{i,j}) < E(Max_{i,j})$, 则由 $Max_{i,j}$ 产生一个子代智能体 $Child_{i,j}$ 来替代 $L_{i,j}$, 算法设计如算法 1 所示; 否则, $L_{i,j}$ 保持不变.

算法 1. 竞争算子.

输入: 邻域内能量最大的智能体 $Max_{i,j}$, $Max_{i,j}(P) := \langle m_1, m_2, \dots, m_N \rangle$; 竞争概率 P_c .

输出: 由 $Max_{i,j}$ 产生的子代 $Child_{i,j}(P) := \langle c_1, c_2, \dots, c_N \rangle$.

在下面的步骤中, $Swap(x,y)$ 表示交换排列中 x 与 y 的值, $rang(0,1)$ 表示 0,1 之间的一个随机数, $Random(N,i)$ 表示任意一个不等于 i 的从 1 到 N 的整数, N 为图节点的个数. 在竞争行为中被替换掉的智能体, 由于其排列已经被改变, 因此允许在后面的进化中具有自学习的权利, 在算法最后将其自学习标志还原为 True.

Step 1. 初始化 $Child_{i,j}$, 将 $Max_{i,j}(P)$ 的排列赋给 $Child_{i,j}(P)$, 令 $i \leftarrow 1$.

Step 2. 产生一个 $rang(0,1)$, 若 $rang(0,1) < P_c$, 则转 Step 3; 否则, 令 $i \leftarrow i + 1$, 转 Step 4.

Step 3. 产生一个 $l \leftarrow Random(N,i)$, 执行 $Swap(c_i, c_l)$, 令 $i \leftarrow i + 1$.

Step 4. 若 $i > N$, 则将新得到的 $Child_{i,j}(P)$ 的排列赋给 $L_{i,j}$, 转 Step 5; 否则, 转 Step 2.

Step 5. 令 $L_{i,j}(SL) \leftarrow \text{True}$, 计算 $E(L_{i,j})$ 并更新智能体网格 L 与能量网格 $Energy$, 结束.

B. 自学习行为

由于智能体在求解问题的过程中具有与问题相关的知识,因此我们利用局部搜索技术设计智能体的自学习行为.在这个行为当中,智能体使用知识来提高能量以提升自己的竞争能力.当自学习行为作用于一个智能体后,该智能体再次通过这种行为提高能量的可能性就非常小了,因此最后将其自学习标志置为 False . 为了降低计算代价,我们只允许当代最优的智能体获得一次自学习机会.

算法 2. 自学习算子.

输入:当代能量最优的智能体 $agent_{best}, agent_{best}(P) := \langle p_1, p_2, \dots, p_N \rangle$.

输出:执行自学习行为之后的智能体 $agent_{learn}, agent_{learn}(P) := \langle a_1, a_2, \dots, a_N \rangle$.

在下面的步骤中, $E(agent_{best})$ 表示 $agent_{best}$ 的能量, $E(agent_{learn})$ 表示 $agent_{learn}$ 的能量. $Iteration$ 是一个迭代参数,表示对某一位选择替换位的次数,最大限为 $N.Repeat$, 这是一个表示迭代是否继续的布尔型变量. 设 $L_{i,j} := agent_{best}$.

Step 1. 初始化 $agent_{learn}$, 将当前的 $agent_{best}(P)$ 赋给 $agent_{learn}(P)$, 将 $E(agent_{learn}) \leftarrow E(agent_{best})$, 令 $Repeat \leftarrow \text{False}, k \leftarrow 1, Iteration \leftarrow 1$.

Step 2. 若 $Iteration \geq N - 1$, 则令 $k \leftarrow k + 1, Iteration \leftarrow 1$.

Step 3. 产生一个 $l \leftarrow \text{Random}(N, k), \text{Swap}(a_k, a_l)$, 重新计算 $E(agent_{learn})$.

Step 4. 若 $E(agent_{best}) \geq E(agent_{learn})$, 则再执行 $\text{Swap}(a_k, a_l), Iteration \leftarrow Iteration + 1$, 转 Step 2; 否则, 令 $Repeat \leftarrow \text{True}, k \leftarrow k + 1, Iteration \leftarrow 1$, 转 Step 5.

Step 5. 若 $k < N$, 则转 Step 3; 否则, 转 Step 6.

Step 6. 若 $Repeat = \text{True}$, 则 $k \leftarrow 1, Iteration \leftarrow 1$ 转 Step 3; 否则, 令 $L_{i,j}(SL) \leftarrow \text{False}$, 更新 L 与 $Energy$, 结束.

