基于连接强度的PPI网络蚁群优化聚类算法

雷秀娟[1],黄 旭[1],吴 爽[1],郭 玲[2]
(1.陕西师范大学计算机科学学院,陕西西安 710062; 2.陕西师范大学生命科学学院,陕西西安 710062)

摘 要: 由于PPI网络数据的无尺度和小世界特性,使得目前对此类数据的聚类算法效果不理想.根据PPI网络的拓扑结构特性,本文提出了一种基于连接强度的蚁群优化(Joint Strength based Ant Colony Optimization,JSACO)聚类算法,该算法引入了连接强度的概念对蚁群聚类算法中的拾起/放下规则加以改进,以连接强度作为拾起规则,对结点进行聚类,并根据放下规则放弃部分不良数据,产生最终聚类结果.最后采用了MIPS数据库中的PPI数据进行实验,将JSACO算法与PPI网络数据的其他聚类算法进行比较,聚类结果表明JSACO算法正确率高,时间开销低.

关键词: PPI网络; 连接强度; 蚁群优化算法; 聚类
中图分类号: TP301.6 **文献标识码:** A **文章编号:** 0372-2112 (2012)04-695-08
电子学报URL: http://www.ejournal.org.cn **DOI:** 10.3969/j.issn.0372-2112.2012.04.012

Joint Strength Based Ant Colony Optimization Clustering Algorithm for PPI Networks

LEI Xiu-juan[1], HUANG Xu[1], WU Shuang[1], GUO Ling[2]
(1. *School of Computer Science, Shaanxi Normal University, Xi'an, Shaanxi* 710062, *China*;
2. *School of Life Science, Shaanxi Normal University, Xi'an, Shaanxi* 710062, *China*)

Abstract: Due to the sale-free and small-world characters of Protein-Protein Interaction (PPI) network data, current clustering algorithms did not perform well. According to the topological structural characters of PPI networks, this paper proposed an ant colony optimization clustering algorithm based on joint strength (JSACO). This method modified the pickup/drop rules of ACO algorithm by means of introducing the concept of joint strength, which regarded the joint strength as pickup rule to cluster the protein nodes. In addition, the protein nodes which had the low joint strength were abandoned in accordance with drop rule and the final clustering result was obtained. Finally the PPI data in MIPS database was used to test the algorithm and the clustering result was compared with other PPI clustering methods. The simulation results show that JSACO algorithm performs better in terms of precision value and consumes less time.

Key words: PPI Network; joint strength; ACO algorithm; clustering

1 引言

在生物信息学近几年的发展中,基因和蛋白质组学是发展得最快的几个领域之一.由于海量的蛋白质数据的出现,使得生物信息学对蛋白质的研究转向了数据分析.在蛋白质相互作用(Protein-Protein Interaction,PPI)网络的研究中,一个关键的问题是如何尽可能多和尽可能准确地发现PPI网络中的蛋白质复合物及功能模块.对于维数高且具有小世界、无尺度[1]特性的蛋白质数据,算法必须能够高效地提取出PPI网络中具有生物学意义的子网络.

聚类是对PPI网络分析的常用方法之一.PPI网络分析的目的是希望找到其中的蛋白质复合物,复合物内部的蛋白质之间联系较紧密且具有一定的相似性,而聚类的特性[2]正好符合这一要求.

传统的聚类方法可分为以下几种:基于划分的方法、基于层次的方法、基于密度的方法、基于网格的方法和基于模型的方法[2].但这些方法都有一定的缺陷,有的算法需要预先知道聚类数目或其他先验知识,有的算法无法对不规则形状的数据进行聚类,有的算法对稀疏数据的聚类效果不理想等等.继这些传统方法之后,近几年还提出了一些新的算法,如谱聚类算法、功能流聚

收稿日期:2011-04-01;修回日期:2011-12-27
基金项目:国家自然科学基金(No.61100164,No.61173190);陕西省自然科学基础研究计划(No.2010JQ8034);陕西师范大学中央高校基本科研业务费专项资金(No.GK200902016)

类算法等等.而谱聚类算法在数据降维后最终要回归到传统的聚类方法上.Young-Rae Cho[3]提出了一种聚类效果相对较好的功能流聚类算法,但该算法中的部分参数需要根据经验人为给定,这就影响了其聚类结果的准确度和算法的稳定性.金弟等人[4]提出了基于局部探测的快速复杂网络聚类算法,在随机网络上效果较好,对于复杂网络也做了简单测试,算法优化的是网络模块性函数 Q,但反映的是网络的局部信息.我们曾尝试将人工蜂群思想融入到功能流聚类算法中,但算法的时间复杂度比较高[5].

人们在对鱼群、鸟群、蜂群、蚁群等群居生物的研究过程中发现,群居生物常常体现出一定的群智能行为.在这些生物群体的研究对象中,蚂蚁[6]是最为出名的群体之一,它们在作为一个群体时能够完成单个蚂蚁无法完成的许多事情.Dorigo[6,7]进行了一系列的研究并首先提出了蚁群算法的概念.该算法模拟了蚂蚁觅食的过程,并利用了蚂蚁在该过程中的高度自组织性[6]去解决旅行商问题(TSP).随后基于蚁群思想的群智能算法不断涌现,比如 Jamaludin Sallim[8]将 ACO 算法与 TSP 问题结合并应用到聚类分析上;Amita Lanjewar[9]则直接将 ACO 算法用于解决移动机器人聚类问题等等.

