

文章编号: 1671-8836(2005)01-0033-06

# 改进的蚁群算法在 2D HP 模型中的应用

何莲莲<sup>1</sup>, 石 峰<sup>1†</sup>, 周怀北<sup>2</sup>

(1. 武汉大学 数学与统计学院, 湖北 武汉 430072; 2. 武汉大学 计算机科学学院, 湖北 武汉 430072)

**摘 要:** 针对蛋白质二维格模型(2D HP)折叠问题提出了一种改进的蚁群算法(Ant Colony Optimization Algorithm), 在算法的搜索阶段采用了牵引移动(pull moves)的方法: 首先按照一定规则移动一个或两个顶点的位置, 然后将其他顶点沿着链依次向前移动两个位置, 一旦达到一个新的有效构象则停止该移动. 该方法的优点是大多数移动只需改变很少的顶点位置, 使得改进后的蚁群算法具有较快的收敛速度. 求解基准实例的结果表明, 该算法在保证解的质量的前提下能大大缩短计算时间.

**关 键 词:** 蛋白质折叠; 格模型; 蚁群算法; 生物信息学

**中图分类号:** O 242.28

**文献标识码:** A

## 0 引 言

由意大利学者 Dorigo M. 等人首先提出的蚁群算法(Ant Colony Optimization Algorithm)是受到人们对自然界中真实的蚁群集体行为的研究成果的启发而提出的一种基于种群的模拟进化算法, 属于随机型智能搜索算法<sup>[1,2]</sup>, 它充分利用了蚁群搜索食物的过程与著名的旅行商问题(TSP)之间的相似性, 通过人工模拟蚂蚁搜索食物的过程(即: 通过个体之间的信息交流与相互协作最终找到从蚁穴到食物源的最短路径)来求解 TSP 问题. 自从蚁群算法在 TSP 问题<sup>[3]</sup> 和工作排序问题<sup>[4]</sup> 上取得成效以来, 已陆续渗透到其他类似问题领域中, 如图的着色问题<sup>[5]</sup>、车辆调度问题<sup>[6]</sup>、蛋白质折叠问题<sup>[7,8]</sup> 等等, 在许多方面表现出相当好的性能.

蛋白质是生命活动的重要承担者, 蛋白质的生物功能由其空间结构决定, 在天然蛋白质中, 蛋白质的最终构象是由氨基酸序列唯一确定的<sup>[9]</sup>. 蛋白质折叠问题就是确定一个给定氨基酸序列的蛋白质的天然结构的问题. 确定天然结构有一些物理化学方法, 如 X 射线晶体衍射、核磁共振等. 但是, 这些方法耗资巨大而且耗时耗力, 而且能测的结构也很受限制. 随着计算机技术的发展, 在计算机上进行蛋白质折叠模拟已经成为蛋白质结构预测的重要方法.

计算机模拟中用的最多的是一些简化模型, 而简化模型中最重要的一类就是格模型, 这些模型的共同特征为: ① 单体(或残基)用一个统一的尺寸代表; ② 键长是统一的; ③ 单体的位置被严格的限制在格位置上; ④ 简化的能量函数<sup>[10]</sup>. 然而即使是简化模型, 该问题仍然是 NP 困难的<sup>[11,12]</sup>.

用蚁群算法求解 2D HP 模型时, 在局部搜索阶段文献[7]采用了单点变异(point mutations)和宏变异(macro mutations)的方法, 该方法容易产生不可行构象; 而文献[8]则采用了大范围移动(long range move)的方法, 该方法的计算量较大. 本文结合文献[13]对此做了改进, 采用了牵引移动(pull moves)的方法, 该方法比较简单直观, 计算量小, 结果表明改进后的蚁群算法能够大大缩短运行时间.

## 1 蛋白质的 2D HP 折叠问题

格模型的种类很多, Dill 提出的疏水性-亲水性(HP)模型是最简单的一种, 也是应用较为广泛的一种<sup>[14,15]</sup>. HP 模型提出的理论基础是氨基酸的疏水性是小的球蛋白形成天然构象的主要驱动力<sup>[14]</sup>.

