

# 基于社群联盟的冲突消解原则求解图着色问题

郑皎凌<sup>1</sup>, 舒红平<sup>1</sup>, 许源平<sup>1</sup>, 乔少杰<sup>2</sup>, 文立玉<sup>1</sup>

(1. 成都信息工程大学软件工程系 成都 610225;

2. 西南交通大学信息科学与技术学院 成都 610031)

**【摘要】** 该文提出了一种基于群体协作的计算模型。该模型首先将输入的数据单元建模成微观个体, 然后基于求解目标设计个体间的协作规则, 最后通过个体在协作过程中涌现出的宏观现象来得到全局最优解。通过运用群体协作模型求解具有NP-完全复杂度的最优图着色问题, 结果表明该模型的性能优于若干启发式方法, 并且得到如下结论: 1) 如果算法的动力学特征类似于混沌边缘现象, 则算法能够在线性或亚线性时间复杂度求解问题。2) 如果算法的动力学特征呈现出完全随机性或强收敛性, 则算法将退化成蛮力搜索。

**关键词** 协作规则; 涌现计算; 图着色; 群体协作; NP-完全; 社会计算

中图分类号 TP311

文献标志码 A

doi:10.3969/j.issn.1001-0548.2016.01.001

## Solving Graph Coloring Problem Based on Conflict Resolution Strategies of Social Community Alliance

ZHENG Jiao-ling<sup>1</sup>, SHU Hong-ping<sup>1</sup>, XU Yuan-ping<sup>1</sup>, QIAO Shao-jie<sup>2</sup>, and WEN Li-yu<sup>1</sup>

(1. Department of Software Engineering, Chengdu University of Information Technology Chengdu 610225;

2. School of Information Science and Technology, Southwest Jiaotong University Chengdu 610031)

**Abstract** This study proposes a computing model based on group cooperation. The data unit of the input is first modeled as group individual, the cooperation rules between individuals are then designed based on the target of the problem. Finally, the problem's global optimal solution is obtained through the emergence phenomenon of individuals' cooperation process. By applying group cooperation model to solve graph coloring problem which has NP-complete complexity, it shows that the proposed model works better than several heuristic methods. Experimental results illustrate that: 1) if the algorithm's dynamics exhibits the edge of chaos phenomenon, the algorithm can solve the problem in linear or sub linear time complexity; 2) if the algorithm's dynamics is completely random or exhibits strong convergence, the algorithm degenerates into brutal force search.

**Key words** cooperation rules; emergent computing; graph coloring; group cooperation; NP-complete; social computing

群体计算是目前研究界和产业界关注的热点。很多社会学和生物学方面的发现都证实了群体计算的巨大潜力。著名的荷兰崔克坦市交通实验<sup>[1]</sup>显示了在没有宏观交通管理机制的情况下, 个体机动车辆能够通过相互协作来保持良好的交通状况。维基百科网站的迅速发展显示了在松散管理机制下互联网个体进行大规模协作的潜力<sup>[2]</sup>。另一方面, 计算理论研究者正尝试通过群体协作机制来解决计算机科学领域的难题。如MIT尝试通过大量肥皂泡之间的自组织来求解Steiner树问题<sup>[3]</sup>。基于群体参与的互联网游戏Foldit基于大量游戏者的相互协作来设计

蛋白质折叠结构, 结果通过互联网游戏所设计出的蛋白质结构有效性超过了生物专家的设计结果<sup>[4]</sup>。

本文通过对图着色问题的求解设计基于群体协作规则的算法, 并进一步探索在不同协作规则指导下所产生的微观个体动力学特征与整个群体宏观特征之间的内在联系, 以及不同的微观, 宏观特征与算法复杂度之间的内在联系。

## 1 问题归约

### 1.1 为什么将图着色问题归约成社群联盟问题

计算理论中可归约性理论已经证明, 如果问题A可归约到问题B, 且B是可判定的, 那么A也是可判

收稿日期: 2014-09-30; 修回日期: 2015-03-27

基金项目: 国家自然科学基金(61202250, 61203172); 四川省教育厅重点项目(ZA150184)

作者简介: 郑皎凌(1981-), 女, 博士, 副教授, 主要从事社会计算、复杂系统、数据挖掘等方面的研究。

定的。当问题 $A$ 具有NP-难或NP-完全计算复杂度, 而问题 $B$ 具有线性或亚线性复杂度, 如果能够将问题 $A$ 很好的归约成问题 $B$ , 那么问题 $A$ 的复杂度就可大大降低。

所以, 基于上述归约思路来解决问题 $A$ 不是设计出直接针对问题 $A$ 特性的协作规则, 而是需要首先找到与问题 $A$ 相似的问题 $B$ , 通过归约问题 $B$ 的求解过程来解决问题 $A$ 。这里存在以下两个难点。首先, 需要找到社会或生物界中与问题 $A$ 相似的问题 $B$ ; 其次, 问题 $B$ 的计算复杂度应当是已知的, 并且远远小于 $A$ 的计算复杂度。

下面的几个小节首先阐述了图着色问题和社群联盟问题的相似性, 然后给出了恰当的协作规则能够快速求解社群联盟问题的理论依据, 最后阐述了将图着色问题归约成社群联盟问题的宏观过程。

## 1.2 图着色问题和社群联盟问题的相似性

**最优图着色问题:** 设 $G(V, E)$ 是一个无向图,  $V$ 是 $G$ 的节点集合,  $E$ 是 $G$ 的边集合。图着色问题是将 $G$ 中的每个节点都着上颜色, 使得任意相邻的两个节点的颜色不同, 最优图着色问题是使图的着色数最小。

**最优社群联盟问题:** 为了保存自身或抗衡外部, 社会中各个行动者(行动者可以是一个企业、集团、国家等)根据彼此之间的友好或敌对关系形成联盟。最优社群联盟的目标是使得最终形成的联盟结构是纳什均衡的, 并使得整个联盟的负能量最小。纳什均衡联盟结构指社群联盟中没有任何单个行动者有动机从当前所在联盟转到另一个联盟。联盟负能量是根据联盟中成员之间的敌对关系来决定的。如果一个联盟中两个行动者彼此敌对, 那么联盟中的负能量大。

首先, 两个问题具有相似数据结构。两个问题中的基本元素都是单个的个体, 图着色问题中的节点可以看成社群联盟问题中的单个社群, 图着色问题中节点之间关系可以看成社群联盟问题中社群之间的关系, 如可以将存在连边的两个节点看作具有敌对关系的两个社群, 这两个节点不能着一种颜色类比为两个敌对社群不能存在于同一个社群联盟。

其次, 两个问题具有相似的最优化目标。最优图着色问题旨在形成不同的颜色团, 并且颜色团中任意一个或多个节点无法变更到其他颜色团以使得颜色团的数量进一步减小, 该目标可以类比为最优社群联盟问题旨在形成具有纳什均衡的社群联盟, 使得联盟中没有单个或多个行动者有动机从当前所在联盟转到另一个联盟去, 并且联盟的负能量最小。

所以, 最优图着色问题中达到图着色数最小的目标可以类比为最优社群联盟问题中达到联盟负能量最小的目标。

## 1.3 社群联盟问题能够被快速求解的依据

文献[5]详细分析了为什么在纷繁的社会生活中个体的行为无法预知, 但当个体数量达到一定程度时, 群体的行为反而表现得有章可循, 于杂乱中显现秩序和稳定。其中, 提出了若干达到最优社群联盟的协作规则, 如果社群个体按照规则进行协作, 那么会在较短的时间步长内形成具有纳什均衡, 并且负能量最小的联盟格局。

## 1.4 将图着色问题归约成社群联盟问题的过程

首先是两个问题数据结构的归约: 将图的节点集合看成进行协作的社群, 将图的单个节点和边看作群体中的个体和个体之间的关系。

其次是传统搜索算法向群体协作算法的归约: 传统搜索算法将问题的解看成一个巨大的搜索空间, 通过设计不同搜索算法在解空间上进行各种遍历来得到问题的解。基于群体协作的算法不设计搜索过程, 而是设计协作规则。具有不同颜色的节点群体可通过协作规则自由合并或拆分, 最终获得解的过程不是通过搜索得到, 而是通过群体中的个体根据协作规则进行交互所涌现出的自组织现象得到。

设要对图1进行最优图着色, 例1描述了一个简单的基于群体协作来求解图1的图着色问题的过程。

**例1:** 首先是数据归约: 对 $G$ 进行初始化着色, 图1中的节点1、5、8着色第一种颜色; 节点2、6、9着色第二种颜色; 节点3、7着色为第三种颜色; 节点4、10着第四种颜色。故节点分成了4个不同的群体, 每个节点群体中的节点着相同颜色, 且互不邻。颜色团1~4可以看成4个不同的社群联盟。

其次, 是将得到最优社群联盟的协作规则转化成图着色问题中各个节点的协作规则。文献[5]提到, 要形成具有最小负能量的联盟, 不一定需要高瞻远瞩的理性决策, 只需要每个参与者从获得局部改善的角度出发, 便可以实现同一过程。

根据上述原则, 本文设计如下的图节点之间的简单协作规则: 1) 考虑单个节点能否在颜色团之间移动, 设计规则为当一个颜色团中不包含该节点的邻居时, 则将该节点移动到这个颜色团中; 2) 考虑两个节点能否在颜色团之间移动, 设计规则为当两个节点各自所在的颜色团中不包含对方节点的邻居时, 则将该这两个节点在两个颜色团中对调。

基于规则1和规则2的求解图的最优着色数可以通过对节点及其所在颜色团反复运行上述协作规则来得到，具体如下：1) 运用规则1将颜色团4中的节点4移动到颜色团1，颜色团4中的节点10移动到颜色团2，得到着色格局2；2) 运用规则2将颜色团1中的节点1和颜色团3中的节点3对调，运用规则2将颜色团2中的节点9和颜色团3中的节点7对调，得到着色格局3；3) 运用规则2将颜色团1中的节点8和颜色团3中的节点9对调，运用规则2将颜色团2中的节点2和颜色团3中的节点1对调，得到着色格局4；4) 运用规则1将颜色团3中的节点2移动到颜色团1，颜色团3中的节点8移动到颜色团2，得到着色格局5。此时，得到图的最优着色数为2。