在各种群智能优化算法中,有不少算法只是针对目标函数最优化问题,如遗传算法(GA)、粒子群优化(PSO)算法、人工蜂群(ABC)算法等.我们[5]尝试将群智能优化算法融入到基因和 PPI 数据的聚类分析中,但这些方法只能解决整个聚类算法中的一个或几个小问题,并没有完全利用和发挥出智能优化算法的优势.在群智能优化算法中,蚁群算法与以上几种算法在思想和应用范围上有所不同.蚁群算法本身就能直接进行聚类操作,而不需要借助其他聚类算法实现聚类的目的.运用群智能优化机理来解决 PPI 网络功能模块分析而非将其作为一个辅助工具,更能发挥出群智能算法的特点及优越性.

本文提出了一种基于连接强度的 PPI 网络蚁群优化(Joint Strength based Ant Colony Optimization,JSACO)聚类算法,算法采用蚂蚁的拾起规则进行聚类,在此过程中,对拾起规则进行了一定的改进,以提升算法对 PPI 网络数据的聚类效果,同时降低算法的时间复杂度.

2 基本理论

2.1 蚁群算法

蚁群算法是由 M Dorigo 首次提出的[6].与其他的群智能优化算法类似,蚁群算法实际上也是一种算法的指导思想,任何具有该算法思想框架的实际算法都可以称之为蚁群算法[10,11].在此以旅行商问题为例,对蚁群算法的基本概念进行介绍,假设:

m 为蚂蚁数;

$b_i(t)$为在 t 时刻位于城市 i 上的蚂蚁数;

d_{ij}为城市 i 和城市 j 之间的距离;

$\tau_{ij}(t)$为 t 时刻在城市 i 和城市 j 之间的信息素;

$\Delta\tau_{ij}$为 τ_{ij}的变化量;

$p_{ij}^k(t)$为蚂蚁 k 在时刻 t 从城市 i 转移到城市 j 的概率;

α 为启发式因子,表征信息素残余量的重要程度;

β 为探索因子,表征能见度的重要性;

ρ 为蒸发因子,表征信息素的蒸发速度;

每一只蚂蚁从其中的一个城市出发,以一定的概率选择下一个城市,蚂蚁 k 从城市 i 移动到城市 j 的概率的计算公式如式(1)所示:

$$P_{ij}^k(t)=\frac{[\tau_{ij}(t)]^\alpha[\eta_{ij}]^\beta}{\sum_{l\in N_i^k}[\tau_{il}(t)]^\alpha[\eta_{il}]^\beta}\quad \text{if}\quad j\in N_i^k \tag{1}$$

其中 $\eta_{ij}=1/d_{ij}$是一个权重因子,两个城市之间的距离越远,则权重值越小.N_i^k 是未被访问的城市集合,也就是禁忌表.经过一次循环以后,更新每条通路上的信息素余量,对每条通路给出信息素反馈质量.该反馈质量将作为以后循环的判断依据,为蚂蚁选择通路提供信息.信息素的更新如式(2)和式(3)所示:

$$\tau_{ij}(t+n)=(1-\rho)\cdot\tau_{ij}(t)+\Delta\tau_{ij}(t) \tag{2}$$

$$\Delta\tau_{ij}(t)=\sum_{k=1}^{m}\Delta\tau_{ij}^k(t) \tag{3}$$

其中 $\Delta\tau_{ij}^k$表示蚂蚁 k 在走过城市 i 和城市 j 时留下的信息素的总量.

蚁群算法首先进行参数的初始化,随机选择一条路径并更新禁忌表.在第一次循环过后,更新信息素余量,然后根据式(1)计算得到的转移概率重新进行循环迭代,直到达到最大迭代次数或满足终止条件为止.

2.2 连接强度

虽然密度是一种较常用的聚类判断准则,但其在区分不同的拓扑结构时仍然存在一定的不足.在文献[12]中曾指出如下的情况:在图 1 中,令黑色结点组成的子图为 K,白色结点 e,f 为待考察的结点.从直观上判断图 1(a)中的结点要比图 1(b)中的结点之间的连接更紧密些.但在这两种情况中,图的密度都是 8/15.

因此,对于不同的拓扑结构,李敏[12]提出了连接强度的概念,其定义如下:

设结点 v 与图 K 中的结点的连接边数为 M_{vK},图 K 的结点数为 n_K,则结点 v 与图 K 的连接强度 P_{vK}的计算公式为:

$$P_{vK}=\frac{M_{vK}}{n_K} \tag{4}$$

以图 1 为例,在图 1(a)中的结点 e、f 即定义中的结点 v,由黑色结点组成的子图即定义中的图 K,e、f 与该子图的连接强度都为 0.5;而在图 1(b)中的连接强度则为 0.25 和 0. 这样就有效地区分了由拓扑结构所带来的差异性.