C. 变异行为

在算法中,变异行为类似于遗传算法中的变异算子,作为另外两种行为的辅助,该行为扩大了算法的搜索空间,避免过早地收敛于局部最优.由于智能体具有一定的知识,而最优的智能体具有最丰富的信息、最强有力的基因,是最有潜力接近问题目标的个体,因此我们使用最优智能体扩散策略,令当代种群当中的一部分个体死亡,由当代最优智能体根据变异概率 P_m 自身发生微小变异,产生相同数量的子代智能体替换这些死亡智能体的位置.在实验中我们发现,竞争行为和自学习行为具有较为快速的收敛效果,但同时将多数的次优解扩散到种群中,使其很难继续进化,变异行为有效地保持了种群的多样性,避免了早熟收敛.

设变异行为作用在某个智能体 $L_{i,j}$ 上,且 $L_{i,j}(P) = \langle p_1, p_2, \dots, p_N \rangle$, 则执行下面的操作:

$$\text{if } \text{rand}_k(0,1) < P_m, \text{ then } L_{i,j}(P_k) = \text{Child}(agent_{best}) \quad (9)$$

其中, $k=1,2,\dots,N \times 0.35, P_m$ 是一个预先设定好的, 0,1 之间的实数,被施加变异行为的所有智能体, $L_{i,j}(SL) \leftarrow \text{True}$. 变异后更新智能体网格 L 与能量网格 $Energy$.

2.3 求解广义图染色问题的多智能体进化算法(MAEA-TCP)

在算法中,我们使用进化的方式控制各个算子.在每一代的开始,首先将竞争行为作用于智能体网格上的所有智能体,这样,具有较低能量的智能体被清除出网格,给有潜力的智能体以更大的发展空间.之后,根据自学习标志实行自学习行为,为了降低计算代价,只将该行为作用于当代最优的智能体上.最后,变异行为作用于网格当中的一部分智能体上.整个过程不断迭代,直到找到最优解或者达到预设的最大计算代价为止.

算法 3. MAEA-TCP.

输入:最大计算次数 $Evaluation_{\max}$;最大进化代数 Gen_{\max} ;智能体网格规模 L_{size} ;竞争概率 P_c ;变异概率 P_m .

输出:最优的智能体 $agent_{best}$ 和 $sp_T(G)$.

在下面的步骤中, L' 表示进化到第 t 代时的智能体网格, $agent'_{best}$ 表示在 L^0, L^1, \dots, L' 中最优的一个智能体, $agent'_{t,best}$ 表示在 L' 中最优的一个智能体,而 $agent'_{t,worst}$ 则表示 L' 中最差的智能体.

- Step 1. 初始化智能体网格 L^0 , 更新 $agent_{best}^t$, 并将所有智能体的 SL 均置为 True, 令 $t \leftarrow 1, Evaluations = L_{size} \times L_{size}$.
- Step 2. 对 L 中的所有智能体, 若 $rand(0,1) < P_c$, 则执行竞争算子, $Evaluations \leftarrow Evaluations + 1$.
- Step 3. 若 $agent_{t,best}^t(SL) = True$, 则对 $agent_{t,best}^t$ 执行自学习算子, $agent_{t,best}^t(SL) \leftarrow False, Evaluations \leftarrow Evaluations + 1$.
- Step 4. 若自学习行为结束后 $E(agent_{t,best}^t) > E(agent_{best}^t)$, 则令 $agent_{best}^t \leftarrow agent_{t,best}^t, agent_{t,worst}^t \leftarrow agent_{t,best}^t$, 更新 L .
- Step 5. 对 L 中 35% 的智能体根据变异概率施加变异行为, 并将其自学习标志置为 True, 更新 $L, t \leftarrow t + 1, Evaluations \leftarrow Evaluations + N \times 0.35$.
- Step 6. 若 $t > Gen_{max}$ 或 $Evaluations > Evaluation_{max}$, 则输出 $E(agent_{best}^t)$ 为算法求得最优值; 否则, $t \leftarrow t + 1$, 转 Step 2.

3 多智能体进化算法求解频率分配问题

3.1 编码思想

从第 2 节中我们看到, 频率分配问题的数学原型就是广义图染色问题, 但在实际应用中是有所不同的. 频率分配问题研究的对象是移动蜂窝网络, 为了便于控制, 一个地区的网络通常被分为若干个蜂窝, 每个蜂窝内有若干个用户, 会对蜂窝内以及其他蜂窝中的用户发出通话请求. 同一个蜂窝内部发出的通话相互之间需要更大的频率间隔以避免发生干扰, 与其他蜂窝中的用户通话的频率间隔取决于空间距离的远近, 距离越远, 需要的频率间隔就越小, 甚至在一定范围之外可以采用相同频率而不产生通话干涉. 因此, 在频率分配问题中, 虽然每一个通话可以看作广义图染色中的节点, 但 TCP 中的节点是各个独立而不同的, 而 FAP 中的通话是以蜂窝为单位成组出现的, 同一个蜂窝中的通话具有相同的外部约束.