HP 模型中, 20 种氨基酸按疏水性和亲水性被分为两类: 疏水性残基(H)和极性残基(P), 因此蛋白质序列被抽象为一个由 H 和 P 组成的序列. 这个

序列根据自避免原则被限制在一个格上(本文考虑的是二维平方格). 图 1 是一个以 2D HP 模型表示的蛋白质构象. 图中所示序列为表 1 中序列 2.

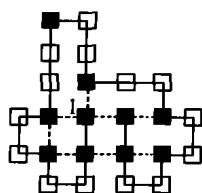


图 1 以 2D HP 模型表示蛋白质构象的例子

蛋白质序列为: HHPPHPPHPPHPPHPPHPPHPPH,  
黑色方格表示疏水性氨基酸, 白色方格表示极性氨基酸. 实线表示键长, 虚线表示计入能量的 H-H 接触

在 HP 模型中, 一个构象的能量定义为: 序列中并不相邻但是拓扑相邻的疏水氨基酸接触数目. 也就是说, 一个有  $n$  个 H-H 接触的构象  $c$  的自由能  $E(c) = n \cdot (-1)$ . 例如, 图 1 所示构象的能量为  $-9$ , 同时也是该序列的最优构象.

蛋白质的 2D HP 折叠问题正规提法如下: 给定一个氨基酸序列  $s = s_1, s_2, \dots, s_n$ , 找到  $s$  的最低能量构象, 也就是找到  $c^* \in C(s)$ , 使得  $E^* = E(c^*) = \min\{E(c) \mid c \in C\}$ , 其中  $C(s)$  是  $s$  的所有有效构象集. 蛋白质的天然构象是吉布斯(Gibbs)自由能最低的构象, 这是利用能量极小化方法预测蛋白质结构的热力学基础<sup>[16]</sup>.

表 1 求解 2D HP 蛋白质折叠问题的基准实例及其最低能量  $E^*$

序列编号	长度/个	$E^*$	蛋白质序列
1	20	-9	1H1P1H1P1P2H1P1H2P1H1P2H2P1H1P1H
2	24	-9	2H2P1H2P1H2P1H2P1H2P1H2P1H2P2H
3	25	-8	2P1H2P2H4P2H4P2H4P2H
4	36	-14	3P2H2P2H5P7H2P2H4P2H2P1H2P
5	48	-23	2P1H2P2H2P2H5P10H6P2H2P2H2P1H2P5H
6	50	-21	2H1P1H1P1H1P1H1P4H1P1H3P1H3P1H4P1H3P1H3P1H1P4H1P1H1P1H1P1H1P2H
7	60	-36	2P3H1P8H3P10H1P1H3P12H4P6H1P2H1P1H1P
8	64	-42	12H1P1H1P1H2P2H2P2H2P1H2P2H2P2H2P1H2P2H2P2H2P1H1P1H1P12H
9	85	-53	4H4P12H6P12H3P12H3P12H3P1H2P2H2P2H2P1H1P1H
10	100	-50	3P2H2P4H2P3H1P2H1P2H1P4H8P6H2P6H9P1H1P2H1P11H2P3H1P2H1P1H2P1H1P3H6P3H
11	100	-48	6P1H1P2H5P3H1P5H1P2H4P2H2P2H1P5H1P10H1P2H1P7H11P7H2P1H1P3H6P1H1P2H

## 2 用于求解 HP 模型的蚁群算法

对一个给定的 HP 序列, 由 ACO 算法中的蚂蚁构建候选解, 为了进一步提高解的质量, 按照一定的原则对候选解应用局部搜索策略, 并且根据解的质量进行信息素的更新, 框架如图 2.