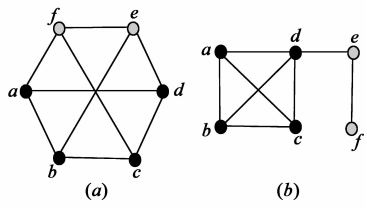


图1　拓扑网络图

文献[13]指出在 PPI 网络中,一些极大子图或接近极大子图的结构往往就是一个具有较完整功能的蛋白质复合物.因此对极大子图或极大子图的扩展规则——连接强度就显得十分有效.连接强度的概念能够在兼顾密度的同时还区分了拓扑结构,其具有如下定理[12]:

定理 1　给定一个无向简单图 G、常量 t 及子图 K,若 K 中每个顶点 v 都满足 $P_{vK} \geqslant t$,则子图 K 的密度 $d_s(K) > t$.

由此定理可知,采用连接强度能够在考虑到密度影响的同时,还能对不同的拓扑结构进行区分,使聚类效果更有效.

2.3　评价准则

在 PPI 网络中,功能模块的识别可以为未知蛋白质的功能预测提供指导作用.由于采用不同的方法会产生不同的效果,而且同一种方法采用不同的参数也可能会产生不同的效果,因此聚类结果评价方法的选择具有很重要的作用.比较常用的评价方法就是正确率和查全率.假定一个模块 X 被映射到标准数据集的一个类 F_i 中,正确率 $precision$[1] 表示模块 X 和 F_i 的交集的结点个数与模块 X 中的蛋白质结点个数的比值,查全率 $recall$[1] 则表示模块 X 和 F_i 的交集的结点个数与类 F_i 的蛋白质结点个数的比值.

$$precision = \frac{|X \cap F_i|}{|X|} \tag{5}$$

$$recall = \frac{|X \cap F_i|}{|F_i|} \tag{6}$$

当所有的蛋白质结点都分到了一个类中,那么该类的查全率就达到最大值.反之,如果聚类模块比较小,那么查准率就会非常高.

3　JSACO 算法描述

3.1　算法思想

JSACO 算法采用 Lumer 和 Faieta[14] 提出的基于拾起/放下规则的蚁群聚类算法,并修改了其中的拾起规则.LF 算法[14] 的主要思想描述如下:

将要进行聚类操作的数据随机地散落在一个二维平面上,并在平面上随机产生部分蚂蚁.蚂蚁在移动的过程中,可以对自身可视范围内的数据进行相似性判断,若自身携带的数据与周围数据的相似度较低,则将该数据拾起;反之,则放下该数据.

在该算法中,设蚂蚁的能见度为 S,所处的位置为 r,要比对的对象为 O_i,则相似度计算公式为:

$$f(O_i) = \begin{cases} \dfrac{1}{S^2} \displaystyle\sum_{O_j \in Neigh_{S\times S}(r)} \left[1 - \frac{d(O_i, O_j)}{\alpha}\right], & \text{if} \quad f(O_i) > 0 \\ 0, & \text{otherwise} \end{cases} \tag{7}$$

其中 $d(O_i, O_j)$ 是对象 i 与对象 j 之间的距离,α 为衡量相异度的因子.蚂蚁拾起或放下数据的概率按式(8)和式(9)计算:

$$P_p(O_i) = \left(\frac{k_1}{k_1 + f(O_i)}\right)^2 \tag{8}$$

$$P_d(O_i) = \begin{cases} 2 \cdot f(O_i), & \text{if} \quad f(O_i) < k_2 \\ 1, & \text{if} \quad f(O_i) > k_2 \end{cases} \tag{9}$$

其中 k_1 和 k_2 都是常数,拾起或放下的规则是:随机产生一个概率与拾起和放下概率比较,若大于该概率,则执行相应的操作.

由以上对 LF 算法的描述可以看出,随机产生的概率很可能导致对一个数据反复地拾起和放下,这样对聚类算法的效率会产生较大的影响.因此 JSACO 算法在该算法思想的基础上,作了如下的改进:

首先,初始的蚂蚁位置从一部分较优数据处随机产生,而非完全随机生成.目的在于通过智能选取较优的起始状态来提升算法效率,同时也保留了一定的随机性,以避免算法陷入局部最优.

其次,对拾起规则进行调整.考虑到连接强度的概念能够更好地反映结点与类之间联系的紧密性,且计算方式简单,因此采用连接强度计算公式(4)代替式(7)到式(9)的方法进行判断.