### 1.5 群体协作模型与启发式模型的区别

如果将算法本身看作是一个动力学系统，对于图着色问题，该动力学系统就是不断地选择不同的节点尝试改变其颜色。虽然群体协作算法模型与启发式算法模型中都有规则的概念，这使得二者具有一定的相似性，但二者对图中节点所产生的动力学效果却非常不同。在5.2节中详细分析了启发式规则与群体协作规则在指导节点改变颜色的过程中所形成的完全不同的动力学过程，正是由于二者对个体节点在微观上造成的不同动力学过程，导致了协作算法模型与启发式算法在宏观上非常不同的求解速度。

## 2 相关工作

本节将从群体协作的实际效果，群体协作如何被运用到计算机科学问题的求解以及图着色问题研究现状三方面介绍相关的研究成果。

大规模协作现象广泛存在于人类社会生产生活的各个方面，许多研究者认为具有协作功能的群体比不具有协作功能的群体能够更好地解决问题。文献[6]认为如果社群之间的沟通能力能够发挥到极致，那么人类将变得聪明许多，因此提出了思想的梅特卡夫定律。文献[7]认为具有协作关系的社群更能适应城市生存和具有复杂社会网络等现代化特征的社会生活。文献[8]认为通过群体协商所作出的决策往往优于每一个成员单独作出的决策，并且认为只有集体智慧才能塑造成功的商业、经济、社会与民族。文献[9]认为人群之间的合作可以在彼此没有任何朋友关系，甚至在具有敌对关系的群体中产生，而具有协作关系的群体的生存能力要优于存在激烈竞争的群体。文献[10]认为具有协作关系的群体在响

应速度、应急处理等方面要优于基于集中独裁式管理机制的群体。文献[11-12]发现对社群进行宏观干预会影响社群中个体的生产效率。文献[13-15]提出了收集社群中个体所产生的海量数据的方法。文献[16-18]研究了社群中的个体之间在电话、短信等方面的交互模式，发现社群中的大量个体会重复若干简单的交互模式。

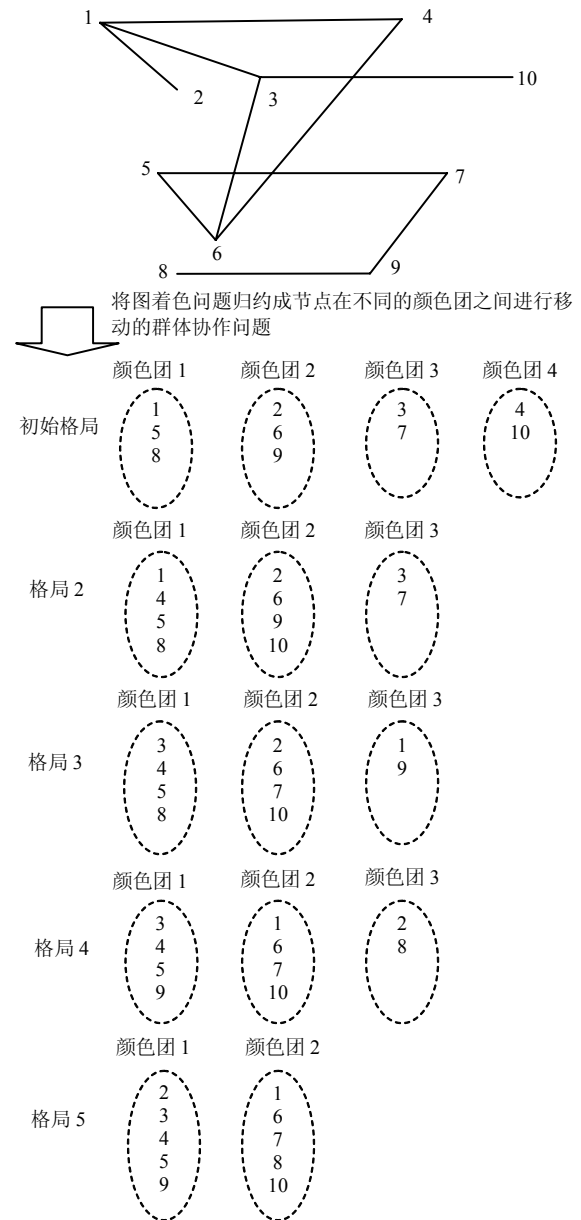


图1 将图着色问题归约成社群联盟问题

由于传统搜索算法很难克服数据规模增长所带来的“组合爆炸”问题，很多研究者尝试将原问题归约成群体协作问题，通过群体协作产生的涌现现象来得到问题的解。文献[19]通过基于自组织协作的方法来解决N-皇后问题，取得了很好的效果。文献[3]

通过大量肥皂泡之间符合物理规律的协作过程来解决Steiner tree问题, 该问题具有NP-难复杂度。文献[4]通过设计出一款名为Foldit的游戏来让玩家协作探索蛋白质的折叠结构, 并且得到了生物学专家都难以得到的好结果。维基百科等大规模协作网站及伴随其涌现出的高级群体智能, 促进了包括“人类认知组”计划在内的对于社群大规模认知规律的研究及应用。文献[20]从维基百科中人们对词条的协作编辑规律来训练计算语言学模型。

图着色问题在调度排产、教室安排等许多方面有广泛应用, 但由于图着色是具有NP-难复杂度的组合优化问题, 目前多数算法不是找出它的精确最优解, 而是力求快速找到问题的近似最优解。所以, 多数算法采用了基于启发式规则的方法来搜索问题的解。增强SEQ算法和DSATUR算法是两种传统的直接启发式方法<sup>[21-22]</sup>。增强SEQ算法的基本思想是按照图中结点度由大到小的次序逐个给结点着色, 着色时使用的颜色是编号最小的可用颜色。DSATUR算法的基本思路是每次挑选那个其邻域内已用颜色数目最多的结点进行着色, 着色时使用的颜色是编号最小的可用颜色。在此基础上又提出了一些较为复杂的启发式方法, 如禁忌搜索<sup>[23]</sup>、模拟退火<sup>[24]</sup>及遗传算法<sup>[25]</sup>等。

### 3 图着色节点移动规则

#### 3.1 节点移动规则

例1给出了如何将群体协作原则过程归约成图着色问题中个体节点的移动规则, 本节设计了5种节点移动规则, 并且在本节最后总结了这5种规则是如何从群体协作原则归约而来的。

为了清楚地描述协作规则, 表1给出了必要的符号及其解释。本文把将节点颜色变成另一个节点颜色的操作称为将节点从其原来所在的颜色团移动到另一个颜色团; 把将节点颜色变成一种新的颜色的操作称为增加一个新的颜色团, 并将节点移动到这个新的颜色团中。

**规则 1** 颜色团中单个节点的移动规则

规则1设整个图的着色数为 $K$ , 如果Node  $i$ 的邻居集合  $Neighbors(i)$  的着色数  $|Color\_Set(Neighbors(i))| < K-1$ , 那么就可以将Node  $i$ 移动到不包含其邻居节点的颜色团中。规则1的算法形式如下。

规则1: Rule1(Node  $i$ )

输入: 图 $G(V,E)$ , 每个节点的颜色

输出: Node  $i$ 的新颜色

```

1  If ( $|Color\_Set(Neighbors(i))| < K-1$  )
2  {
3    Candidate_color_set = Color_Set( $V$ ) -
      Color_Set(Neighbors( $i$ )) - Color( $i$ );
4    Color = one-of (Candidate_color_set);
5    set_color( node( $i$ ), color );
6  }
```

如果Node  $i$ 的邻居节点没有分布于所有其他颜色团中(第1行), 则首先找出候选颜色集Candidate\_color\_set(第3行), 候选颜色集是指不包含节点 $i$ 所有邻居节点的颜色, 也不包含节点 $i$ 自身颜色的所有其他颜色集合, 然后将Node  $i$ 着色为Candidate\_color\_set中的任意一种颜色(第5行)。下面通过例2来对规则1进行说明。

表1 符号描述

符号	含义
$G(V,E)$	无向图 $G$ , $V$ 是节点集合, $E$ 是边集合
$e(i,j)$	$e(i,j)=1$ , 如果 $i, j$ 相邻, 否则 $e(i,j)=0$
Color( $i$ )	节点 $i$ 所着的颜色
Color_Set( $V$ )	节点集合 $V$ 中所有节点所着颜色的集合, $Color\_Set(V) = \{Color(i)   i \in V\}$
Node_Set( $k$ )	着第 $k$ 种颜色的节点集合, $Node\_Set(k) = \{i   i \in V \text{ 并且 } Color(i) = k\}$
Neighbors( $i$ )	节点 $i$ 的邻居节点集合 $Neighbors(i) = \{j   e(i,j) = 1\}$
Neighbors( $i,k$ )	节点 $i$ 的邻居节点中所有着第 $k$ 种颜色的节点集合, $Neighbors(i,k) = \{j   e(i,j) = 1 \text{ 并且 } Color(j) = k\}$
Color_Set(Neighbors( $i$ ))	节点 $i$ 的邻居节点集合所着颜色的集合, $Color\_Set(Neighbors(i)) = \{Color(j)   e(i,j) = 1\}$
Neighbors(Node_Set( $k$ ))	着第 $k$ 种颜色的节点集合的邻居集合, $Neighbors(Node\_Set(k)) = \{j   e(i,j) = 1 \text{ 并且 } i \in Node\_Set(k)\}$

例2: 设图1中图 $G$ 的节点初始着色格局如图2所示, 即颜色团1={1,5,8}, 颜色团2={2,6,9}, 颜色团3={3,7}, 颜色团4={4,10}。图2描述了使用规则1对颜色团1中的节点5和节点8进行移动的过程。因为颜色团3, 4不含节点8的邻居节点, 颜色团4不含节点5的邻居节点, 此时可以将节点8移动到颜色团3, 将节点5移动到颜色团4。