基于问题的这种特性, 很多算法采用集体分配的思想, 将通话以蜂窝为单位进行频率分配. 分配的顺序采用一种“复杂度”的策略. 蜂窝的复杂度取决于该蜂窝内部的通话请求数目, 与其临近蜂窝的数量、远近以及它们内部的通话请求数量等. 蜂窝越复杂, 在分配中占用的资源越多, 因此具有优先分配的资格, 以利于减少使用的频率值. “度”的数学定义在文献[4]中已有所描述, 第 i 个蜂窝的度 d_i 定义如下:

$$d_i = \left(\sum_{j=1}^N c_{ij} m_j \right) - c_{ii}, \quad 1 \leq i \leq N \quad (10)$$

其中, N 表示蜂窝的总数, m_j 表示第 j 个蜂窝中发出的 calls 的数目.

文献[4]是 Won-Young 等人在 2006 年提出的一种启发式算法, 称为频率插入策略(FIS). 该算法采用将蜂窝按度从高到低的顺序分配其中的通话, 在分配过程中允许产生不满足约束的冲突, 当分配完一个蜂窝中所有的通话之后, 在产生冲突的地方插入频率来解决冲突.

实验中我们发现, 完全按照度的方式来指派频率会极大地缩小搜索范围, 导致算法陷入局部最优而无法跳出. 因此, 在我们的算法中, 编码方式不以蜂窝为单元, 而是所有通话的排列, 用蜂窝的序号作为通话的代号. 下面我们继续用多智能体进化算法求解频率分配问题. 首先给出在频率分配中使用的智能体的定义.

定义 4. 在频率分配问题中, 智能体定义为所有蜂窝内通话的一个排列, 表达为一个可重复的自然数序列:

$$agent = (c_1, c_2, \dots, c_l), \quad c_i \in IN \quad (11)$$

其中, l 是所有通话的总数目, c_i 表示来自蜂窝 i 的一个通话.

定义 5. 类似于对 TCP 中能量的定义, 在 FAP 中我们定义智能体的能量为频率跨度 $span$ 的相反数:

$$\forall agent \in \mathbf{S}, \quad Energy(agent) = -span(FAP) \quad (12)$$

我们的目标是寻找一种指派的方式, 使得给每个通话在指派了频率之后, 频率的跨度最小, 这主要是基于经济因素的考虑. 智能体的信息结构与智能体网格的定义均与 TCP 中的智能体相似, 这里不再赘述.

3.2 求解FAP的多智能体进化算法(MAEA-FAP)

在算法中,我们依然使用算法3的设计思想,同时针对频率分配问题对算子作了一定的改进.在竞争行为中,邻域内最优智能体产生子代的方式有所改变,在交换任意两个位置的通话时,保证交换到前面的一个通话所在的蜂窝,是度较大的一个.具体步骤如下,其中, $p(x)$ 表示智能体中通话 x 的位置,当 x 是排列中第1个通话时, $p(x)=1$;当 x 是排列中最后一个通话时, $p(x)=l$. $bigger(x,y)$ 表示 x 与 y 中较大的一个的值, $smaller(x,y)$ 则表示其中较小的一个的值.

算法4. 在FAP中的竞争行为.

Step 1. 初始化 $Child_{i,j}$, 将 $Max_{i,j}(P)$ 的排列赋给 $Child_{i,j}(P)$, 令 $pos \leftarrow 1$.

Step 2. 若 $pos > l$, 则转 Step 5; 否则, 转 Step 3.

Step 3. 产生一个 $rand(0,1)$, 若 $rand(0,1) < P_c$, 则转 Step 4; 否则, 令 $pos \leftarrow pos + 1$, 转 Step 2.

Step 4. 产生一个 $k \leftarrow Random(l, pos)$, 令 $b \leftarrow bigger(p(pos), p(k))$, $s \leftarrow smaller(p(pos), p(k))$, 若 $d_b > d_s$, $Swap(c_{pos}, c_k)$, 则令 $pos \leftarrow pos + 1$, 转 Step 5; 否则, $pos \leftarrow pos + 1$, 转 Step 2.

Step 5. 若 $pos > l$, 则将新得到的 $Child_{i,j}(P)$ 的排列赋给 $L_{i,j}$, 转 Step 6; 否则, 转 Step 3.

Step 6. 令 $L_{i,j}(SL) \leftarrow True$, 计算 $E(L_{i,j})$ 并更新智能体网格 L 与能量网格 $Energy$, 结束.

在自学习行为中,我们使用同样的策略,对两个交换的通话,以一定的概率保证度较大的一个通话被换到前面,我们称这样的概率为“交换概率”,用 P_d 表示. $Iteration$ 的最大限也由算法2中的编码长度 N 变为FAP中的 $l/4$. 设当前最优智能体为 $agent_{best}=L_{i,j}$.