与此同时,该算法还对蚂蚁的拾起/放下规则进行调整.蚂蚁不仅是数据的搬运工具,而且还可以作为收集数据的个体.每个蚂蚁首先不断地对数据进行拾起操作而不进行放下操作,通过限定蚂蚁对数据的装载量及行动范围(蚂蚁在拾起数据达到一定数量时就会停止拾起,或者在走到一定远的距离时就会停止,只要满足数量或者距离的其中一个条件时就会停止拾起数据)达到限制结果增长规模的目的.在停止拾起数据后,再通过放下规则判断是否有需要舍弃数据.采用此方式的优点在于拾起和放下操作都只进行了一次,避免了蚂蚁不断对同一数据执行拾起和放下操作,提高算法的性能.改进之后的 JSACO 算法思想如下:

对于一个输入的 PPI 网络,从度数较高的蛋白质中

随机选取部分蛋白质作为蚂蚁起点,起点集的定义如式(10)所示:

$$\{a \mid degree(a) > degree\_value\} \tag{10}$$

其中 $degree(a)$表示结点 a 的度,$degree\_value$ 表示符合初始选取条件的最小度数,其取值越大,可选的初始结点就越少,聚类结果的类平均大小就越大.考虑到这个参数的选取对聚类结果的影响比较小,所以在仿真结果部分不做具体分析.

蚂蚁在 PPI 网络中的蛋白质相互作用的边上进行移动,移动的范围上限为一个固定值(实验中取值为 5,即蚂蚁当前位置与初始位置之间隔了 4 个结点以上就不再移动,停止拾起数据了.这个值是根据标准数据库中的类平均大小设定的).在移动的过程中,蚂蚁的可视范围为当前位置结点的直接邻接结点.接下来对可视范围内的蛋白质计算其连接强度,若高于预设的接收阈值,则拾起该结点,否则放弃.

当蚂蚁已访问了其可移动范围内的所有蛋白质结点,或拾起的数据已超过其装载量,则停止拾起数据.并对已拾起的数据计算其放下概率,放弃不良数据并生成一个聚类结果.

3.2 算法步骤

JSACO 算法步骤如下:

步骤 1 对数据进行预处理:计算所有结点的度,并按度的大小对数据进行排序,在度较高的结点中随机选择一个结点作为初始蚂蚁;

步骤 2 在蚂蚁的可视范围内计算结点的连接强度,以连接强度作为拾起规则,对结点进行聚类;

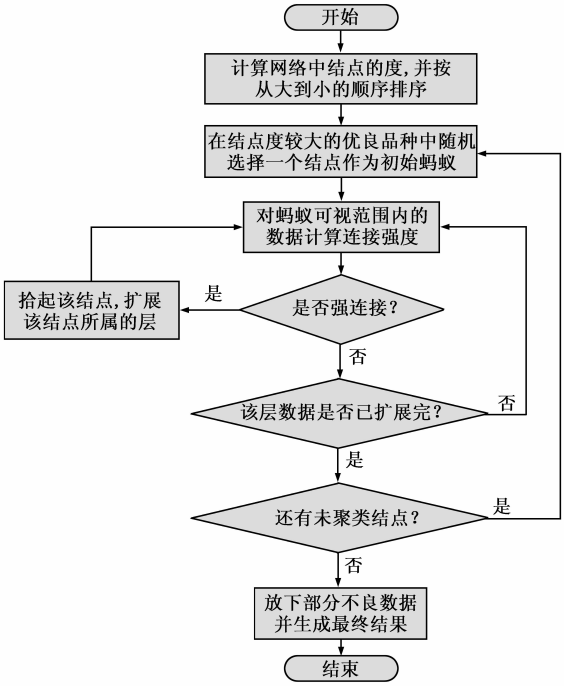


图2 JSACO算法流程图

步骤 3 若有新拾起的结点,则前进到新结点,并根据蚂蚁的行走距离判断是否继续前进.若是,则返回步骤 2,对新的可视范围进行聚类;若否,则转入步骤 4;

步骤 4 检查是否还有未聚类的结点,若有,选择一个较优结点作为蚂蚁的新起点,返回步骤 2;否则进入步骤 5;

步骤 5 对聚类结果进行处理:根据连接强度作为放下规则放弃部分不良数据,并产生最终聚类结果.

算法的流程图如图 2 所示.

3.3 算法的时间复杂度

设 PPI 网络的数据规模为 N,算法在聚类的过程中,每个蛋白质结点最少会被访问 1 次.而在最坏的情况下(每个结点在每一次的聚类中都被访问到,但是又都没有被拾起),此时每个结点被访问了 N 次.因此在最好的情况下,算法的时间复杂度为 $O(N)$,在最坏的情况下,时间复杂度为 $O(N^2)$.

4 仿真实验及结果分析

4.1 实验数据及实验环境

本次实验的运行环境:Windows XP 操作系统,1G 物理内存,Matlab 7.0.本次实验的运行以 MIPS 数据库[15]为实验数据,以 MIPS 提供的标准结果库为参照标准,用于衡量采用本文算法得到的聚类结果的准确率以及算法的运行效率.数据集是通过无向带权图表示的,图的表示方式与文献[3]是相同的.为了方便和简化计算,每一个蛋白质都用数字进行编码,蛋白质之间的相互作用权重用一个二维的邻接矩阵表示,横坐标和纵坐标确定的值为该横坐标和纵坐标表示的蛋白质之间的相互作用强度.

4.2 参数分析

本次实验的结果比对标准是参考给定的 MIPS 数据标准库[15],该标准库的部分信息如表 1 所示.

表 1 标准比对库信息

聚类个数	聚类蛋白质数	总蛋白质数	最大类	最小类	类平均大小
89	516	1376	35	1	5.76

JSACO 算法中的各参数及说明如表 2 所示.