**规则 2** 颜色团中所有节点的整体移动规则

规则2和规则1类似, 但使用条件更加严格。规则1只移动单个节点到其他颜色团, 规则2要求只有当给定颜色团中所有节点都能移动到其他颜色团时才能使用, 所以使用一次规则2就能使图的着色数减少1, 下面给出了规则2的算法形式。

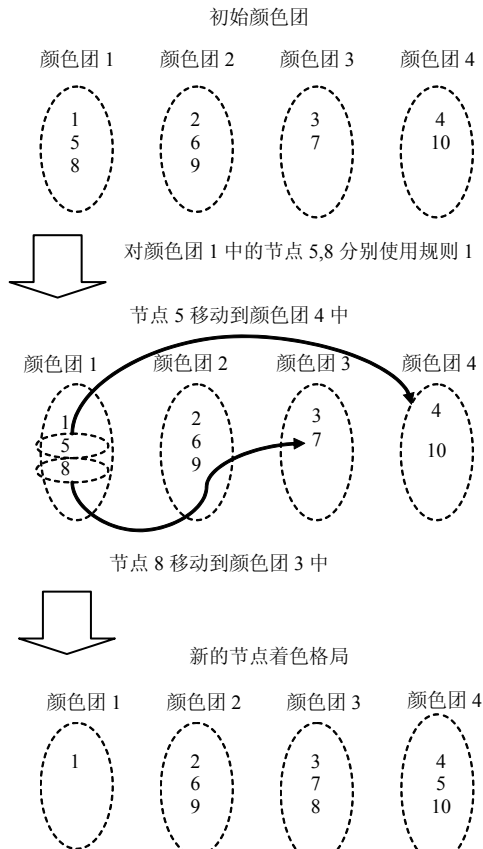


图2 对颜色团1中的节点5,8分别使用规则1

规则2: Rule2(Node\_Set( $k$ ))

输入: 图 $G(V,E)$ , 每个节点的颜色

输出: Node\_Set( $k$ )中的各节点的新颜色

```

1  Foreach node  $i$  in Node_Set( $k$ )
2  {
3    Rule1(Node  $i$ );
4  }
5  If (Node_set( $k$ )所有节点都能使用Rule1)
6    Do Nothing;
7  Else Rollback;

```

第1~4行表示对Node\_Set中所有节点依次执行Rule1。第5~7行表示行表示规则2是对颜色团进行整体移动, 即要么将颜色团中的所有节点都移动到其他颜色团, 要么颜色团中的所有节点都不移动, 如果其中有一个节点无法按照规则1进行移动, 则对已经进行移动的其他节点进行回滚操作, 使得该颜色团中已经被使用了规则1的节点的颜色变成使用规则1之前的颜色。

**规则 3** 颜色团中单个节点与其邻居节点集合的协作移动规则

图2中颜色团分布情况为颜色团1={1}, 颜色团

2={2,6,9}, 颜色团3={3,7,8}, 颜色团4={4,5,10}。在这种着色格局下, 无法使用规则1或规则2对颜色团1进行压缩, 因为节点1的邻居集合{2,3,4}分别分布在除了颜色团1之外的所有其他颜色团中。此时可以考虑使用规则3来移动节点1。

规则3对Node  $i$ , 如果能够直接对其使用规则1, 则对Node  $i$ 使用规则1; 如果无法对Node  $i$ 使用规则1, 也即Color\_Set(Neighbors( $i$ ))等于 $K-1$ , 此时无法直接移动该节点到其他颜色团, 此时可以先压缩该节点的邻居节点集合, 然后再将该节点移动到不包含其邻居节点的颜色团中。规则3的算法形式如下。

规则3: Rule3(Node  $i$ )

输入: 图 $G(V,E)$ , 每个节点的颜色

输出: Node  $i$ 及Neighbors( $i$ )的颜色

```

1  If (能够对Node  $i$ 使用Rule 1) Rule1(Node  $i$ );
2  Else
3  {
4    Foreach Color  $j$  in Color_Set(Neighbors( $i$ ))
5    { If (能够对Neighbors( $i,j$ )使用Rule2)
6      {
7        Rule2(Neighbors( $i,j$ ));
8        将Node  $i$ 移动到颜色团 $j$ 中;
9        Break;
10   } } }

```

第1行表示如果节点能够直接移动到其他颜色团中, 则直接移动该节点, 第2~10行表示如果节点不能直接移动到其他颜色团中, 则尝试先使用规则2压缩该节点的邻居节点集合, 然后将该节点移动到不包含其邻居节点的颜色团中。4~10行表示依次压缩Node  $i$ 邻居节点集合中着同一种颜色 $j$ 的节点集合Neighbors( $i,j$ ), 如果Neighbors( $i,j$ )是可压缩的, 则压缩Neighbors( $i,j$ )并将Node  $i$ 移动到不包含的颜色团 $j$ 中, 因为此时颜色团 $j$ 中已经没有Node  $i$ 的邻居节点。

例3: 图2中图 $G$ 的最终着色格局为颜色团1={1}, 颜色团2={2,6,9}, 颜色团3={3,7,8}, 颜色团4={4,5,10}。此时, 继续采用规则1无法将颜色团中的节点1移动到其他任意一个颜色团中, 可以采用规则3移动颜色团1中的节点1, 如图3所示。首先, 压缩节点1着第2种颜色的邻居集合, 即节点集合{2}, 将节点2移到颜色团3中, 此时节点1的邻居集合就只分布在颜色团3和4中; 然后, 将节点1移动到颜色团2中。如图3所示, 当节点1移动到颜色团2中后, 颜色团1就消失了, 图的着色数从4降低到3。

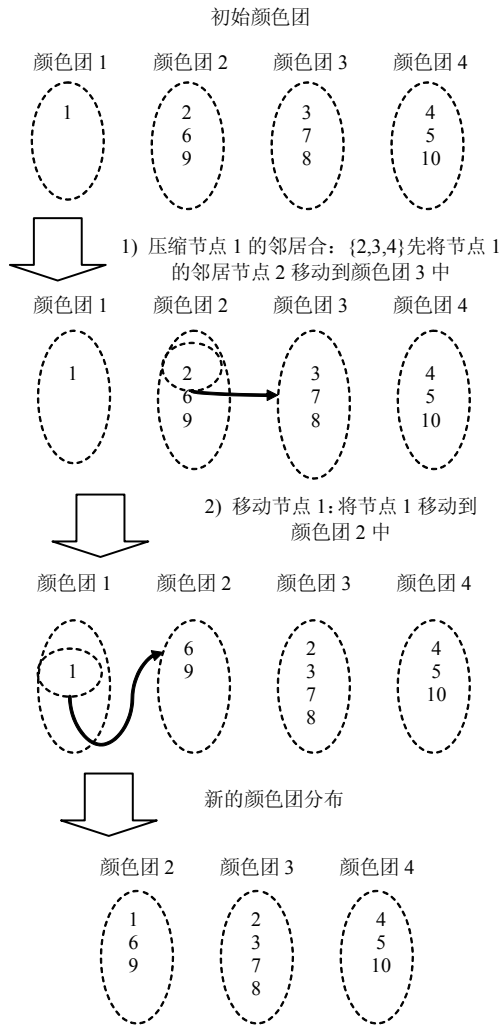


图3 对颜色团1中的节点1使用规则3

**规则 4** 颜色团中所有节点与其邻居节点集合的协作移动规则

规则4和规则3类似, 但使用条件更加严格。规则3只移动单个节点到其他颜色团, 规则4则要求当给定颜色团中所有节点都可以移动到其他颜色团时才使用, 所以使用一次规则4一定能使图的着色数减少1, 下面给出了规则4的算法形式。

规则4与规则2很相似, 与规则2不同之处在于, 规则4使用规则3来移动给定颜色团中的每个节点, 而规则2使用规则1来移动给定颜色团中的每个节点。第5~6行表示如果Node\_Set中的所有节点都能使用规则4, 则不进行任何后续操作, 如果Node\_Set中存在至少1个节点不能使用规则4, 所有对Node\_Set中其他节点进行的移动操作都会被撤销。

规则4: Rule4(Node\_Set(k))

输入: 图 $G(V,E)$ , 每个节点的颜色

输出: 颜色团 Node\_Set(k) 的节点颜色及

Node\_Set(k)的邻居节点集合Neighbors (Node\_Set(k)) 的节点颜色

```

1  Foreach node  $i$  in Node_Set(k)
2  {
3      Rule3(Node  $i$ );
4  }
5  If (Node_set(k)所有节点都能使用Rule3)
6      Do Nothing;
7  Else
8      Rollback;

```

**规则 5** 随机拆分颜色团规则

规则5对给定的颜色团进行拆分, 即随机地对颜色团中的节点着上新的颜色, 故当运行规则5之后, 整个图的着色数会增加。其主要目的是与规则2、4配合使用, 因为当图的着色数越少时, 能够对颜色团或颜色团中的节点继续使用规则2、4的机会就越小, 为了避免图着色格局陷入了局部最优, 可以采用规则5来重新调整图的颜色团的分布情况, 使得规则2、4可以继续使用。下面给出了规则5的算法形式。

规则5: Rule5(Node\_set(k))

输入: 图 $G(V,E)$ , Node\_Set(k)中每个节点的颜色

输出: 使用规则5后Node\_Set(k)中每个节点的颜色

```

1  Foreach node  $i$  in Node_Set(k)
2  {   Max_color = Max(Color_Set(V));
3      Rand = random(0,1);
4      If(Rand < T)
5          {   set_color( node( $i$ ), Max_color + 1);
6              } }

```

第2行表示取得整个图的颜色标号中的最大值, 第4行中的 $T$ 是一个阈值, 当所产生的随机数Rand小于该阈值时, 规则5将Node  $i$ 着上新的颜色, 新的颜色的标号是当前颜色标号中的最大值加1。 $T$ 越大, 将会有越多的节点被着上新的颜色, 也即整个图的着色数或颜色团数量将越多。例6对规则5进行了举例说明。

例4: 设图 $G$ 中节点的着色格局如图4所示, 即颜色团颜色团2={1,6,9}, 颜色团3={2,3,7,8}, 颜色团4={4,5,10}。当采用规则5来对图 $G$ 的节点随机重新着色后, 图的着色格局就变为6个颜色团。