算法5. 在FAP中的自学习行为.

Step 1. 初始化 $agent_{learn}$, 将当前的 $agent_{best}(P)$ 赋给 $agent_{learn}(P)$, 将 $E(agent_{learn}) \leftarrow E(agent_{best})$, 令 $Repeat \leftarrow False$, $pos \leftarrow 1$, $Iteration \leftarrow 1$.

Step 2. 若 $Iteration \geq l/4 - 1$, 则令 $pos \leftarrow pos + 1$, $Iteration \leftarrow 1$; 否则, 转 Step 3.

Step 3. 产生一个 $k \leftarrow Random(l, pos)$ 和一个 $rand(0,1)$, 令 $b \leftarrow bigger(p(pos), p(k))$, $s \leftarrow smaller(p(pos), p(k))$, 假如 $d_b > d_s$ 且 $rand(0,1) < P_c$, $Swap(a_{pos}, a_k)$, 计算 $E(agent_{learn})$.

Step 4. 若 $E(agent_{best}) \geq E(agent_{learn})$, 则再执行 $Swap(a_{pos}, a_k)$, $Iteration \leftarrow Iteration + 1$, 转 Step 2; 否则, 令 $Repeat \leftarrow True$, $pos \leftarrow pos + 1$, $Iteration \leftarrow 1$, 转 Step 5.

Step 5. 若 $pos < 1$, 则转 Step 3; 否则, 转 Step 6.

Step 6. 若 $Repeat=True$, $pos \leftarrow 1$, $Iteration \leftarrow 1$, 转 Step 3; 否则, 令 $L_{i,j}(SL) \leftarrow False$, 更新 L 与 $Energy$, 结束.

在求解频率分配问题时,我们继续使用算法3的流程,只需将其中的竞争行为和自学习行为分别替换为算法4和算法5即可.

4 对比实验与结果分析

4.1 随机图实验结果

我们采用Costa在文献[1]中提出的随机图产生方法来产生实例以测试算法性能.该方法针对受限广义图染色问题,根据产生一般图染色问题中无向图的规则,产生不同复杂度的广义随机图.首先需要定义3个参数:

- N : 节点个数;
- $d \in [0,1]$: 无向图边密度;
- Sep : 颜色约束值最大范围.

我们首先根据不同的节点个数和边密度来产生无向图,边的数目等于 $d(N(N-1))/2$, 然后随机选取 $[1, Sep]$ 中的值赋给每一条边,作为相邻节点的颜色约束.节点个数分别取 $N \in \{30, 100, 300, 500, 1000\}$, 边密度分别取 $d \in \{0.1, 0.5, 0.9\}$, 颜色约束取值为 $Sep=5$, 由此我们产生了15个广义随机图,分别以 $N.d.Tcol$ 的方式命名,根据节点不同分为5组,边的增长变化如图2所示.

30.1.Tcol	100.1.Tcol	300.1.Tcol	500.1.Tcol	1000.1.Tcol
30.5.Tcol	100.5.Tcol	300.5.Tcol	500.5.Tcol	1000.5.Tcol
30.9.Tcol	100.9.Tcol	300.9.Tcol	500.9.Tcol	1000.9.Tcol

Fig.2 Instances of random graphs used in the experiments

图2 实验中的随机图实例

在实验中,我们对图2中的所有15个随机图用多智能体进化算法进行求解,并与以“饱和度”为基础的经典Dsatur算法所求得的结果进行了比照.我们采用所有计算次数中的最优解和若干次独立随机运算中的平均最优值作为测试算法性能的指标,分别用 sp_{best} 和 sp_{ave} 表示.在MAEA-TCP中,各参数的取值如下: $P_c=0.6$, $P_m=0.5$, $Gen_{max}=200$,最大计算次数为 10^6 .独立随机运算次数根据问题复杂度不同而有所改变,对较为简单的30.1.Tcol进行100次随机运算,而对最为复杂的问题1000.9.Tcol,我们只进行10次随机运算.实验结果及对比见表1.结果表明,对这15个问题,我们的算法求得的最优值和平均值均好于Dsatur算法,显示了MAEA-TCP在求解复杂问题上的优势和潜力.