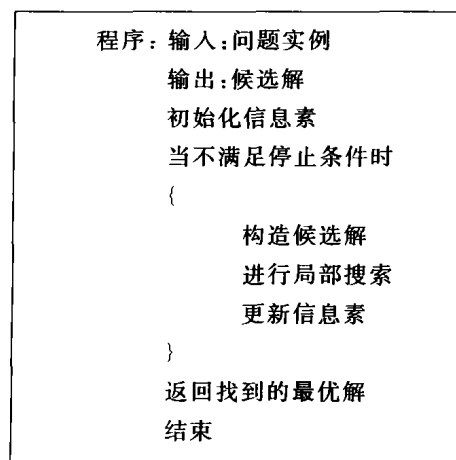


图 2 求解组合优化问题的蚁群算法框架图

候选解采取文献[17]中的局部结构基元表示: 直走(S)、左转(L)和右转(R), 在二维格上氨基酸的

位置分别如图 3 所示.

经过旋转的构象认为是不变的, 所以不失一般性的假设开始的两个氨基酸位置是固定的. 则可以通过  $n-2$  个局部结构基元表示一个长度为  $n$  的蛋白质序列的候选构象. 图 1 中的构象可以表示为序列: SRRLRRLSSRRSLRRLRRLRS.

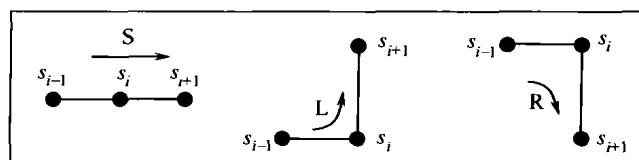


图 3 算法在解的构建和局部搜索阶段采用的局部结构基元

### 2.1 解的构建、信息素表示

ACO 算法的构建阶段, 每只蚂蚁随机的在给定序列中选择一个初始点. 从初始点开始, 序列同时向两个方向折叠, 每次增加一个氨基酸. 构建的每一步根据由期望值  $\eta_{i,d}$  和信息素(轨迹强度)  $\tau_{i,d}$  共同决定的概率选择折叠方向, 其中  $i$  是序列位置,  $d \in \{S, L, R\}$  是在位置  $i$  的折叠方向. 该方向与由三个连续的序列位置  $s_{i-1}, s_i, s_{i+1}$  组成的结构基元相对应, 这些

方向相当于旅行商(TSP)问题中城市之间的边。

当折叠方向是由位置  $i$  向位置  $i-1$  进行时信息素和期望值分别用  $\tau'_{i,d}$  和  $\eta'_{i,d}$  表示。算法中,  $\tau'_{i,L} = \tau_{i,R}$ ,  $\tau'_{i,R} = \tau_{i,L}$ ,  $\tau'_{i,S} = \tau_{i,S}$ ,  $\eta'$  的值依此类推。这些关系反映了折叠过程的一种基本对称的特征:从  $i$  到  $i+1$  扩展,把  $s_{i+1}$  置于  $s_i$  的右转(从  $s_{i-1}$  看)或者从  $i$  到  $i-1$  扩展,把  $s_{i-1}$  置于  $s_i$  的左转(从  $s_{i+1}$  看),得到的是  $s_{i-1}s_i s_{i+1}$  的同一局部构象。

期望值  $\eta_{i,d}$  将解的构建过程引向高质量候选解,也就是说,引向增大 H-H 接触数目的构象。本文中  $\eta_{i,d}$  是根据变量  $h_{i+1,d}$  定义的,向前折叠时(向后折叠类似)  $h_{i+1,d}$  是指将  $s_{i+1}$  置于  $s_i$  和  $s_{i-1}$  的方向  $d$  时新增加的 H-H 接触数目。需要注意的是,如果  $s_{i+1} = P$ ,则  $h_{i+1,S} = h_{i+1,L} = h_{i+1,R} = 0$ ;而且对于  $1 < i < n-1$  有关系式  $h_{i+1,d} \leq 2$ ,并且  $h_{n,d} \leq 3$ ;  $h_{i+1,d}$  的值通过检查二维平方格上  $s_{i+1}$  的 7 个邻位(显然,  $s_i$  位不需检查)很容易得出。为了保证对所有的  $i$  和  $d$  都满足  $\eta_{i,d} > 0$ ,定义  $\eta_{i,d} = h_{i+1,d} + 1$ ,这对保证在解的构建过程中不排除  $s_{i+1}$  的任何优先位子很重要。