### 3.2 节点移动规则与社群联盟规则的关系

上面给出了5种求解最优图着色的节点协作规则, 下面将进一步分析规则1~4与社群联盟协作原则之间的关系。

社会学研究表明,如果社群成员能够按照恰当的协作规则行动,那么会在较短的时间步长内形成具有纳什均衡特征,并且负能量最小的联盟格局。而具有最小负能量的联盟格局正好可以对应图着色问题中的具有最少颜色数的着色格局。文献[5]给出了两条社群联盟中个体见的协作原则:1)实现负的下降,不一定需要高瞻远瞩的理性决策,只需要每个参与者从获得局部改善的角度出发,便可以实现同一过程;2)对联盟进行的调整是通过各个参与者的渐近行动来实现,各个参与者不会联合起来去改变现有的联盟格局。其中,原则1指出联盟中的个体仅根据自身当前的局部信息行动,便可以从宏观上达到降低联盟负能量的作用。

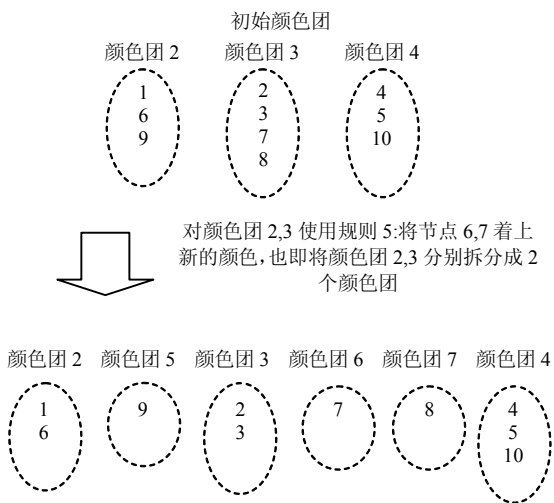


图4 对颜色团2,3使用规则5

根据原则1设计出了图着色问题中个体间的协作规则1和3,两条规则都是让图中的节点根据自身的局部信息来进行移动,规则1和3的不同之处在于规则1中个体完全是基于自身状态行动,而规则3中个体会考虑其邻居节点的状态来行动。

原则2指出联盟中的个体应当回避多个个体共同行动的行为方式。由于原则2是一种回避性准则,无法设计出与之直接相对应规则。所以,为了将原则2转换成可以直接运行的节点移动规则,本文设计出与原则2相反的图着色的协作规则2和4,这两条规则严格要求一个颜色团中的所有个体共同行动。设计这两条规则的目的在于考察如果不按照社群联盟协作原则来设计图着色问题的协作规则,基于该规则进行协作的群体效率是否会有所下降,在第5节的实验中证实了这个结论。

## 4 基于协作规则的演化算法

### 4.1 算法框架

输入:无向图 $G(V,E)$ ,使用的协作规则Rule

输出:图 $G$ 的着色数 $K$

```

1 Color_Set = Initialize( $G$ );
2  $K = |Color\_Set|$ ;
3 while( $K >$  图 $G$ 的最优着色数)
4     Evolution(Rule);
5 End While
6 stop

```

第1行表示对图 $G$ 的颜色团进行初始化,初始化方法就是对每个节点着一种颜色,故最初 $K=|V|$ 。行3~行5表示当图的着色数 $K$ 大于最优着色数时,对 $G$ 中的各个颜色团或颜色团中的各个节点循环执行不同的协作规则,直到找到最优解。第4行Evolution(Rule)中的Rule表示第3节中的规则1~规则4,其中规则2、4需要配合规则5来使用,下面给出了Evolution函数中4种规则的运行过程。

Evolution(Rule 1)

```

1 ForEach Color in Color_Set( $V$ )
2     ForEach Node  $i$  in Node_Set( $G, Color$ )
3         Rule 1(Node  $i$ )//对每个颜色团中的
// 每个节点依次执行规则1
4     End For
5 End For

```

Evolution(Rule 2)

```

1 ForEach Color in Color_Set( $V$ )
2     Rule 2(Node_Set( $G, Color$ ))
//对每个颜色团的节点集合执行规则2
3     Rule 5(Node_Set( $G, Color$ ))
//对每个颜色团的节点集合执行规则5
4     End For
5 End For

```

Evolution(Rule 3)

```

1 ForEach Color in Color_Set( $V$ )
2     ForEach Node  $i$  in Node_Set( $G, Color$ )
3         Rule 3(Node  $i$ )//对每个颜色团中的
// 每个节点依次执行规则3
4     End For
5 End For

```

Evolution(Rule 4)

- 1 Foreach Color in Color\_Set( $V$ )
- 2 Rule 4(Node\_Set( $G$ , Color))  
//对每个颜色团的节点集合执行规则4
- 3 Rule 5(Node\_Set( $G$ , Color))  
//对每个颜色团的节点集合执行规则5
- 4 End For
- 5 End For

## 4.2 理论分析

本小节理论分析的目的旨在证明按照社群联盟协作原则所设计出的节点移动规则能够高效求解出对应的计算机问题, 而没有按照社群联盟协作原则所设计出的群体协作规则无法高效求解问题。

3.1节中的规则1、3是社群联盟协作原则1所鼓励的个体行为模式, 3.1节中的规则2、4是社群联盟协作原则1所回避的个体行为模式。下面的理论分析证明了在规则1、3指导下得到图最优着色数的概率要优于规则2、4。

**引理 1** 设有无向图 $G(V, E)$ , node  $i$ 和node  $j$ 是 $G$ 中的任意两个节点, 设Neighbors(node  $i$ )和Neighbors(node  $j$ )是两个节点的邻居集合, 且Neighbors(node  $i$ )和Neighbors(node  $j$ )的着色数都是可压缩的, 设 $\Delta$ 是对Neighbors(node  $i$ )的着色数进行压缩后, 仍然可以对Neighbors(node  $j$ )的着色数进行压缩的概率, 则 $\Delta$ 正比于图 $G$ 中所有节点度的平均值。

证明: 设图5的左右两个椭圆区域表示Neighbors(node  $i$ )和Neighbors(node  $j$ ), 其中区域II表示Neighbors(node  $i$ )和Neighbors(node  $j$ )的交集。对Neighbors(node  $i$ )的着色数进行压缩实际上是将着不同颜色的多个节点变成只着一种颜色。不失一般性, 设node  $a$ 和node  $b$ 是Neighbors(node  $i$ )中的任意两个节点,  $a$ 和 $b$ 着不同的颜色, 对Neighbors(node  $i$ )的着色数进行压缩包含如下4种情况: 1)  $a$ 、 $b$ 都属于区域I, 此时, 将node  $a$ 的颜色变成node  $b$ 的颜色不会影响对Neighbors(node  $j$ )着色数的压缩; 2)  $a$ 属于区域I,  $b$ 属于区域II, 因为node  $b$ 也属于Neighbors(node  $j$ ), 将node  $b$ 的颜色变成node  $a$ 的颜色只可能使Neighbors(node  $j$ )的着色数不变或增加, 从而造成Neighbors(node  $j$ )着色数无法继续压缩; 3)  $a$ 属于区域II,  $b$ 属于区域I, 因为node  $b$ 不属于Neighbors(node  $j$ ), 故将node  $b$ 的颜色变成node  $a$ 的颜色不会影响对Neighbors(node  $j$ )着色数的压缩; 4)  $a$ 、 $b$ 都属于区域II, 所以 $a$ 、 $b$ 也属于Neighbors(node  $j$ ),

将node  $b$ 的颜色变成node  $a$ 的颜色只可能会使Neighbors(node  $j$ )的着色数不变或减小, 所以这种情况不会影响对Neighbors(node  $j$ )着色数压缩。

所以, 只有情况2)可能会影响node  $j$ 的邻居集合着色数的压缩, 且易知 $a$ 、 $b$ 属于区域I或II的概率正比于两个区域的面积, 设 $b$ 属于区域II的概率为 $p$ , 则 $a$ 属于区域I的概率为 $1-p$ , 故二者的联合概率为 $p \times (1-p)$ , 而区域II表示Neighbors(node  $i$ )和Neighbors(node  $j$ )的交集, 易知 $p$ 正比于图 $G$ 节点平均度。当对Neighbors(node  $i$ )的着色数进行压缩后, 仍然可以继续对Neighbors(node  $j$ )着色数进行压缩的概率为 $(1-p \times (1-p))$ , 该多项式与 $p$ 成正比。证毕。

设图 $G$ 当前着色数为 $K$ ,  $P(K-1, \text{Rule } i | K, \text{Rule } i)$ 表示当使用规则 $i$ ( $1 \leq i \leq 4$ )将图着色数从 $K+1$ 降低到 $K$ 后, 再使用规则 $i$ ( $1 \leq i \leq 4$ )将图着色数从 $K$ 降低到 $K-1$ 的概率。定理1, 2证明了规则1、3优于规则2、4。

**定理 1**  $P(K-1, \text{Rule } 3 | K, \text{Rule } 3) > P(K-1, \text{Rule } 4 | K, \text{Rule } 4)$

证明: 设 $m$ 是颜色团中的节点数量,  $m$ 正比于图 $G$ 各个颜色团的平均节点数量,  $\Delta$ 的含义与引理1中相同。不失一般性, 设节点集合 $\{a, b, c, \dots, k\}$ 是图 $G$ 的其中一个颜色团 $i$ , 按照规则4, 如果要压缩颜色团 $i$ , 需要满足如下的连续压缩条件, 即对Neighbors(node  $a$ )的着色数进行压缩后, 仍可以对Neighbors(node  $b$ )的着色数进行压缩; 对Neighbors(node  $b$ )的着色数进行压缩后, 仍可对Neighbors(node  $c$ )着色数进行压缩, 依次类推。根据引理1,  $P(K-1, \text{Rule } 4 | K, \text{Rule } 4)$ 正比于 $\Delta^m$ 。