Table 1 Comparison between the performances of Dsatur and MAEA-T-coloring

表1 MAEA-TCP与Dsatur的性能比较

Random graphs	Dsatur		MAEA-T-coloring	
	sp_{best}	sp_{ave}	sp_{best}	sp_{ave}
30.1.Tcol	9	9.0	8	8.0
30.5.Tcol	23	25.2	19	19.0
30.9.Tcol	40	44.9	34	35.6
100.1.Tcol	23	23.0	17	17.8
100.5.Tcol	63	66.1	58	59.0
100.9.Tcol	124	125.0	112	114.1
300.1.Tcol	40	43.8	35	35.6
300.5.Tcol	159	161.3	156	157.8
300.9.Tcol	304	318.9	292	294.5
500.1.Tcol	61	62.5	55	56.6
500.5.Tcol	240	247.5	231	232.3
500.9.Tcol	483	493.1	473	474.1
1000.1.Tcol	104	105.1	92	93.7
1000.5.Tcol	436	441.4	426	430.3
1000.9.Tcol	896	904.2	879	886.7

4.2 随机图实例参数分析

为了找到较为合适的有效的参数取值,我们对两个主要参数竞争概率 P_c 和智能体网格规模 L_{size} 进行测试,观察若干次随机运算的平均结果Average Span和平均函数评价次数Average Evaluation随着这两个参数变化所产生的结果.在对所有随机图实例参数测试之后,我们选取了其中较有代表性的3个复杂度依次递增的测试问题,将结果展示如下(图3~图5).

在测试中,我们对每一个参数的取值组合随机运行100次,其中每一次的最大进化代数都是100代.从图3~图5中我们基本可以看出,对较为简单的图30.1.Tcol来说, L_{size} 取值为8~14, P_c 在0.1~0.45之间,可以保证有较好的平均最优值;平均函数评价次数随着 L_{size} 的增加平稳上升,在 P_c 取值0.2~1之间较为平坦,而在 P_c 取值0.05~0.2之间迅速下降.对于复杂一些的另外两个图(100.5.Tcol,100.9.Tcol)来说,平均最优值在 P_c 取值为0.05~0.3时迅速上升,在0.3~1之间变化不大,略有上扬,而随着 L_{size} 的增加略有下降,变化不大;平均函数评价次数在 P_c 取值为0.15~0.05之间变化剧烈,极速上升,随着 L_{size} 的增加平稳上升,基本呈线性增长.

由此可以得出两个结论:对于较为简单的问题, L_{size} 的值越小则函数评价次数越低,但是考虑计算结果的性能, P_c 取值在0.1~0.5之间, L_{size} 取值在8左右,可以保证取得较好的目标值,同时使用较少的评价次数;随着问题复杂度的增加,函数评价次数随着 L_{size} 的增加呈线性增加趋势,在 P_c 取值为0.1~1之间变化不大,但在 P_c 取0.05时迅速增加;而平均最优值随着 P_c 的减小会变优.因此, P_c 与 L_{size} 对算法性能的影响是一对矛盾,需要根据问题的不同特性和规模选取合适的取值范围.

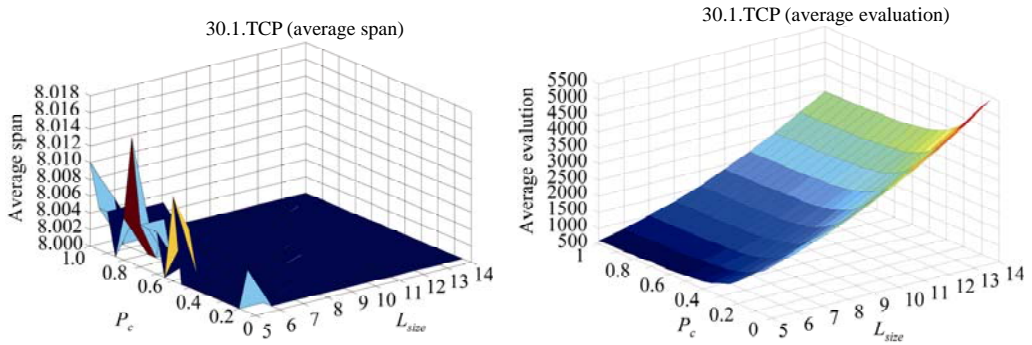


Fig.3 Parameter test results of 30.1.Tcol ($N=30,E=43$)

图3 30.1.Tcol 参数测试结果($N=30,E=43$)

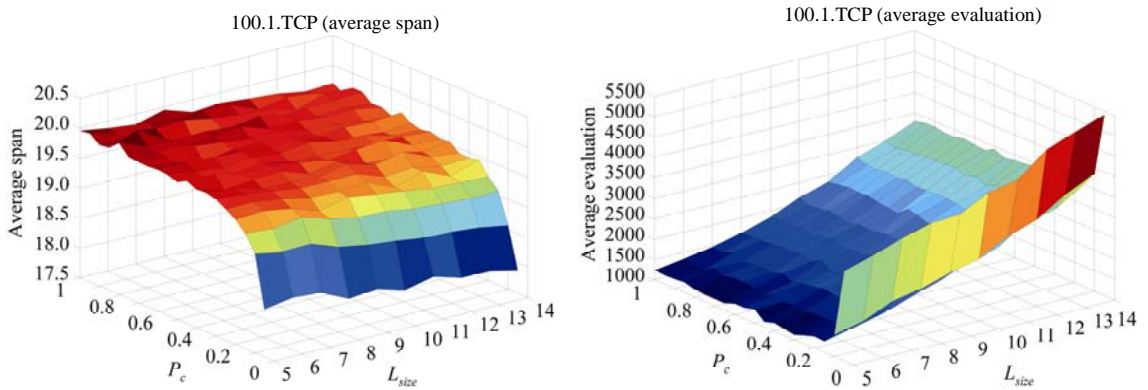


Fig.4 Parameter test results of 100.5.Tcol ($N=100,E=495$)

