生物物理学报 第二十卷 第五期 二〇〇四年十月
ACTA BIOPHYSICA SINICA Vol.20 No.5 Oct. 2004

蚂蚁群落优化算法在蛋白质折叠二维亲-疏水格点模型中的应用

李冬冬， 王正志， 杜耀华， 晏 春
（国防科技大学机电工程与自动化学院，长沙 410073）

摘要：氨基酸的亲疏水格点模型是研究蛋白质折叠的一种重要的简化模型，其优化问题是一个非确定型的多项式问题。采用蚂蚁群落优化算法对这一问题进行了研究，对测试数据的计算结果表明，在一定规模下，此算法能够有效地获得亲-疏水格点模型的最优解，其效率优于传统的 Monte Carlo 仿真等方法。

关键词：蛋白质折叠；HP 模型；蚂蚁群落优化算法；Monte Carlo 仿真
中图分类号：Q61

1 引 言

蛋白质的折叠问题是生物信息学中的一个重要问题，它研究蛋白质从一维序列到三维结构的过程。由于这一过程极其复杂，人们往往利用简化的模型来进行研究，亲-疏水格点（hydrophobic and polar monomers, HP）模型[1]就是 Dill 等人提出的一种最为重要，也是人们最常研究的简化模型。HP 模型认为疏水性作用是蛋白质折叠的最主要的相互作用，构成蛋白质的 20 种氨基酸按照其残基的疏水性不同被分成两类：疏水性（hydrophobic，H）基团和极性（polar，P）基团。在格点模型中，这些基团组成的序列在网格空间形成了一条自回避路径，而蛋白质的折叠过程则是在网格空间中建立一个自由能最低的构象的过程。人们已经证明，此优化问题是一个非确定型的多项式（nondeterministic polynomial, NP）问题[2]。

对 HP 模型，一般采用的方法大致可以分为进化算法（evolutionary algorithms，EAs）和 Monte Carlo（MC）仿真两类[2-4]，两者各有利弊，并没有一种算法在结果和效率上都能够明显强于其他所有算法。最近有人[5,6]提出了一种新的思路，把蚂蚁群落优化（ant colony optimization，ACO）算法用于二维 HP 模型，获得了良好的效果。其算法是在一般 ACO 算法的基础上引入局部搜索操作，以此来提高整体算法的效率。本文采用了一种不同的思路，把 ACO 算法看作是进化算法和 MC 算法的一种混合，通过引入克隆和淘汰操作，进一步提高了算法的效率。

2 算法与实现

蚂蚁群落优化算法是一种仿生优化算法，也称作蚂蚁算法（ant algorithm），其思想来源于蚂蚁的行为特性，是一种解决组合优化问题的多智能体算法[7]。其基本方法是使用一定数量的智能体（称为蚂蚁）反复地构造对问题的可能解，每一次构造的结果都会留下一些记忆，称为信息素（pheromone），而这种记忆将会以一定的概率方式指导下一次构造的过程。人们已经把 ACO 算法成功地应用到了一些组合优化问题上，包括 TSP 问题、网络路由问题等[8,9]。

在我们的算法中，这一构造过程可以分为两个循环过程：内部循环对所有的蚂蚁进行，每只蚂蚁都并行完成一次构造过程；外部循环对构造的次数进行，一次新的构象构造过程重新开始。很自然地，我们把内部循环看作是对一代蚂蚁的操作，而外部循环则是代数的循环。这样，对每一代蚂蚁，可以采用常用的 MC 仿真来计算，而在两代蚂蚁之间，则可以借鉴进化计算的思想，上一代计算的结果以某种形式影响下一代的计算。得到算法如下：

蚂蚁为给定的序列产生的候选构象，采用相对折叠方向（左转、前行、右转）来表示（如图 1 所

收稿日期：2003-12-22
基金项目：军队基础研究项目（JC-02-03-021）
通讯作者：李冬冬，电话：(0731)4574991，
E-mail：Li_dong_dong@yahoo.com.cn

示）。在前进第 i 步时，蚂蚁从当前的位置选择一种相对方向前进一步，到达一个新的位置。如果这个新的位置以前没有到达过，则成功前进了一步，蚂蚁将把序列上第 i+1 个字符（H 或者 P）放入这个新的位置，并开始下一步行走。反之，如果这个新的位置以前曾经到达过，那么构象构造失败。