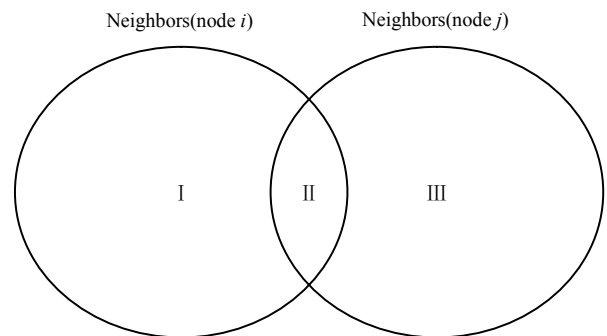


图5 节点 $i$ 和节点 $j$ 的邻居集合分布情况

设图中节点的平均度为 $d$ , 当前着色数为 $K$ , 每个颜色团的节点数为 $m$ , 根据规则3, 颜色团中的单个节点能够在没有其邻居颜色团中自由移动, 如果对每个颜色团中的每个节点都运行一次规则3, 则可以生成不同 $K$ -颜色团的数量正比于 $d \times m \times K$ , 只要其



中有一种颜色团分布能够降低着色数,那么整个图的着色数就会从 $K$ 降低到 $K-1$ ,所以 $P(K-1, \text{Rule 4} | K, \text{Rule 4})$ 正比于 $d \times m \times K \times \Delta^m$ , 证毕。

将定理1结论应用到适应性地形(fitness landscape)理论,在规则4指导下群体只能从一个局部高峰跳到另外若干个局部高峰,且当每个颜色团的节点数 $m$ 越来越大时,可以继续降低着色数的概率正比于 $\Delta^m$ ,群体会迅速收敛于很少几个局部高峰。而在规则3指导下,可继续降低着色数的概率正比于 $(C_N^m - C_N^d)\Delta^m$ ,群体始终拥有一定数量局部高峰可选择,使群体既不会迅速收敛到少数局部高峰也不会大量低矮地形中随机徘徊。

实验5.2节中对不同规则所形成的节点动力学行为进行了分析,发现在规则4的指导下,节点的时间序列会迅速收敛到少数几个节点,而在规则3的指导下,节点的时间序列会呈现出混沌边缘现象,即在随机序列中,总会有一些有秩序和模式的节点序列不断地涌现出来,这与定理1及上述基于适应性地形的分析在本质上是是一致的。

**定理 2**  $P(K-1, \text{Rule 1} | K, \text{Rule 1}) > P(K-1, \text{Rule 2} | K, \text{Rule 2})$

定理2的证明与定理1类似,这里不再赘述。

## 5 实验结果及分析

实验均在 dual-core 3.0 GHz Pentium IV, windows XP, 2 G内存的环境下进行。协作规则和进化算法基于Netlogo仿真平台完成。Netlogo<sup>[26]</sup>平台是一个多主体(multi-agent)可编程模型环境,能够对图中的每个节点设置节点之间的协作规则。实验数据是从文献[27]获得的真实数据,如表2所示。

表2 实验数据集

图编号	图名称	节点数	边数	最优着色数
1	Anna.col	138	986	11
2	mulsol.i.2.col	188	3 885	31
3	zeroin.i.1.col	211	4 100	49
4	fpsol2.i.1.col	496	11 654	65
5	inithx.i.1.col	864	18 707	54
6	le450_5a.col	450	5 714	5
7	le450_5b.col	450	5 734	5
8	le450_5c.col	450	9 803	5
9	le450_5d.col	450	9 757	5

### 5.1 实验结果

表3给出了4种协作规则求解表2中图数据的实验结果,以及采用启发式方法的求解结果,其中基

于禁忌搜索的源码来自文献[28],基于DSATUR搜索的源码来自文献[29]。在下面的分析中,本文认为如果算法的迭代次数与图的节点个数相似,则算法具有线性时间复杂度,如果算法的迭代次数远大于图的节点个数,则算法具有指数时间复杂度。

#### 1) 社群协作规则与传统启发式规则进行比较

社群协作规则优于传统启发式规则。在4种协作规则中Rule3得到了所有图的最优着色数,而两种启发式规则都只得到了部分图的最优着色数,表3c中,禁忌搜索算法只找到图1的最优着色数,而其他几个图都存在冲突节点。DSATUR算法未找到图2、5、6、7的最优着色数。

#### 2) 不同社群协作规则彼此之间的比较

首先,对于相同的数据,不同的协作规则可以有不同的时间复杂度,如对表2中的图2,采用规则1~3求解能够达到线性复杂度,而采用规则4求解则需要指数复杂度;其次,对于相同的协作规则,不同的数据需要不同的时间复杂度进行求解,如规则3对表2中的图1~图5需要线性时间复杂度进行求解,而规则3对表2中的图6~图9需要指数时间复杂度进行求解。

表3a 实验结果

图编号	Rule1			Rule2		
	着色数	迭代次数	时间/s	着色数	迭代次数	时间/s
1	11	150	1	11	319	1
2	31	193	1	31	157	1
3	49	229	1	50	199	1
4	65	477	1	65	5 957	1
5	54	848	1	54	2 509	2
6	12	6 300	270	13	2 022	15
7	13	5 857	248	13	2 100	15
8	11	6 519	280	11	1 310	9
9	13	6 776	290	13	660	5

表3b 实验结果

图编号	Rule3			Rule4		
	着色数	迭代次数	时间/s	着色数	迭代次数	时间/s
1	11	127	1	11	143	1
2	31	156	1	32	2 715	1
3	49	162	1	50	607	1
4	65	431	2	65	1 650	1
5	54	810	8	54	846	2
6	5	5 275	211	12	5 358	32
7	5	6 700	350	11	6 000	43
8	5	3 780	180	8	5 360	32
9	5	5 360	224	9	6 500	50

表3c 实验结果

图编号	禁忌搜索		DSATUR	
	着色数	时间/s	着色数	时间/s
1	11	1	11	1
2	--	9	49	1
3	--	8	49	1
4	--	18	65	1
5	--	15	--	--
6	--	180	9	3
7	--	220	9	3
8	--	190	5	18
9	--	200	5	77

5.2 实验结果的动力学分析

本节旨在基于算法运行过程得到的节点时间序列的动力学特征来分析造成5.1节实验结果的原因, 并在此基础上分析不同的协作规则求解最优图着色的复杂度。由于算法本身可以看成是一个动力学系统, 对于图着色问题, 算法迭代过程的本质就是不断选择不同节点改变其颜色的过程, 故算法动力学过程的输出是一个由节点构成的时间序列, 每个时间点上的节点表示在该时刻改变了颜色的节点。

构造图6、图7两种方式来表达算法动力学过程。图6表示随时间推移依次改变颜色的节点编号, 图7由图6得到, 但去掉了时间维。图7是图6在一个单位时滞上节点之间的先后关联关系, 图7中任一个点(x,y)表示在t时刻改变颜色的节点编号是x, 在t+1时刻改变颜色的节点编号是y。

实验发现, 不同算法的动力学过程主要分为两大类, 一类具有混沌边缘特征, 另一类具有随机性或收敛性特征, 下面进行将详细描述。

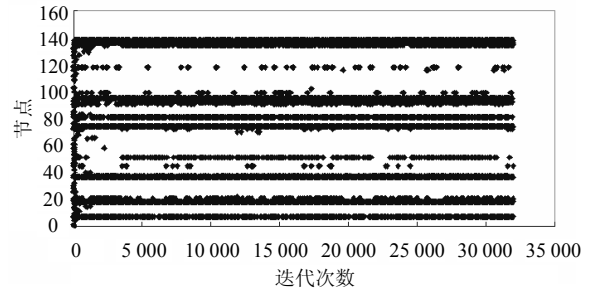
1) 迭代过程具有混沌边缘动力学特征的图

本小节主要针对表2中图1~图5分析在不同的群体协作规则和启发式规则指导下形成的算法动力学特征。首先, 通过图6来分析群体协作规则3与基于禁忌搜索启发式规则的算法动力学区别, 然后, 通过图7来分析基于规则3, 规则1和基于禁忌搜索启发式规则的算法动力学区别。

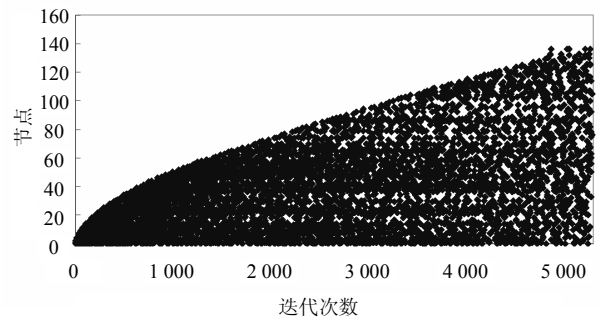
对表2中图1~图5, 如图6所示, 基于禁忌搜索规则的启发式方法在很短时间便收敛到对少数几个点反复地进行颜色改变, 而规则3在整个迭代过程中始终能够更换不同的节点来改变其颜色。

进一步观察规则3的节点动力学过程中0~100号节点迭代过程的放大图, 可以发现, 总是一部分节点在整个迭代过程中高频而规律的出现; 而另一部分节点则始终遵循一定的先后关系以较低频率规

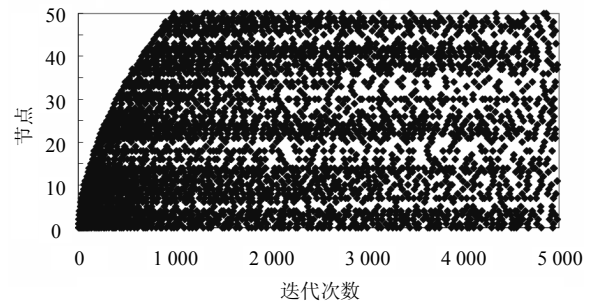
律的出现; 还有一些节点则在整个迭代过程中随机出现。这样的动力学现象与复杂性理论中混沌边缘现象类似, 系统即不完全随机也不遵循固定规律, 而是出于两种状态之间。因此, 当节点的动力学规律展现出类似混沌边缘现象时, 系统能快速得到图的最优着色, 宏观上体现为算法能在线性或亚线性时间复杂度找到最优解。