图4 100.5.Tcol 参数测试结果($N=100,E=495$)

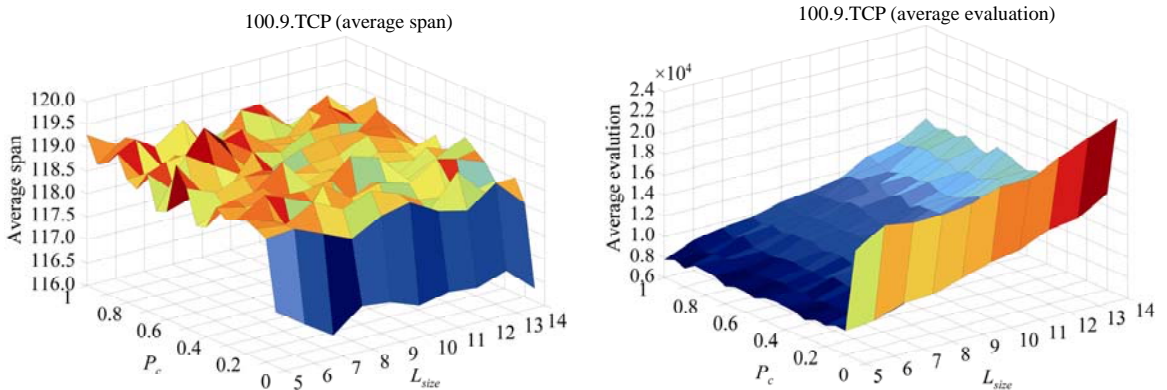


Fig.5 Parameter test results of 100.9.Tcol ($N=100,E=4455$)

图5 100.9.Tcol 参数测试结果($N=100,E=4455$)

4.3 费城实例实验结果

费城实例(Philadelphia benchmark instance)在1973年由 Anderson 提出,分为9组问题,后来被广泛使用,作为评价求解频率分配问题的算法性能的标准测试问题.该实例用21个六边形表示美国费城移动通信网络中的

不同蜂窝,如图 6 所示.每一个蜂窝中都发出不同数量的通话请求,构成一个长度为 21 的通话请求向量.图 7 蜂窝中的数字是对应实例 1 中的通话请求向量,表 2 中所示的是所有 9 个测试问题的通话请求向量.在频率分配问题中,蜂窝间的干扰约束是用邻信道再利用距离 d 描述的,当且仅当两个蜂窝中心距离大于或等于 d 时,它们之间才不会产生干扰,如果这个中心距离小于 d ,则这两个蜂窝内部的通话不可以被指派为相同的频率.这里不仅仅有一种 d ,而是有一系列这样不同的 d 值,我们记为 d^0, \dots, d^k .例如在 $P1$ 中, d^0, \dots, d^5 取值为 $2\sqrt{3}, \sqrt{3}, 1, 1, 1, 0$, 即中心距大于等于 $2\sqrt{3}$ 的蜂窝内的通话,可以被指派相等的频率,这里, k 表示允许使用的频率的最小差值.因此中心距为 0(即在同一个蜂窝内)的通话只能使用相差大于等于 5 的频率值.对测试集中的其他问题的距离约束条件,见表 3.