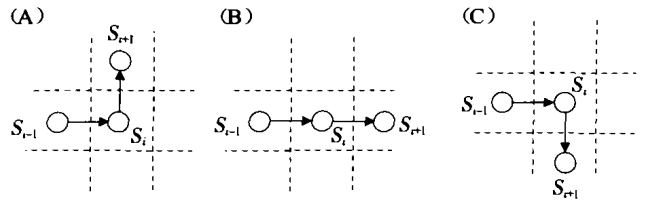

Fig.1 Relative folding directions. (A) Turn left; (B) Straight; (C) Turn right

图 1 中，从位置 S_i 出发，下一个位置有三种选择，以上一步前进的方向为基准，这三种选择可以分别称为向左、向前和向右，而不论其绝对方向如何。第一步没有相对方向，规定其绝对方向为向上，这一规定不影响计算结果。

在构造的过程中，每个位置的三种折叠方向上都有一定的初始信息素，蚂蚁在该位置前进的时候，对可以选择的每个相对方向（L、S 或 R）都能够获得两种信息：信息素的值（$\tau_{i,d}$，$d\in$ {L,S,R}）和启发式信息（$\eta_{i,d}$）。这里的启发式信息 $\eta_{i,d}$ 可以定义为新到达的位置（i+1）对整个系统自由能的贡献的函数，在一般的 MC 仿真中，此函数通常具有 Bolzmann 分布的形式，即：

$$\eta_{i,d}=\exp(-\Delta E_{i+1}/k_BT) \tag{1}$$

其中 k_B 为 Boltzmann 常数，T 为温度。在我们的算法中，也采用这一形式。信息素 $\tau_{i,d}$ 存在两种操作：耗散过程和增强过程。在计算过程中，信息素会以速率 ρ（$0<\rho<1$）减小，这是耗散过程，它表示系统记忆的消退。而当蚂蚁获得了一条成功的路径时，就会对所经过的每一条边的信息素进行增强。信息素更新的公式为：

$$\tau_{i,d}=\rho\tau_{i,d}+\Delta_{i,d,c} \tag{2}$$

式中 $\Delta_{i,d,c}$ 的数值是一个固定的增强常数值，在我们的算法中，它是该路径的自由能与理论上最低的自由能之间的比值。从而，蚂蚁在位置 i 选择下一个方向对方向的概率由以下公式决定：

$$p_{i,d}=\frac{\tau_{i,d}\eta_{i,d}}{\sum_{e\in\{L,S,R\}}\tau_{i,e}\eta_{i,e}} \tag{3}$$

由于蚂蚁构造候选构象的过程是一个自回避随机行走过程，因此，当构象比较紧密、序列比较长的时候，必然会出现行走进入到死路的情况，从而使得构造无效，这将导致计算效率的极大降低。因此，我们提出了克隆和淘汰的操作。即在第 i 只蚂蚁构造构象的过程中，如果在某一步构造时出现无效的构象，它将会被淘汰。同时，由于蚂蚁构造候选构象的操作是并行进行的，因此，我们可以按照某种概率选取第 j（$0<j<i$）只蚂蚁进行克隆操作，以这只克隆出来的蚂蚁代替第 i 只蚂蚁，继续运行。由于第 j 只蚂蚁在这一步构造中是有效的构象，这一操作能够有效地去除无效的构象。在实际操作中，如果出现 i=1 的情况，上述操作将会失效，因此，我们还记录了前一步构造中具有最佳自由能的蚂蚁（k），它也以一定的概率参与克隆操作，而且，对 i=1 的情况，直接对蚂蚁 k 进行克隆。