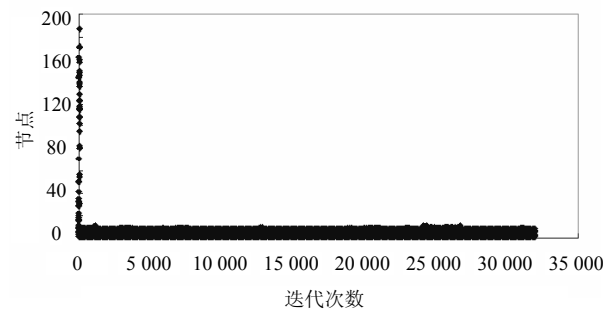
a. 求解Anna.col的禁忌搜索启发式算法动力学过程全过程



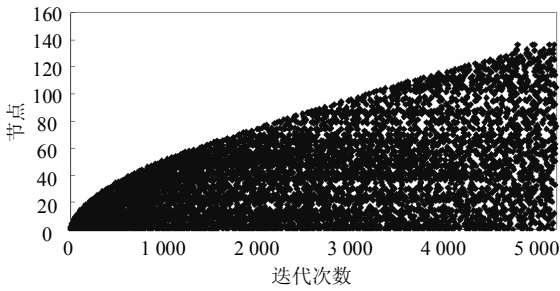
b. 求解Anna.col的群体协作算法(基于规则3)动力学过程全过程



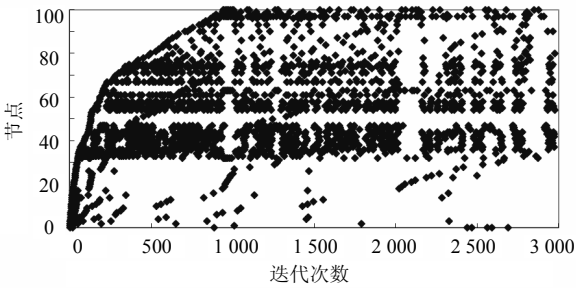
c. 求解Anna.col的群体协作算法(基于规则3)动力学过程全过程: 0~50号节点动力学过程的放大图



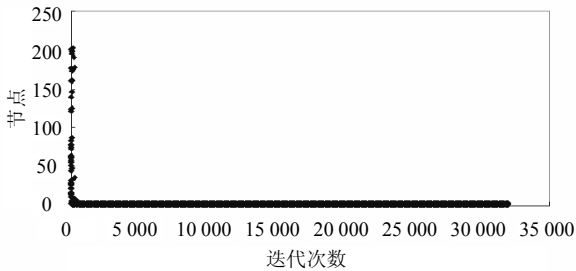
d. 求解mulsol.i.2.col的禁忌搜索启发式算法动力学过程全过程



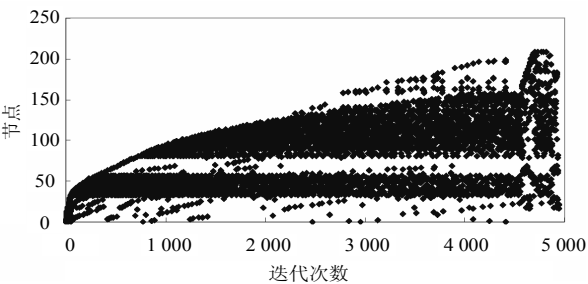
e. 求解mulsol.i.2.col的群体协作算法(基于规则3)动力学过程全过程



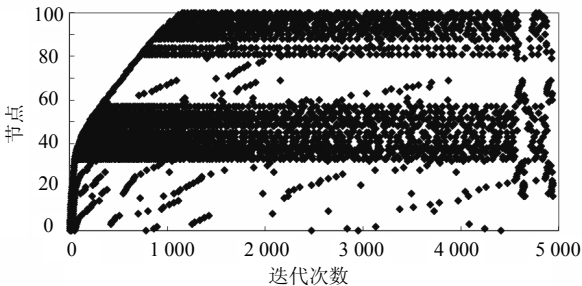
f. 求解mulsol.i.2.col的群体协作算法(基于规则3)动力学过程全过程: 0~100号节点动力学过程的放大图



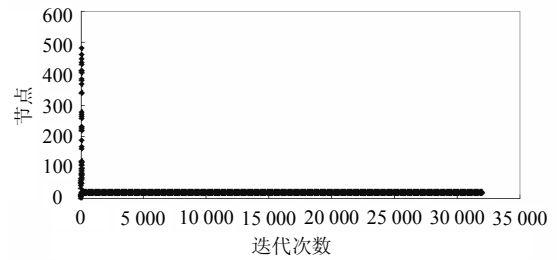
g. 求解zeroin.i.1.col的禁忌搜索启发式算法动力学过程全过程



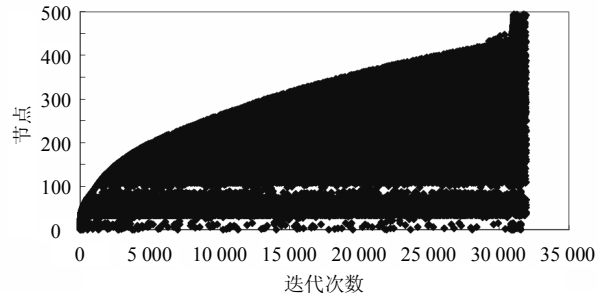
h. 求解zeroin.i.1.col的群体协作算法(基于规则3)动力学过程全过程



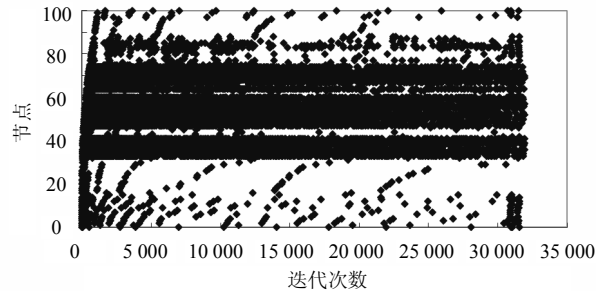
i. 求解zeroin.i.1.col的群体协作算法(基于规则3)动力学过程全过程: 0~100号节点动力学过程的放大图



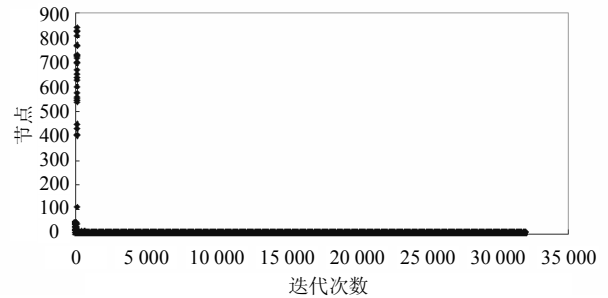
j. 求解fpsol2.i.1.col的禁忌搜索启发式算法动力学过程全过程



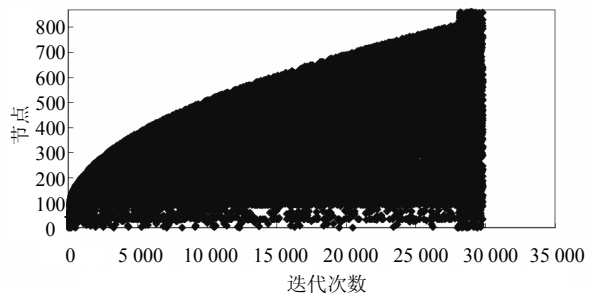
k. 求解fpsol2.i.1.col的群体协作算法(基于规则3)动力学过程全过程



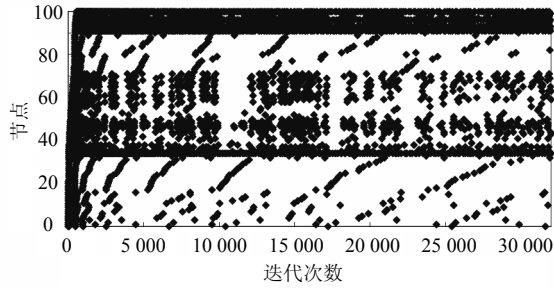
l. 求解fpsol2.i.1.col的群体协作算法(基于规则3)动力学过程全过程: 0~100号节点动力学过程的放大图