在算法的构建阶段,假设构象已经扩展为  $s_k, \dots, s_i$ ,即将扩展  $s_{i+1}$ ,则  $s_{i+1}$  相对于  $s_{i-1}s_i$  的方向取决于下式由期望值和信息素决定的概率:

$$p_{i,d} = \frac{(\tau_{i,d})^\alpha (\eta_{i,d})^\beta}{\sum_{e \in \{L,R,S\}} (\tau_{i,e})^\alpha (\eta_{i,e})^\beta}$$

如果构象已扩展为  $s_i, \dots, s_m$ ,即将扩展  $s_{i-1}$ ,情况类似。

当蚂蚁从一个随机初始位置  $i$  (每只蚂蚁的初始位置的选择相互独立)开始折叠时,折叠随机的向两个方向进行扩展,而向每个方向扩展的概率等于该方向未折叠的残基数与总的未折叠的残基数的比值。

在解的构建阶段,尤其是对较长的蛋白质序列而言,经常会碰到不可行的构象。这种情况的发生是由于一个未完成的构象在折叠时因为所有的邻位都已经被其他氨基酸占据而不能继续扩展下去。本文在构建过程中遇到该情况时采取了一种回溯机制:短序列( $n \leq 64$ )回退一步重新开始构建过程,而较长序列( $n > 64$ )则回溯至已折叠的一半位置处重新开始构建过程。

## 2.2 局部搜索

局部搜索是蚁群算法中很重要的一部分。解的构建阶段结束后,选择一定数目的蚂蚁根据候选解分别进行局部搜索。文献[7,8]中所采取的局部搜索较为复杂,计算量大,本文对此作了改进,采用了文献[7]中提出的“牵引移动(pull moves)”方法。

首先定义几个概念:格上的一个节点称为一个位置(location),对应于一个坐标对  $(x, y)$ ;两个位置不管是水平相邻还是垂直相邻均称为相邻(adjacent);和同一个位置分别水平相邻和垂直相邻的两个位置称为对角相邻(diagonally adjacent)。链上的节点称为顶点(vertex),均有一个标签, H 或 P。链上的顶点有从 1 到  $n$  的连续标号。一个构象如果在格上的路径  $C$  遵守不交叉(non-self-intersecting)原则并且相邻的顶点占有相邻的位置则称为有效构象。为了方便描述该移动,构象将随时间变化:  $C(t)$  表示时刻  $t$  的构象。顶点  $i$  在时刻  $t$  的位置表示为  $(x_i(t), y_i(t))$ 。如果在时刻  $t$  一个位置没有被顶点占有则说其是自由的(free),称为空位。

假设顶点  $i$  在时刻  $t$  的位置为  $(x_i(t), y_i(t))$ ,空位  $F$  是  $(x_{i+1}(t), y_{i+1}(t))$  的邻位,同时也是  $(x_i(t), y_i(t))$  的对角相邻位置。具体可参考图 4。由图可知,  $(x_i(t), y_i(t)), (x_{i+1}(t), y_{i+1}(t))$  和空位  $F$  构成了一个正方形的三个角落,假设第四个角落为  $C$ 。只有满足以下两种情况之一才能进行牵引移动:  $C$  为空位;  $C$  恰好为  $(x_{i-1}(t), y_{i-1}(t))$ 。

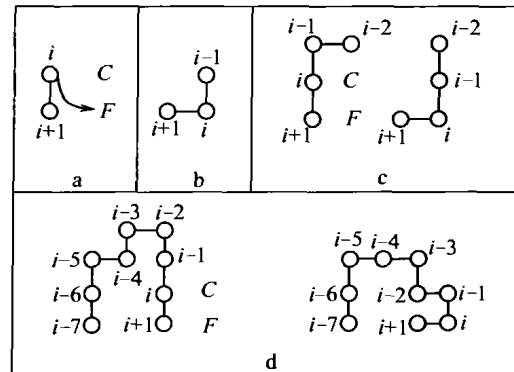


图 4 牵引移动详解

(a) 将顶点  $i$  移至空位  $F$ ; (b) 若  $C$  恰为顶点  $i-1$ , 移动完成; (c) 否则, 将顶点  $i-1$  移至空位  $C$ , 移动可能已完成; (d) 若移动仍未完成, 将顶点  $i-2$  移至顶点  $i$  先前位置, 将顶点  $i-3$  移至顶点  $i-1$  先前位置, 直到重新达到一有效构象