本文中使用的 HP 模型的完整 ACO 算法如下：

1）初始化路径上的信息素；

2）外部循环（循环，直到停止条件满足）：

(1) 初始化所有的蚂蚁；

(2) 内部循环（循环，如果没有到达序列的结束位置，作如下操作）：

①对每只蚂蚁，前进一步，并计算其当前的自由能；

②获得当前具有最佳自由能的蚂蚁；

③对每只蚂蚁，如果有必要，执行克隆和淘汰操作；

(3) 内部循环结束；

(4) 路径上的信息素耗散；

(5) 对获得较佳的自由能的蚂蚁，做路径信息素增强操作；

(6) 记录获得的最好结果；

3）外部循环结束；

4）算法结束。

3 结果与讨论

我们采用文献中通常使用的测试序列对算法进行了测试。测试序列（包括其已知的最佳构象的能量值）如表 1 所示：

Table 1 Test sequences

No.	Length	Energy	Sequences
1	20	−9	$(HP)_2PH_2PHP_2HPH_2P_2HPH$
2	24	−9	$H_2P_2(HP_2)_6H_2$
3	25	−8	$P_2HP_2H_2P_4H_2P_4H_2P_4H_2$
4	36	−14	$P_3H_2P_2H_2P_5H_7P_2H_2P_4H_2P_2HP_2$
5	48	−23	$P_2HP_2H_2P_2H_2P_5H_{10}P_6H_2P_2H_2P_2HP_2H_5$
6	50	−21	$H_2(PH)_3PH_4PHP_3HP_3HP_4HP_3HP_3HPH_4(PH)_3PH_2$
7	60	−36	$P_2H_3PH_8P_3H_{10}PHP_3H_{12}P_4H_6PH_2PHP$
8	64	−42	$H_{12}(PH)_2(P_2H_2)_2P_2H(P_2H_2)_2P_2H(P_2H_2)_2P_2HPHPH_{12}$
9	85	−53	$H_4P_4H_{12}P_6(H_{12}P_3)_3HP_2(H_2P_2)_2HPH$

我们在算法实现中的参数选择如下：蚂蚁的规模选为 500（这一规模的选择对算法的效率有一定的影响，但是不会影响结论的成立），信息素的耗散速率 ρ=0.6，计算 Δ_{i,d_c} 过程中用到的自由能的理论最小值由序列中的 H 残基的数目计算出来（序列内部的每个 H 残基被放入网格中时，自由能最多降低 2 个单位，两端的 H 残基最多导致自由能降低 3 个单位）。测试的结果如表 2，作为对比，我们还列出了其它文献中给出的对相同序列的测试结果，包括遗传算法（genetic algorithm，GA）、进化 MC 仿真（evolution monte carlo，EMC）和文献[6]的 ACO 算法的结果。其中 GA 和 EMC 算法测试的计算机配置未知，ACO[6]的 CPU 主频为 1 GHz，我们测试所用的 CPU 主频为 733 MHz，前 6 条序列运行时间是对 100 次运行计算的平均时间，其余的是 10 次计算的平均时间。

Table 2 Calculate result of test sequences

No.	Length	Energy	GA		EMC		ACO[6]		Our ACO	
			Optimum solution	Time(s)	Optimum solution	Time(s)	Optimum solution	Time(s)	Optimum solution	Time(s)
1	20	−9	−9	30 492	−9	9 374	−9	3.33	−9	0.25
2	24	−9	−9	30 491	−9	6 929	−9	2.52	−9	1.71
3	25	−8	−8	20 400	−8	7 202	−8	10.62	−8	50.00
4	36	−14	−14	301 339	−14	12 447	−14	11.81	−14	12.72
5	48	−23	−23	126 547	−23	165 791	−23	405.79	−23	51.11
6	50	−21	−21	592 887	−21	74 613	−21	4 953	−21	23.87
7	60	−36	−34	208 781	−35	203 729	−36	62 471	−36	2 898
8	64	−42	−37	187 393	−39	564 809	−42	5 845	−42	4 588
9	85	−53	---	---	−52	44 029	−51	21 901	−51	11 237

从表 2 中可以看到，与传统的 GA 和 MC 算法相比，ACO 算法在大部分情况下能够获得更好的运行效果，另外，文献[6]的 ACO 算法通过引入局部搜索操作来提高算法的效率，由于这种操作的代价比较大，因而对效率的提高是有限的。本文中则采用了克隆和淘汰操作，这两种操作本身的代价是相当小的，但是它们能够有效地克服在算法中出现的效率瓶颈。两者的测试结果对比表明，本文的算法对较长的序列获得了高于 ACO[6]算法的效率。