对于表 2 中的 max_size,由于标准库中最大类包含 35 个蛋白质,考虑到可扩展性,此处单个类的最大规模上限设为 50.由文献[16]可知,蛋白质复合物内部的平均距离一般不超过 2,网络中结点的特征路径平均长度小于 5.因此,在这里将 max_dist 的值设为 5.

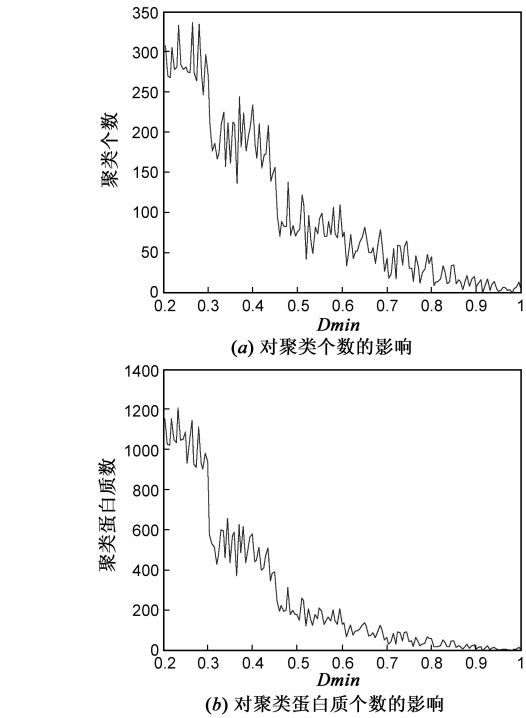
对于 $Dmin$ 值在聚类过程中的作用,本次实验进行了测试.其他参数的值不变,将 $Dmin$ 的值在 0.2 到 1 之间以 0.05 为间隔取值,每个值运行 10 次.运行结果如

图 3 所示.

表 2　实验参数

参数名	参数含义	参数作用
Dmin	相互作用边的最小接受阈值	对最终聚类的蛋白质数和聚类个数产生影响
accept-rate	蛋白质接收的最低连接强度	对最终结果的平均类规模和聚类个数产生影响
max _ size	类的最大规模	对单个类的最大规模作限定
max _ dist	类结点的最大距离	对类内各结点间的最大距离作限定

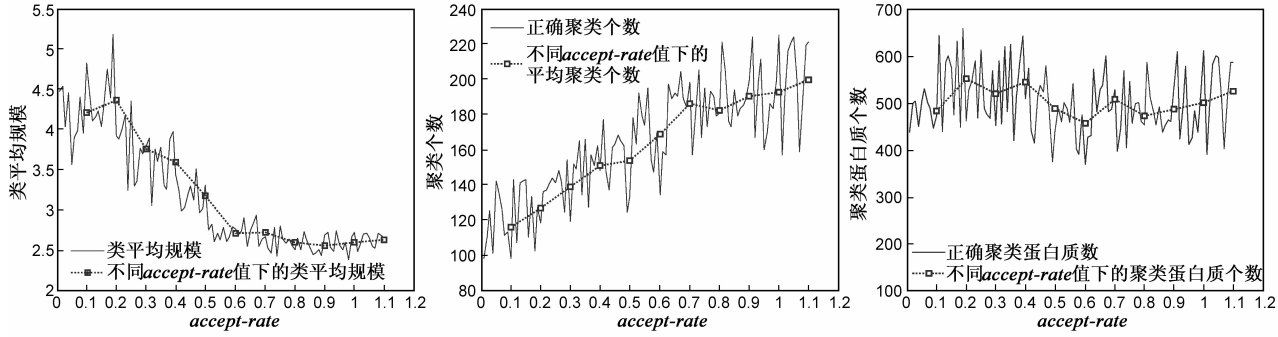
从图 3 可以看出随着 *Dmin* 值的不断增大,最终得到的聚类个数以及蛋白质的数目均呈不断下降的趋势.这是由于随着 *Dmin* 值的增大,使得在聚类的过程中有效的相互作用关系越来越少,蛋白质之间的联系也降低,因此结果中成功聚得的类数和蛋白质数也会不断地减少.通过与标准库的比对结果发现,在 *Dmin* 取 0.4 到 0.6 之间的值时,聚类的平均类数与标准结果比较接近;在 *Dmin* 取 0.3 到 0.5 之间的值时,聚类的蛋白质数与标准结果比较相近.由于标准结果也具有一定的不完备性,考虑到实际结果规模要比标准库规模更大一些,因此 *Dmin* 的值取在 0.3 到 0.4 之间更合理些.



(*a*) 对聚类个数的影响　(*b*) 对聚类蛋白质个数的影响

图3　参数*Dmin*对聚类结果的影响

同样在其他参数不变的情况下,接下来研究 *accept-rate* 的取值对算法可能产生的影响,从 0 到 1 以 0.1 为间隔取值,每个值重复运行 10 次.运行结果如图 4 所示.