当  $C = (x_{i-1}(t), y_{i-1}(t))$  时, 只需把顶点  $i$  移至  $F$  即完成此次局部移动。当  $C$  是空位时, 首先把顶点  $i$  移至  $F$ , 把顶点  $i-1$  移至  $C$ , 然后在达到一个有效构象前进行以下移动: 从顶点  $j = i-2$  开始直到顶点 1, 置  $(x_j(t+1), y_j(t+1)) = (x_{j+2}(t), y_{j+2}(t))$ 。也就是说, 达到一个新的有效构象前, 所有的顶点均沿着链向前移动了两个位置。注意这样移动可以确保构象的有效性。

该移动的优点之一为: 只要达到一个有效构象, 就可以停止此次移动, 因此只需改变尽量少的顶点位

置. 实验中可以看出大多移动只需移动很少的顶点.

同样, 可以考虑另外一个方向的移动. 即从一个既是  $(x_{i-1}(t), y_{i-1}(t))$  的邻位, 也是  $(x_i(t), y_i(t))$  的对角相邻位置的空位  $F$  开始, 过程类似.

为了保证该移动可逆, 需要对端点做特别处理. 考虑一个路径的端点, 如果有两个空位, 并且其中一个为顶点  $n$  的邻位, 只需把顶点  $n-1$  和  $n$  移至这两个空位, 然后做以下移动: 从顶点  $j=n-2$  开始直到顶点 1, 置  $(x_j(t+1), y_j(t+1)) = (x_{j+2}(t), y_{j+2}(t))$ , 直到达到一个新的有效构象. 对端点 1 的处理类似. 文献[7]证明了该移动集是完全的, 并且是可逆的.

本文中, 每完成一个循环, 就按照精英策略 (elitist strategy) 选择一定数目 (蚂蚁数目的 1%) 的构象按上述原则实行局部搜索, 并且计算新构象的能量值, 如果优于原有的构象则接受该解; 否则放弃.

### 2.3 信息素的更新

每次结束一个循环并完成局部搜索后, 按照精英策略选择一定数目 (蚂蚁数目的 1%) 的构象按下式进行信息素的更新:

$$\tau_{i,d} \leftarrow \rho \cdot \tau_{i,d} + \Delta_{i,d,c}$$

其中  $0 < \rho \leq 1$ ,  $\rho$  表示信息素的衰减度,  $\Delta_{i,d,c}$  是和蚂蚁的候选构象  $c$  中第  $i$  个位置是否包括局部基元  $d$  有关的值, 若不包括则为 0, 为了防止有较大能量的序列在搜索阶段的过早停滞现象, 本文的取值为  $E(c)/E^*$ , 其中  $E^*$  是给定序列已知最小能量值或者是和序列中  $H$  数目相关的一个近似值.