m. 求解inithx.i.1.col的禁忌搜索启发式算法动力学过程全过程

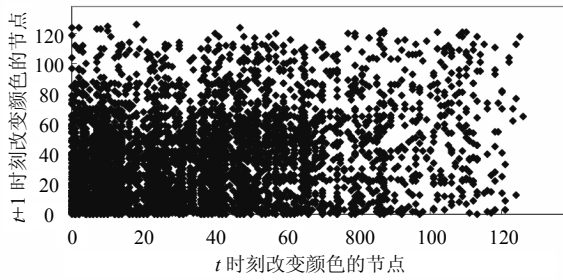


n. 求解inithx.i.1.col的群体协作算法(基于规则3)动力学过程全过程

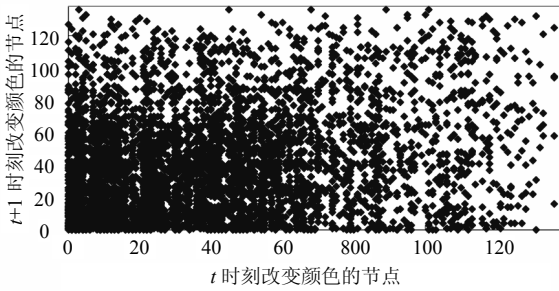


o. 求解inithx.i.1.col的群体协作算法(基于规则3)动力学过程全过程: 0~100号节点迭代过程的放大图

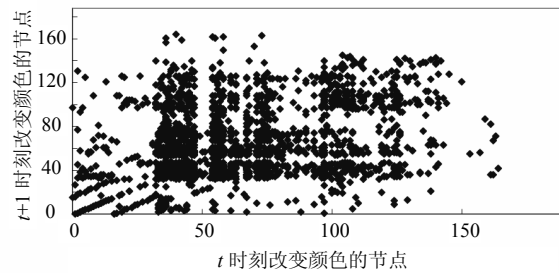
图6 算法的动力学过程



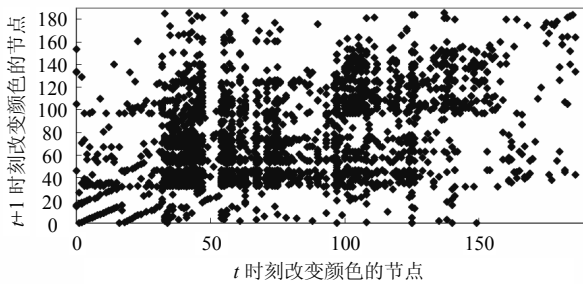
a. 基于规则1求解Anna.col动力学过程的一阶时滞图



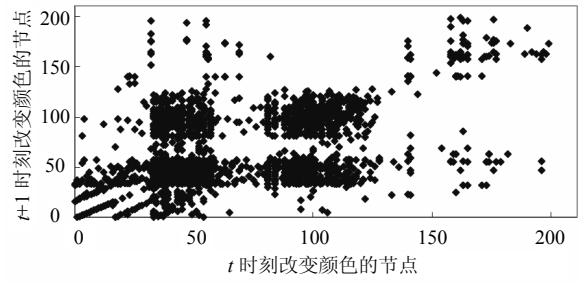
b. 基于规则3求解Anna.col动力学过程的一阶时滞图



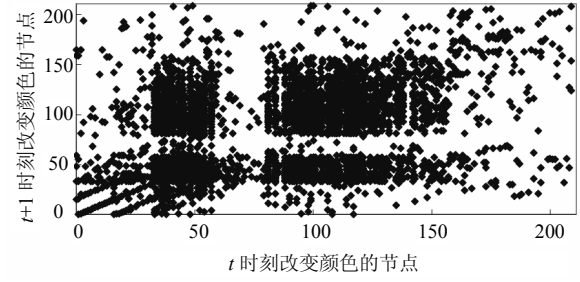
c. 基于规则1求解mulsol.i.2.col动力学过程的一阶时滞图



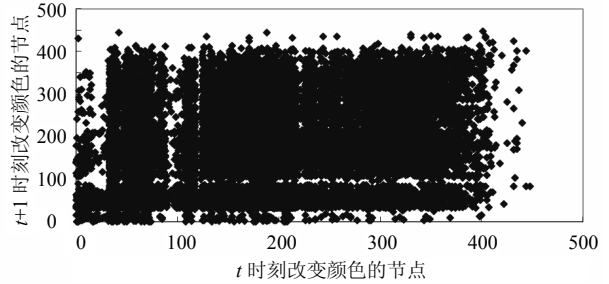
d. 基于规则3求解mulsol.i.2.col动力学过程的一阶时滞图



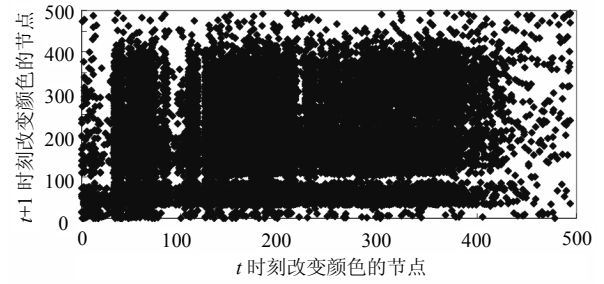
e. 基于规则1求解zeroin.i.1.col动力学过程的一阶时滞图



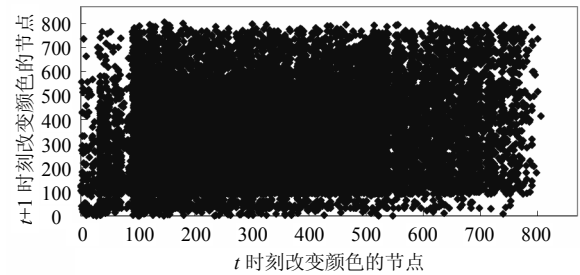
f. 基于规则3求解zeroin.i.1.col动力学过程的一阶时滞图



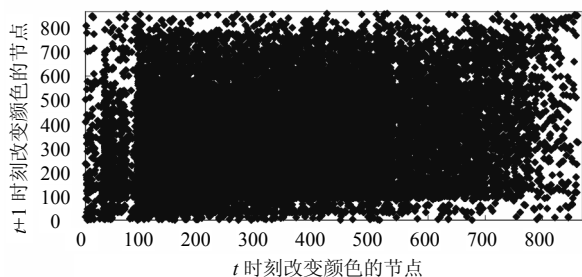
g. 基于规则1求解fpsol2.i.1.col动力学过程的一阶时滞图



h. 基于规则3求解fpsol2.i.1.col动力学过程的一阶时滞图



i. 基于规则1求解inithx.i.1.col动力学过程的一阶时滞图



j. 基于规则3求解inithx.i.1.col动力学过程的一阶时滞图

图7 规则1和规则3动力学过程的一阶时滞图

图7是图6在一个单位时滞上节点之间的关联关系, 可以看到在规则1和规则3指导下, 算法的动力学模式在大体上是相似的, 但规则3比规则1拥有更加丰富的节点协作模式。如对图zeroin.i.1.col和mulsol.i.2.col, 都明显的有4个密集区域, 但规则3的4个区域都明显大于规则1的4个区域, 这表示节点之间的协作关系在规则3的指导下比在规则1的指导下更加多样化。

因此, 本文认为当算法输出的节点序列呈现出混沌边缘现象后, 该节点序列的一阶时滞图能够反映该节点间协作模式的丰富程度, 具有更加多样化协作模式的规则更容易找到最优解。

2) 迭代过程具有随机性或收敛性动力学特征的图

本节主要针对表2中图6~图9分析在不同群体协作规则和启发式规则指导下形成的算法动力学特征, 结果如图8所示。

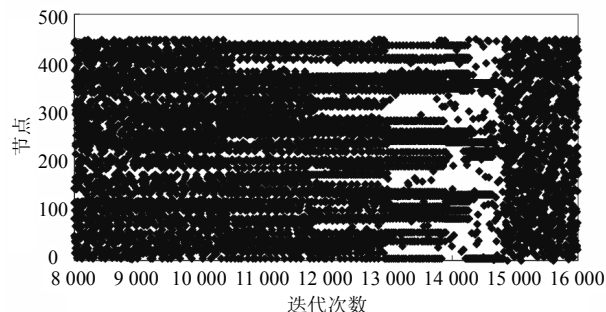
对规则3, 算法动力学过程呈现出很强的随机性, 改变颜色的节点随时间随机变化, 此时规则3要指数时间复杂度才能找到最优着色数。

对规则1, 算法的动力学过程首先是呈现出很强的随机性, 接着会在一段时间后收敛到少数几个节点, 此时为了使其跳出局部最优, 将规则1与规则5配合使用, 当使用一次规则5后, 基于规则1的算法动力学过程又会首先呈现出很强的随机性, 接着在一段时间后收敛到少数几个节点, 如此循环。图8中基于规则1的算法动力学过程均表示使用规则5(规则5中随机数阈值 $T$ 设置为0.005)进行一次随机扰动前后规则1的动力学过程从随机变成收敛再变成随机的过程, 但无论使用规则5扰动多少次, 规则1均无法得到最优着色数。

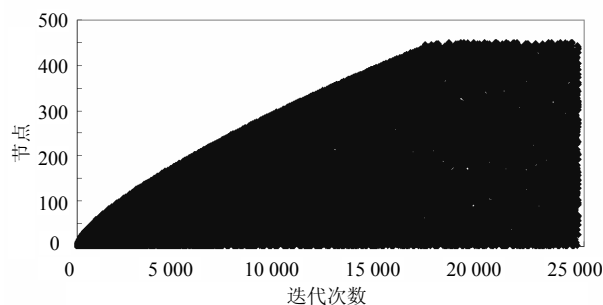
对基于禁忌搜索规则的启发式方法的动力学特征也呈现出较强的随机性(对于表2中的图6, 由于源码本身的原因未能画出节点的时间序列), 但其在较

长时间的搜索过程中未能找到最优着色数。

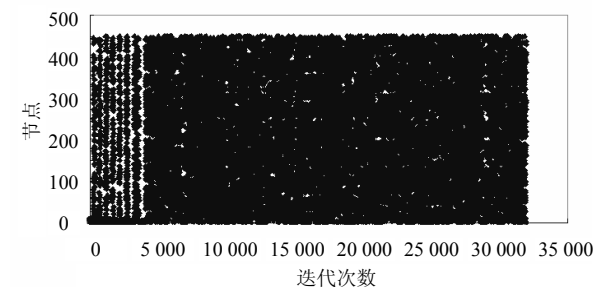
上述现象说明, 当算法动力学特征呈现出随机性或收敛性时, 算法无法有效协调各微观节点进行协作, 失去了快速涌现出最优解的能力, 退化搜索能力与蛮力搜索相似的算法。



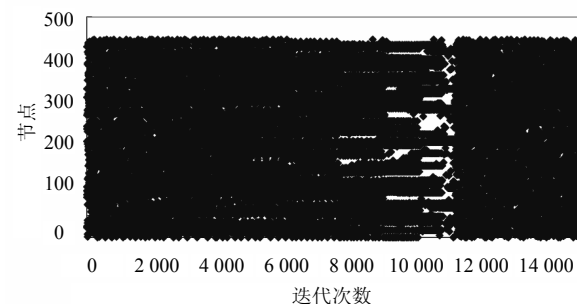
a. 基于规则1的群体协作算法求解图le450\_5a.col: 一次随机扰动前后改变颜色节点的时间序列从随机到收敛再到随机的过程



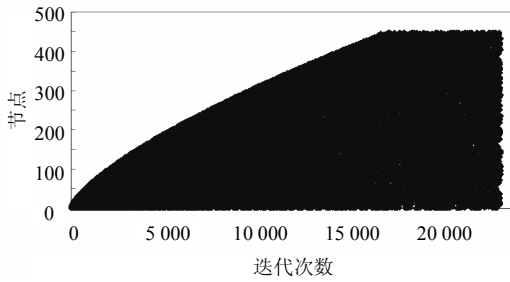
b. 基于规则3的群体协作算法求解图le450\_5a.col: 改变颜色节点的时间序列一直保持随机变化