(*a*) 对类平均规模的影响　(*b*) 对聚类个数的影响　(*c*) 对聚类蛋白质个数的影响

图4　参数*accept-rate*对聚类结果的影响

在图 4 中,带方块标记的曲线表示 *accept-rate* 在取不同值时运行 10 次的平均值.由于 *accept-rate* 限制的是每个蛋白质在聚到一个类中的接受度(比如 *accept-rate* 为 0.5 时,即两个结点的相似度为 0.5 或以上就归为一个类;若为 0.8,则两个结点的相似度为 0.8 或以上才归为一个类).图 4(*a*)表明类平均规模随着参数 *accept-rate* 值的增大呈下降趋势.图 4(*b*)显示随着 *accept-rate* 值的增大,聚类个数的变化曲线呈上升趋势.实际上每个类中包含的蛋白质越少,而类的个数就会越多.因此,当 *accept-rate* 取 0.1 到 0.3 之间的值时,实验所得的类的平均规模和类的个数都与标准库的较为接近.从图 4(*c*)可以看出该参数对聚类结果中包含的蛋白质的个数影响不大,变化曲线基本维持在同一水平.

5　实验结果及分析

结合以上分析,本次实验的各参数取值为:*Dmin* = 0.3, *accept-rate* = 0.3, *max _ size* = 50, *max _ dist* = 5.考虑到标准库的不完备性,算法也将与 PPI 网络聚类中综合效果较好的功能流聚类方法[3,5]进行横向比较.实验结果如表 3 所示.

表 3 中的数据是算法分别运行 50 次所得的最大值、最小值以及平均值.最后一项调和平均值[17],是对

表 3 蚁群聚类结果表

指标名称	最大值	最小值	平均值
聚类个数	169	119	145.46
聚类蛋白质数	673	418	529
最大类	21	12	17.2
最小类	2	2	2
类平均大小	4.034	3.081	3.627
正确率	92.92%	82.14%	86.85%
查全率	30.63%	21.3%	25.99%
正确聚类数	54	35	45.58
调和平均值	0.22467	0.1714	0.1988

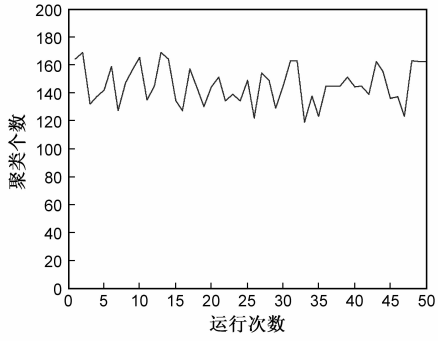


图5 聚类个数变化曲线图

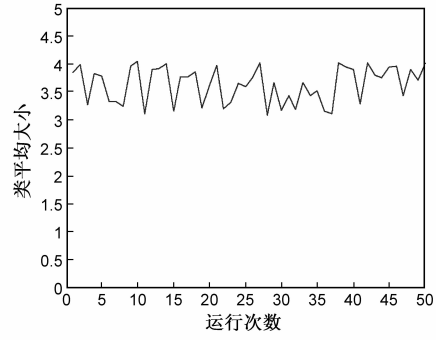


图6 类平均大小变化曲线图

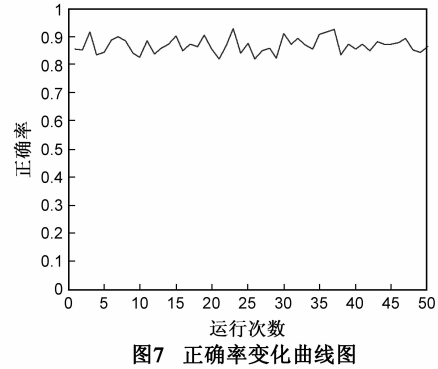


图7 正确率变化曲线图

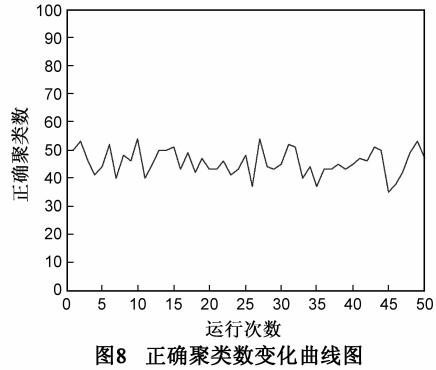


图8 正确聚类数变化曲线图

表 4 蚁群算法与功能流聚类算法结果比对表

算法	正确率	查全率	正确聚类数	运行时间	调和平均值
JSACO	86.85%	25.99%	45.58	49.026	0.55232
Flow	76.78%	43.25%	25.02	225.781	0.51036

表 4 中第一行数据为 JSACO 算法的聚类结果，第二行为功能流聚类方法得到的聚类结果. 其中的结果项中，最后一项的调和平均值，是对正确率、查全率、正确聚类数以及运行时间这 4 项值的一个综合评价，该项指标的最大值为 1，越接近 1 说明结果越好. 在此我们将调和平均值的具体计算公式定义如下：

$$F.score = \frac{4}{\frac{1}{Precision} + \frac{1}{Recall} + \frac{1}{Cluster} + time} \quad (11)$$

正确率、查全率以及正确聚类数的一个综合评价值. 表中的部分参数变化曲线图如图 5～8 所示.