## 3 实验结果

为了评价算法性能, 本文把改进后的蚁群算法应用于表 1 中的 11 个 2D HP 蛋白质折叠问题的基准实例, 这些实例在很多文献中已被广泛采用. 除非

特别指出, 本文中的实验参数为  $\alpha=1, \beta=2, \rho=0.8$ .  $n < 48$  的序列蚂蚁数目为 500 只,  $48 \leq n < 85$  的序列用了 1 500 只蚂蚁,  $n \geq 85$  的序列蚂蚁数目为 5 000 只. 选择 1% 的蚂蚁实行局部搜索和信息素的更新. 实验对每个序列执行的循环数是独立的: 对于较短的序列 ( $n \leq 25$ ), 如序列 1 到序列 3, 采取较少的循环数 (50 次), 每次循环后进行 2 000 次局部搜索即可在 2 s 内找出最优解.  $n > 25$  的序列循环数为 500 个, 进行 5 000 次局部搜索. 经过实验发现该算法对于序列 5 和序列 7 不是很稳定, 不管是增大循环次数还是增大局部搜索次数均不能保证找出最优解. 但是对于序列 5 在 50 s (500 次循环 5 000 次局部搜索) 的时间内均可找出次优解 (-22), 找出最优解 (-23) 的概率为 2%, 而对于序列 7 则可在 100 s (500 次循环 5 000 次局部搜索) 的时间内找出次优解——35. 具体运行时间和结果见表 2. 图 5 给出了本文算法找出的序列 5、序列 7 以及序列 9、序列 10、序列 11 的最优解和次优解.

运行时间以 CPU 时间测量, 所有的实验运行环境为: 2 GHz Pentium IV CPUs, 512 KB cache and 256 MB RAM. 其中原 ACO 算法循环次数分别为: 100 次 ( $n > 64$ ), 300 次 ( $50 < n \leq 64$ ) 和 500 次 ( $n \leq 50$ ), 蚂蚁数目为: 500 只 ( $n \leq 48$ ) 和 1 500 只 ( $n > 48$ ), 运行环境为: 1 GHz Pentium III CPUs, 256 KB cache and 1 GB RAM.

本文为了评价局部搜索的速度, 对于多个序列采用了不同的局部搜索次数, 发现局部搜索的次数对于时间的影响相对于循环数的影响要小一些. 如对于序列 10, 局部搜索次数增加 10 倍 (由 5 000 次到 50 000 次) 时, 运行时间为原来的 3 倍. 而循环次数增加一倍时运行时间即变为原来的 2 倍. 各参数对运行时间和结果的影响本文作者将作进一步研究.

表 2 改进的蚁群算法和原算法结果比较  
(其中  $t_{avg}$  为算法找到该解需要的平均时间)

基准实例			原 ACO 算法		本文中的 ACO 算法	
序号	长度/个	已知最低能量	最低能量	$t_{avg}$ / CPUs	最低能量	$t_{avg}$ / CPUs
1	20	-9	-9	3.33	-9	0.562
2	24	-9	-9	2.52	-9	0.859
3	25	-8	-8	10.62	-8	0.797
4	36	-14	-14	11.81	-14	11.641
5	48	-23	-23	405.79	-23	43.859
6	50	-21	-21	4 952.92	-21	52.156
7	60	-36	-36	62 471.24	-35	95.657
8	64	-42	-42	5 844.93(2 h)	-42	159.453
9	85	-53	-51	21 901.34(6 h)	-51	487.657
10	100	-50	-47	29 707.22(9 h)	-47	1 188.828
11	100	-48	-47	10 835.51(3 h)	-46	2 086.526