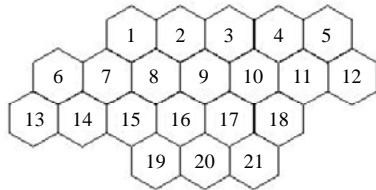


Fig.6 Network structure of the Philadelphia instances

图 6 费城实例网络结构

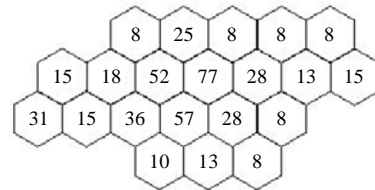


Fig.7 Call demand of vector P1

图 7 向量 P1 中的通话请求

Table 2 Call demand vectors of the Philadelphia instances

表 2 费城实例中的通话请求向量

	Demand vector
$P1$	(8,25,8,8,8,15,18,52,77,28,13,15,31,15,36,57,28,8,10,13,8)
$P2$	(8,25,8,8,8,15,18,52,77,28,13,15,31,15,36,57,28,8,10,13,8)
$P3$	(5,5,5,8,12,25,30,25,30,40,40,45,20,30,25,15,15,30,20,20,25)
$P4$	(5,5,5,8,12,25,30,25,30,40,40,45,20,30,25,15,15,30,20,20,25)
$P5$	(20,20)
$P6$	(20,20)
$P7$	(16,50,16,16,16,30,36,104,154,56,26,30,62,30,72,114,56,16,20,26,16)
$P8$	(8,25,8,8,8,15,18,52,77,28,13,15,31,15,36,57,28,8,10,13,8)
$P9$	(32,100,32,32,32,60,72,208,308,112,52,60,124,60,144,228,112,32,40,52,32)

Table 3 Frequency interference constrains of the Philadelphia instances

表 3 费城实例中的频率约束

Instances	Reuse distances
$P1, P3, P5, P7, P9$	$(2\sqrt{3}, \sqrt{3}, 1, 1, 1, 0)$
$P2, P4, P6$	$(\sqrt{7}, \sqrt{3}, 1, 1, 1, 0)$
$P8$	$(2\sqrt{3}, 2, 1, 1, 1, 0)$

已有很多研究致力于找到费城实例的上、下界.表 4 总结了目前为止所能得到的关于费城实例的理论值情况^[11-18],阴影部分是最优值(数据来源:<http://fap.zib.de/problems/Philadelphia/>).在表 5 中,我们列出了 MAEA-FAP 算法与前面介绍过的 FIS 算法的测试性能比较.算法的终止条件是找到理论最优值,或者达到最大的进化代数 100,或是达到最大的计算次数 10^6 次.在算法中, L_{size}, P_c, P_m 和 P_d , 分别取值为 15, 0.5, 0.4, 0.8.我们用能够求得的最优值 sp_{best} 和 50 次独立运算的平均最优值 sp_{ave} 作为比较参数,正如表 5 中所显示的,我们的算法在所有问题中求得的最优解和平均最优解均优于或等于 FIS 的最优值,在 9 个问题中,MAEA-FAP 求得了其中 6 个问题的理论最优值,充分显示了该算法在求解大规模实际问题中的高性能和巨大潜力.

Table 4 Available experimental results for the Philadelphia instances**表 4** 对费城实例已有的实验结果

Vectors	Lower bounds					Known optimum	Upper bounds					
	Ref. [11]	Ref. [12]	Ref. [13]	Ref. [14]	Ref. [15]		Ref. [12]	Ref. [16]	Ref. [12]	Ref. [17]	Ref. [18]	Ref. [15]
P1	426	426	426	–	426	426	428	426	426	–	426	426
P2	426	426	426	–	426	426	438	426	426	–	426	426
P3	–	257	252	–	257	257	260	258	257	–	257	275
P4	252	252	252	–	252	252	259	253	252	–	252	253
P5	–	239	177	–	239	239	239	239	–	–	239	239
P6	177	178	177	179	177	179	200	198	–	179	179	183
P7	–	855	855	–	855	855	858	856	855	–	855	855
P8	–	524	427	–	523	524	546	527	–	–	524	524
P9	–	1 713	1 713	–	1 713	1 713	1 724	–	1 713	–	–	1 713

Table 5 Comparison between the experimental results of MAEA-FAP and FIS**表 5** MAEA-FAP 与 FIS 的实验结果比较

Vectors	FIS	MAEA-MSFAP		Optima
	SP_{best}	SP_{best}	SP_{ave}	
P1	426	426	426.00	426
P2	426	426	426.00	426
P3	298	261	265.00	257
P4	263	252	254.03	252
P5	268	240	241.15	239
P6	222	194	199.00	179
P7	855	855	855.00	855
P8	538	524	525.65	524
P9	1 713	1 713	1 713.00	1 713

5 结 论

本文提出了一种求解广义图染色问题的多智能体进化算法 MARA-TCP.该算法将多智能体系统与进化算法的机制相结合,充分运用智能体的局部感知能力以及自学习能力,根据问题不同特点设计了相应的智能体编码与行为,同时将算法用于求解具有实际应用的频率分配问题上.在实验中,我们用大规模的随机图和费城实例对算法进行全面的性能测试,对参数设置给出了一定的范围,并且以上所有结果均在相同参数下获得,说明算法的鲁棒性强,便于使用.可以看出,我们的算法有较大的成功率和较优的求解效果,具有很大的挖掘潜力与实际意义.