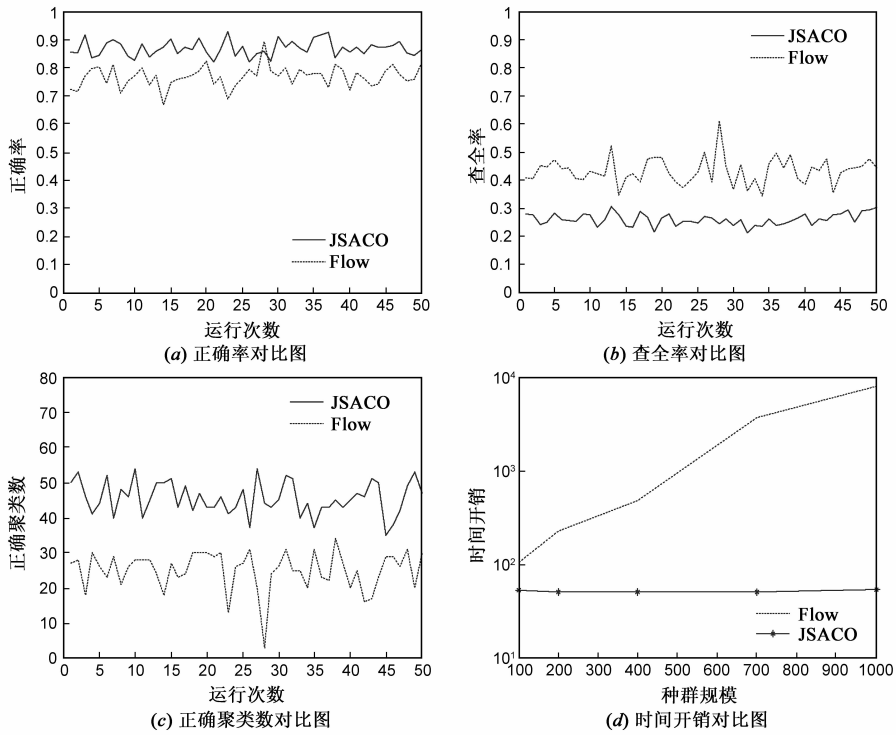
从以上结果中可以看出该算法具有稳定性，并且与标准结果库比对的匹配结果较好，正确率较高，平均正确率超过了 85%. 但查全率[17]相对较低，只达到了 26% 左右. 考虑到标准库中包含的蛋白质数只占到了数据中的蛋白质总数的 50% 以下，因此相对而言，正确率及正确聚类数更能客观地反映算法的聚类结果.

功能流聚类算法[3]在目前的 PPI 网络聚类算法中效率相对较好，其采用了信息流动的概念对数据进行聚类. 因此本实验还将 JSACO 算法与功能流聚类算法进行了比较. 算法的比较结果如表 4 所示.

其中 *F. score* 表示调和平均值，*Precision* 表示正确率，*Recall* 表示查全率，*Cluster* 表示正确聚类数，*time* 表示运行时间. 为了统一数量级，在计算过程中，正确聚类数及运行时间这两列的数据都统一缩小 100 倍，如表 4 中，在计算调和平均值 *F. score* 时，JSACO 算法的运行时间是 0.49s，而 Flow 算法的运行时间是 2.25s.

在图 9 中，实线代表 JSACO 算法在 50 次运行过程中对应的运行结果，虚线则是功能流(Flow)聚类算法运行 50 次的结果，从图 9(*a*)、(*b*)、(*c*)及表 4 可以看出，无论在正确率还是正确聚类个数上，JSACO 算法都要优于 Flow 算法，只是在与标准库比对时得到的查全率上有所不及. 但从几项评价指标的综合评价值——调和

平均值来看,无疑还是 JSACO 算法的性能更好.图 9(*d*)是 JSACO 算法和 Flow 算法的时间开销变化图.可见 JSACO 算法始终保持在一个较低的稳定的时间开销段中.



(*a*) 正确率对比图　(*b*) 查全率对比图

(*c*) 正确聚类数对比图　(*d*) 时间开销对比图

图9　JSACO算法和Flow算法对比

6　结论

本论文提出了一种基于连接强度的 PPI 网络聚类蚁群算法(JSACO).该算法采用拾起/放下规则的蚁群聚类算法,并根据 PPI 网络数据的结构特点,将结点与图的连接强度的概念融入到聚类算法中,对算法的拾起/放下规则加以改进,以降低蚁群算法的计算复杂度及提高对 PPI 网络数据的聚类正确率.在仿真实验中,对 PPI 网络数据进行了聚类分析,并将结果与标准库及功能流聚类算法进行了比较.结果表明 JSACO 算法在聚类结果的正确率上有所提升,时间开销有所下降,并在综合表现上比功能流聚类算法更好.但算法依然还有改进的空间,在后续的工作中将主要研究提高算法的查全率,以适应更为广泛的数据应用要求.

参考文献

[1] Aidong Zhang. Protein Interaction Networks [M]. New York: Cambridge University Press, 2009. 44 – 47.