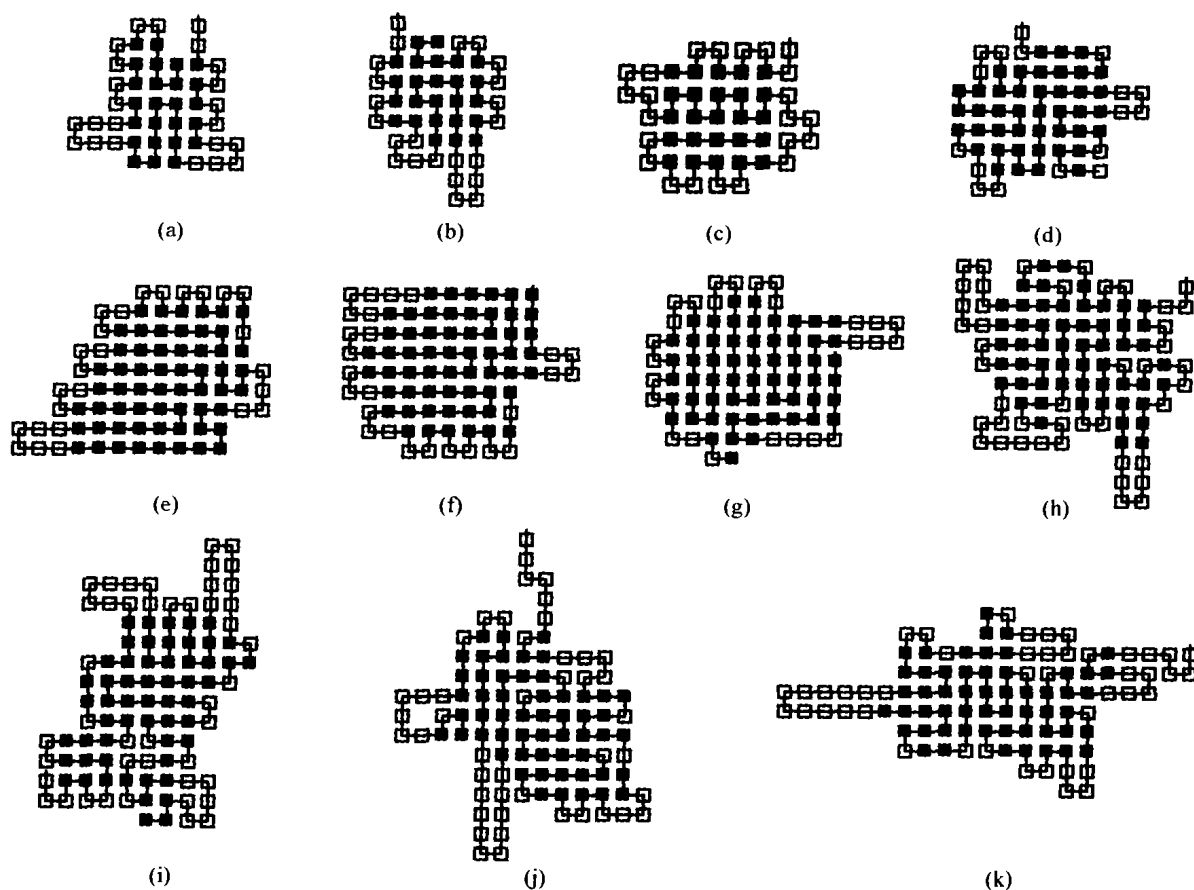


图 5 本文算法发现的序列 5、7、9、10、11 的次优解和最优解

(a),(b) 为序列 5 的两个次优解(-22); (c) 为最优解(-23); (d) 为序列 7 的次优解(-35); (e),(f),(g) 为序列 9 的三个次优解(-51); (h),(i) 为序列 10 的次优解(-47); (j),(k) 为序列 11 的两个次优解(-46)

## 4 结 论

本文在文献[7]的基础上改进了蚁群算法,再次表明该算法在求解 2D HP 模型方面良好性能。经过和原算法作比较,改进后算法的一个明显改进就是:在保证解的质量的前提下,运行时间大大缩短,尤其是对于较长序列。可见本文中采取的局部搜索确实可以改善算法性能。

如果考虑一种更简单的局部搜索机制可能对解决问题更有效,将蚁群算法应用于生物学中的其他问题也将是一件很有意义工作。

## 参考文献:

- [1] Dorigo M, Maniezzo V, Colormi A. The Ant System: An Autocatalytic Optimizing Process[R]. Technical Report No. 91-016, Italy, Milano: Dipartimento di Elettronica, Politecnico di Milano, 1991.
- [2] Dorigo M, Maniezzo V, Colormi A. The Ant System: Optimization by a Colony of Cooperating Agents[J]. *IEEE Transactions on Systems, Man, and Cybernetics-Part B*, 1996, **26**(1):29-41.
- [3] Dorigo M, Gambardella L M. Ant Colony System: A Cooperative Learning Approach to the Travelling Salesman Problem[J]. *IEEE Trans on Evolutionary Computation*, 1997, **1**(1):53-66.
- [4] Colormi A, Dorigo M, Maniezzo V, et al. Ant System for Job-Shop Scheduling[J]. *JORBEL*, 1994, **34**(1):39-53.
- [5] Costa D, Hertz A. Ants Can Colour Graphs[J]. *Journal of the Operational Research Society*, 1997, **48**(3):295-305.
- [6] Bullnheimer B, Hartl R F, Strauss C. Applying the Ant System to the Vehicle Routing Problem[A]. Voss S, Martello S, Osman I H, et al, eds. *Meta-Heuristics: Advances and Trends in Local Search Paradigms for Optimization*[C]. Boston: Kluwer, 1998, 109-120.
- [7] Alena Shmygelska, Holger H Hoos. An Improved Ant Colony Optimisation Algorithm for the 2D HP Protein Folding Problem[A]. Yang Xiang, Brahim Chaib-draa, eds. *Canadian Conference on AI, Lecture Notes in Computer Science*[C]. Berlin: Springer Verlag, 2003, 2671.