References:

- [1] Costa D. On the use of some known methods for T-colorings of graphs. *Annals of Operations Research*, 1993,41(1):343–358.
- [2] Dorne R, Hao JK. Tabu search for graph coloring, T-colorings and set T-colorings. In: Voß S, Martello S, Roucairol C, Osman IH, eds. *Meta-Heuristics'98: Theory and Applications*. Boston: Kluwer Academic Publishers, 1998. 33–47.
- [3] Philippe G, Michel G, Patrick S, Serge B. Solving the frequency assignment problem with polarization by local search and tabu. *4OR: A Quarterly Journal of Operations Research*, 2005,3(1):59–78. doi:10.1007/s10288-004-0056-4
- [4] Won-Young S, Soo YC, Jaewwook L, Chi-Hyuck J. Frequency insertion strategy for channel assignment problem. *Wireless Networks*, 2006,12(1):45–52.
- [5] Liu J, Zhong WC, Jiao LC. A multiagent evolutionary algorithm for constraint satisfaction problems. *IEEE Trans. on Systems, Man, and Cybernetics, Part B*, 2006,36(1):54–73.
- [6] Zhong WC, Liu J, Xue MZ, Jiao LC. A multiagent genetic algorithm for global numerical optimization. *IEEE Trans. on Systems, Man, and Cybernetics (Part B: Cybernetics)*, 2004,34(2):1128–1141.
- [7] Anderson LG. A simulation study of some dynamic channel assignment algorithm in a high capacity mobile telecommunication system. *IEEE Trans. on Communications*, 1973,COM-21(11):1294–1301.
- [8] Hale WK. Frequency assignment: Theory and applications. *Proc. of the IEEE*, 1980,68(12):1497–1513.

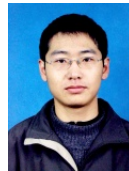
- [9] Cao XB, Luo WJ, Wang XF. A co-evolution pattern based on ecological population competition model. *Journal of Software*, 2001, 12(4):556–562 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/12/556.htm>
- [10] Jiao LC, Liu J, Zhong WC. *Co-Evolution Computation and Multiagent System*. Beijing: Science Press, 2006 (in Chinese).
- [11] Janssen J, Kilakos K. An optimal solution to the “Philadelphia” channel assignment problem. *IEEE Trans. on Vehicular Technology*, 1999,48(3):1012–1014.
- [12] Hurley S, Smith DH, Thiel SU. A system for discrete channel frequency assignment. *Radio Science*, 1997,32(5):1921–1939.
- [13] Sung CW, Wong WS. Sequential packing algorithm for channel assignment under cochannel and adjacent channel interference constraint. *IEEE Trans. on Vehicular Technology*, 1997,46(3):676–685.
- [14] Hellebrandt M, Heller H. A new heuristic method for frequency assignment. Technical Report, TD(00)003, Valencia, 2000.
- [15] Avenali A, Mannino M, Sassano A. Minimizing the span of d -walks to compute optimum frequency assignments. *Mathematical Programming*, 2002,91(2):357–374.
- [16] Valenzuela C, Hurley S, Smith DH. A permutation based genetic algorithm for minimum span frequency assignment. In: Eiben AE, Bäck T, Schoenauer M, Schwefel HP, eds. *Proc. of the 5th Int’l Conf. on Parallel Problem Solving from Nature*. LNCS 1498, London: Springer-Verlag, 1998. 907–916.
- [17] Allen SM, Hurley S, Smith DH, Thiel SU. Using lower bounds in minimum span frequency assignment. In: Voss S, *et al.*, eds. *Meta-Heuristics: Advances and Trends in Local Search Paradigms for Optimization*. Boston: Kluwer Academic Publishers, 1999. 191–204.
- [18] Matsui S, Tokoro K. Improving the performance of a genetic algorithm for minimum span frequency assignment problem with an adaptive mutation rate and a new initialization method. In: Spector L, Goodman ED, eds. *Proc. of the GECCO 2001 (Genetic and Evolutionary Computation Conf.)*. San Francisco: Morgan Kaufmann Publishers, 2001. 1359–1366.

附中文参考文献:

- [9] 曹先彬,罗文坚,王煦法.基于生态种群竞争模型的协同进化.软件学报,2001,12(4):556–562. <http://www.jos.org.cn/1000-9825/12/556.htm>
- [10] 焦李成,刘静,钟伟才.协同进化计算与多智能体系统.北京:科学出版社,2006.



李瑾姝(1982—),女,内蒙古呼和浩特人,硕士,主要研究领域为进化算法,多智能体系统.



胡康(1982—),男,硕士,主要研究领域为进化计算,调度算法.



刘静(1977—),女,博士,副教授,CCF 高级会员,主要研究领域为进化计算,多智能体系统,数据挖掘,图像处理.



王景润(1982—),男,硕士,主要研究领域为进化计算,VLSI 布图规划.



焦李成(1959—),男,博士,教授,博士生导师,CCF 高级会员,主要研究领域为自然计算,智能信息处理.