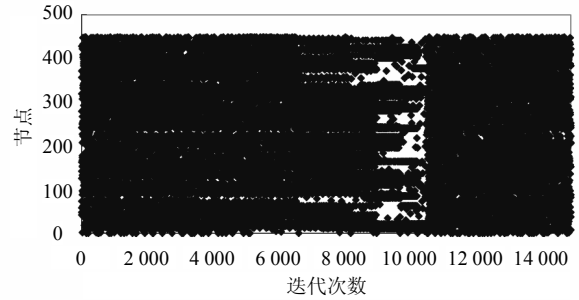
c. 基于禁忌搜索的启发式算法求解图le450\_5b.col: 改变颜色节点的时间序列一直保持随机变化



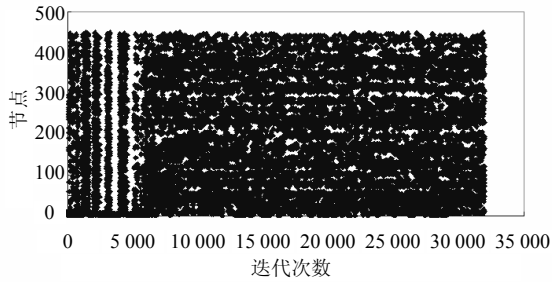
d. 基于规则1的群体协作算法求解图le450\_5b.col: 一次随机扰动前后改变颜色节点的时间序列从随机到收敛再到随机的过程



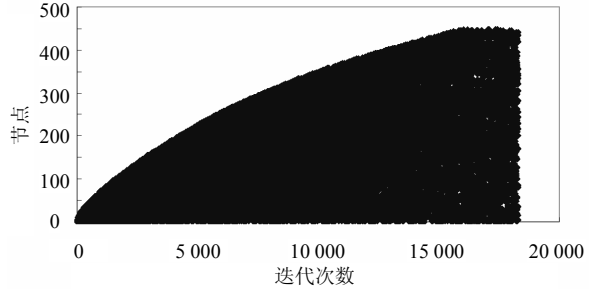
e. 基于规则3的群体协作算法求解图le450\_5b.col: 改变颜色节点的时间序列一直保持随机变化



j. 基于规则1的群体协作算法求解图le450\_5d.col: 一次随机扰动前后改变颜色节点的时间序列从随机到收敛再到随机的过程

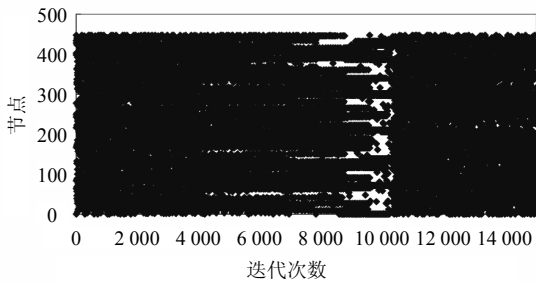


f. 基于禁忌搜索的启发式算法求解图le450\_5c.col: 改变颜色节点的时间序列一直保持随机变化

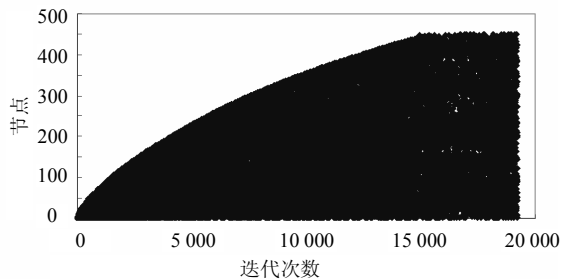


k. 基于规则3的群体协作算法求解图le450\_5d.col: 改变颜色节点的时间序列一直保持随机变化

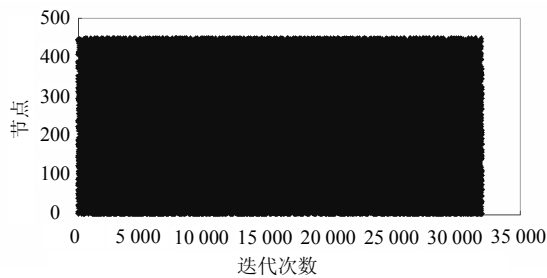
图8 算法的动力学过程



g. 基于规则1的群体协作算法求解图le450\_5c.col: 一次随机扰动前后改变颜色节点的时间序列从随机到收敛再到随机的过程



h. 基于规则3的群体协作算法求解图le450\_5c.col: 改变颜色节点的时间序列一直保持随机变化



i. 基于禁忌搜索的启发式算法求解图le450\_5d.col: 改变颜色节点的时间序列一直保持随机变化

## 6 结束语

本文主要基于社群联盟中真实的社群协作规则来设计求解图着色问题的节点协作规则, 由于社群协作原则能够保证真实的最优社群联盟问题得到快速求解, 所以节点协作原则也能够保证最优图着色问题得到快速求解。通过在真实数据集上的实验证明了群体协作模型的可行性和有效性。

接下来的工作包括: 1) 对图着色问题尝试设计更优化的协作规则, 进一步降低算法迭代时间; 2) 将群体协作的计算模式运用到其他问题上, 如TSP问题等。

本文研究工作得到成都信息工程大学中青年学术带头人科研基金(J201208, J201101)的资助, 在此表示感谢。

## 参 考 文 献

[1] LEEUWARDEN N H. Shared space: Reconciling people, places, traffic[J]. Built Environment, 2008, 34(2): 161-181.  
 [2] BOL L, CUNNINGHAM W. The wiki way[M]. Boston, MA: Addison Wesley Press, 2001.  
 [3] BRINGSJORD S, TAYLOR J. P=NP[J]. Machine Ethics, 2011, 7(7): 361-374.  
 [4] KHATIB F, DIMAIO F. Crystal structure of a monomeric retroviral protease solved by protein folding game players[J]. Nature Structural and Molecular Biology, 2011, 364(19): 1175-1177.

- [5] AXELROD R. The evolution of cooperation[M]. Princeton: Princeton University Press, 1997.
- [6] CHRIS A. The long tail: Why the future of business is selling less of more[M]. New York: Hyperion Press, 2008.
- [7] NICHOLAS C, FOWLER J. Connected: the surprising power of our social networks and how they shape our lives[M]. New York: Brown and Company Press, 2009.
- [8] JAMES S. The wisdom of crowds, economies, societies and nations[M]. New York: Little Brown Press, 2004.
- [9] ROBERT A. The evolution of cooperation[M]. London: Basic Books Press, 2006.
- [10] TAPSCOTT D, WILLIAMS A D. Wikinomics: How mass collaboration changes everything[M]. New York: Portfolio Trade Press .2006.
- [11] ARAL S, ALSTYNE V. Network structure and information advantage[C]//Proceedings of the Academy of Management Conference. Philadelphia. USA: ACM Press, 2007: 3-8.
- [12] DAWBER T. The framingham study: the epidemiology of atherosclerotic disease[M]. Cambridge, MA: Harvard University Press, 1980.
- [13] PENTLAND A. Computational social science[J]. Science, 2009, 323(5915): 721-723.
- [14] PENTLAND A. Pervasive sensing to model political opinions in face-to-face networks[J]. Pervasive Computing, 2011, 14(23): 214-231.
- [15] PENTLAND A. Sensible organizations: Technology and methodology for automatically measuring organizational behavior[J]. IEEE Transactions on Systems, Man, and Cybernetics Part B, 2009, 39(1):43-55.
- [16] GONZALEZ M. Understanding individual human mobility patterns[J]. Nature, 2008, 453: 779-782.
- [17] MONTOULIU R, GATICAL P. Discovering human places of interest from multimodal mobile phone data[J]. Multimed Tools Appl, 2013, 62: 179-207.
- [18] LU Hong. The Jigsaw continuous sensing engine for mobile phone applications[C]//Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems. Zürich, Switzerland: ACM Press, 2010: 71-84.
- [19] LIU Ji-ming, Autonomy oriented computing(AOC): Formulating computational systems with autonomous components[J]. IEEE Transactions on Systems, Man and Cybernetics, 2005, 35(6): 100-139.
- [20] NELKEN R, YAMANGIL E. Mining wikipedia's article revision history for training computational linguistics algorithms[C]//Proceedings of the 1st AAAI Workshop on Wikipedia and Artificial Intelligence. Chicago, USA: ACM Press, 2008:423-430.
- [21] SCHAERF A. A survey of automated time tabling[J]. Artificial Intelligence Review, 2009, 16(48): 87-127.
- [22] DELARK DE WERRA. Heuristics for graph coloring[J]. Computing Supplementum, 1990, 7(2): 191-208.
- [23] 马艳萍, 吴晓军. 解决图着色问题的一种新禁忌搜索算法[J]. 计算机应用与软件, 2012, 29(2): 279-281.
- MA Yan-ping, WU Xiao-jun. A new tabu search algorithm of graph coloring[J]. Computer Applications and Software, 2012, 29(2): 279-281.
- [24] 徐杰, 杜文, 李宗平, 等. 基于模拟退火算法和图着色的调车机车安排研究[J]. 铁道学报, 2003, 25(3): 24-30.
- XU Jie, DU Wen, LI Zong-ping, et al. Study on the plan of using shunting locomotives based on simulated annealing algorithm and graph coloring[J]. Journal of The China Railway Society, 2003, 25(3): 24-30.
- [25] PHILIPE G, HAO Jin-kaio. Hybrid evolutionary algorithms for graph coloring[J]. Journal of Combinatorial Optimization, 1999, 3(4): 379-397.
- [26] RAILSBACK S. Agent-based and individual-based modeling: a practical introduction[M]. Cambridge: Princeton University Press, 2011.
- [27] TRICK M. Graph coloring instances[EB/OL]. (2003-10-21). <http://mat.gsia.cmu.edu/COLOR/instances.html>.
- [28] MEHROTRA A. GraphColoring[EB/OL]. (2007-05-21). <http://graphcol.googlecode.com/files/graphCol.tar.gz>.
- [29] TRICK M. Graph coloring instances[EB/OL]. (2003-11-03). <http://mat.gsia.cmu.edu/COLOR/solvers/trick.c>.

编辑 蒋晓