- [8] Shmygelska A, Hernandez R, HOOS H H. An Ant Colony Algorithm for the 2D HP Protein Folding Problem[A]. Marco Dorigo, Gianni Di Caro, Michael Sampels, eds. *Ant Algorithms, Lecture Notes in Computer Science* [C]. Berlin: Springer Verlag, 2002, 2463.
- [9] Anfinsen C B, Haber E, White F H. The Kinetics of the Formation of Native Ribonuclease During Oxidation of the Reduced Polypeptide Domain[J]. *Proc Natl Acad Sci USA*, 1961, **47**(9):1309-1314.
- [10] Backofen R. The Protein Structure Prediction Problem: A Constraint Optimization Approach using a New Lower Bound[J]. *Constraints*, 2001, **6**:223-255.
- [11] Berger B, Leighton T. Protein Folding in the Hydrophilic-Hydrophobic (HP) Model is NP-Complete[J]. *Journal of Computational Biology*, 1998, **5**(1):27-40.
- [12] Hart W E, Istrail S. Robust Proofs of NP-Hardness for Protein Folding General Lattices and Energy Potentials[J]. *Journal of Computational Biology*, 1997, **4**(1):1-22.
- [13] Neal Lesh, Michael Mitzenmacher, Sue Whitesides. A Complete and Effective Move Set for Simplified Protein Folding[A]. Miller W, Vingron M, Istrail S, et al. Eds. Annual Conference on Research in Computational Molecular Biology [C]. New York: ACM Press, 2003, 188-195.
- [14] Lau K F, Dill K A. A lattice Statistical Mechanics Model of the Conformation and Sequence Space of Proteins[J]. *Macromolecules*, 1989, **22**:3986-3997.
- [15] Dill K A. Theory for the Folding and Stability of Globular Proteins[J]. *Biochemistry*, 1985, **24**:1501.
- [16] Anfinsen C B. Principles that Govern the Folding of Protein Chains[J]. *Science*, 1973, **181**(4096):223-227.
- [17] Krasnogor N, Hart W E, Smith J, et al. Protein Structure Prediction with Evolutionary Algorithms [A]. Banzhaf W, Daida J, Eiben A, et al. Eds. *GECCO-99: Proceedings of the Genetic and Evolutionary Computation Conference* [C]. San Mateo, CA: Morgan Kaufmann, 1999, 1596-1601.

## Application of Improved Ant Colony Optimization Algorithm to the 2D HP Model

HE Lian-lian<sup>1</sup>, SHI Feng<sup>1</sup>, ZHOU Huai-bei<sup>2</sup>

(1. School of Mathematics and Statistics, Wuhan University, Wuhan 430072, Hubei, China;

2. School of Computer, Wuhan University, Wuhan 430072, Hubei, China)

**Abstract:** An improved Ant Colony Optimization (ACO) is proposed for the 2-dimensional hydrophobic-polar (2D HP) protein folding problem, we modified the local search mechanism by using pull moves: First one or two vertices was moved by rule, then, pull the other vertices two spaces up the chain until a new valid configuration is reached. The advantage is that most moves displace few vertices. It can quicken the convergence rate. Our experiments show that our algorithm can observably decrease computing time with the same result of previous ACO algorithm.

**Key words:** protein folding; lattice model; ant colony optimization (ACO) algorithm; bioinformatics