[2] Jiawei Han, Micheline Kamber. Data Minining: Concepts and Techniques (Second Edition) [M]. San Francisco CA: Morgan Kaufmann Publishers, 2006. 223 – 232.

[3] Young-Rae Cho, Woochang Hwang, Murali Ramanathan, Aidong Zhang. Semantic integration to identify overlapping functional modules in protein interaction networks [J]. BMC Bioinformatics, 2007, 265(8): 1 – 13.

[4] 金弟,刘大有,杨博等.基于局部探测的快速复杂网络聚类算法[J].电子学报,2011,39(11):2540 – 2546.
Jin Di, Liu Dayou, Yang Bo, et al. Fast complex network clustering algorithm using local detection [J]. Acta Electronica Sinica, 2011, 39(11): 2540 – 2546. (in Chinese)

[5] Xiujuan Lei, Xu Huang, Aidong Zhang. Improved artificial bee colony algorithm and its application in data clustering [A]. Proceedings of IEEE BIC – TA2010 [C]. Beijing: IEEE Press, 2010. 514 – 521.

[6] Dorigo M, DiCaro G, Gambardella, LM. Ant algorithms for discrete optimization [J]. Artificial Life, 1999, 5(2): 137 – 172.

[7] Colorni A, Dorigo M, Maniezzo V. Distributed optimization by ant colonies [A]. Proceedings of the 1st European Conference on Artificial Life [C]. Amsterdam: Elsevier Publishing, 1991. 134 – 142.

[8] Jamaludin Sallim, Rosni Abdullah, Ahamad Tajudin Khader. ACOPIN: An ACO algorithm with TSP approach for clustering proteins from protein interaction network [A]. Proceedings of EMS'08 [C]. Beijing: IEEE Press, 2008. 203 – 208.

[9] Amita Lanjewar, Vaishali N Sahare, Nilesh Sahare. An approach based on clustering method for object finding mobile robots using ACO [A]. Proceedings of ICMLC 2010 [C]. Bangalore: World Academic Press, 2010. 161 – 165.

[10] M Dorigo, LM Gambardella. Ant colonies for the traveling salesman problem [J]. Bio Systems, 1997, 43(2): 73 – 81.
[11] 夏天扬.蚁群算法在聚类分析中的应用研究[D].湖北武汉:武汉理工大学,2010.5.
Xia Tian-yang. Research and Application of Ant Colony Algorithm on Clustering Aalysis [D]. Wuhan, Hubei: Wuhan University of Technology. 2010.5. (in Chinese)
[12] 李敏.蛋白质网络中复合物和功能模块挖掘算法研究[D].湖南长沙:中南大学,2008.12.
Li Min. Identifying Protein Complexes and Functional Models in Protein Interaction Networks [D]. Changsha, Hu'nan: Central South University, 2008.12. (in Chinese)
[13] Cui GU, Chen Y, Huang DS, et al. An algorithm for finding functional modules and protein complexes in protein-protein interaction networks [J]. Journal of Biomedicine and Biotechnology. 2008, (5): 1 – 10.
[14] Lumer E, Faieta B. Diversity and adaptation in populations of clustering ants [A]. Proceedings of the 3rd International Conference on Simulation of Adaptive Behavior: From Animal to Animals [C]. Cambridge MA: MIT Press, 1994. 499 – 508.
[15] U Güldener, et al. CYGD: the comprehensive yeast genome data base [J]. Nucleic Acids Research, 2005, 33(1): D364 – D368.
[16] 李敏,王建新,陈建二.基于距离测定的蛋白质复合物识别算法[J].吉林大学学报,2010,40(5):1318 – 1323.
LI Min, WANG Jianxin, CHEN Jianer. Distance measure-based algorithm for identification of protein complex [J]. Journal of Jilin University, 2010, 40(5): 1318 – 1323. (in Chinese)
[17] Dan Melamed, Ryan Green, Joseph P Turian. Precision and recall of machine translation [J]. Proceedings of HLT-NAACL2003 [C]. Stroudsburg, PA: Association for Computational Linguistics, 2003. 61 – 63.

作者简介

雷秀娟 女,1975 年 5 月出生于陕西长安.博士,副教授,硕士生导师,中国计算机学会(CCF)会员,访问学者(State University of New York at Buffalo, USA, 2009.10 – 2010.10).研究方向为智能计算与智能优化、生物信息计算等.
E-mail: xjlei168@163.com

黄 旭 男,1985 年 7 月出生于广东广州.陕西师范大学计算机科学学院 2008 级硕士研究生.从事群智能优化算法及蛋白质相互作用网络聚类算法方面的相关研究.
E-mail: ast86318307@hotmail.com

吴 爽 女,1987 出 6 月生于河南南阳.陕西师范大学计算机科学学院 2009 级硕士研究生.研究方向为数据挖掘、生物信息计算.
E-mail: persistencewu@126.com

郭 玲 女,1980 出 6 月生于陕西略阳.陕西师范大学生命科学学院 2011 级博士研究生.主要研究领域为生物信息学.
E-mail: glalguo2@163.com