另外，从前面对算法的描述中我们可以看到，ACO 算法利用了两种信息（称为特征项）来指导蚂蚁的搜索，即信息素 $\tau_{i,d}$ 和自由能变化信息 $\eta_{i,d}$，它们的作用体现在公式（3）中。在这个公式中，如果我们把信息素 $\tau_{i,d}$ 始终保持为 1 不变，那么得到的就是传统的 MC 简单抽样公式；反之，如果把 $\eta_{i,d}$ 始终保持为 1，那么蚂蚁的搜索将完全由上

一代的搜索结果（通过影响信息素的分布）来指导：在上一代搜索结果中出现得多的那些局部路径（局部模式）将更有可能被选中（具有更多的信息素），这一点实际上具有遗传算法的特点。因此，针对 HP 模型的 ACO 算法可以看作是 MC 仿真方法和遗传算法的一种结合，由于它结合了两种算法的优点，因而有可能获得更高的效率。我们曾采用上述测试数据对这三种方法进行过测试，其结果也证明了这一点。

4 结 论

我们给出了一种改进的蚂蚁群落优化算法，通过引入克隆和淘汰操作，使算法的效率得到了较大的提高。对 HP 模型测试序列的计算表明了本文算法的有效性。我们认为，ACO 算法融合了进化计算和 MC 仿真两类算法的特点，从而有可能获得更好的效率。针对 HP 模型的 ACO 算法进一步的研究方向包括局部模式的表达，这是进一步提高算法效率的途径，也是我们的下一个研究方向。

参考文献：

[1] Lau KF, Dill KA. A lattice statistical mechanics model of the conformation and sequence space of proteins. *Macromolecules*, 1989,22(10):3986~3997

[2] Unger R, Moult J. Genetic algorithms for protein folding simulations. *J Mol Biol*, 1993,231(1):75~81

[3] Krasnogor N, Hart WE, Smith J, Pelta DA. Protein structure prediction with evolutionary algorithms. In: Banzhaf W, Daida J, Eiben AE, Garzon MH, Honavar V (Eds). GECCO-99: Proceedings of the genetic and evolutionary computation conference. Orlando, Florida, USA: Morgan Kaufman, 1999. 1596~1601

[4] Liang F, Wong WH. Evolutionary Monte Carlo for protein folding simulations. *J Chem Phys*, 2001,115(7):3374~3380

[5] Shmygelska A, Aguirre-Hernández R, Hoos HH. An ant colony algorithm for the 2D HP protein folding problem, In: Dorigo M, di Caro G, Sampels M(Eds). Proc of ANTS 2002. LNCS 2463: Springer, 2002. 40~52

[6] Shmygelska A, Hoos HH. An improved ant colony optimisation algorithm for the 2D HP protein folding problem. In: Xiang Y, Chaib-draa B(Eds). 16th conference of the canadian society for computational studies of intelligence. AI 2003, Halifax, Canada, LNCS 2671: Springer, 2003. 400~417

[7] Colorni A, Dorigo M, Maniezzo V. Distributed optimization by ant colonies, In: Proc first european conference on artificial life. Paris, France: Elsevier Publishing, 1991. 134~142

[8] Di Caro G, Dorigo M. AntNet: distributed stigmergetic control for communications networks. *Journal of Artificial Intelligence Research*, 1998,9:317~365

[9] Dorigo M, Colorni A. The ant system: optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 1996,26(1):29~41

APPLICATION OF ANT COLONY OPTIMIZATION ALGORITHM FOR 2D HYDROPHOBIC-POLAR PROTEIN FOLDING MODEL

LI Dong-dong, WANG Zheng-zhi, DU Yao-hua, YAN Chun

(*College of Mechatronics Engineering and Automation, National University of Defense Technology, Changsha 410073, China*)

Abstract: Protein folding problem is one of the most prominent problems in bioinformatics, and hydrophobic-polar model (HP model) is a wide abstractional model in study of this problem. A new algorithm of ant colony optimization (ACO) was proposed for the HP model's optimal problem, which was a non-deterministic polynomial problem (NP-hard problem). Two novel operations, clone and elimination, were added into the normal ACO algorithm, which improved the algorithm's computational efficiency greatly. Execution for standard benchmark instances indicated that the efficiency of this new algorithm is better than that of the existent algorithms, such as Monte Carlo algorithm and genetic algorithm.

Key Words: Protein folding; Hydrophobic-polar (HP) model; Ant colony optimization algorithm (ACO); Monte Carlo simulation