

基于改进的禁忌搜索的蛋白质三维结构预测

张晓龙, 程 文

(武汉科技大学计算机科学与技术学院, 武汉 430065)

摘 要: 禁忌搜索算法是一种局部搜索能力很强的全局迭代优化算法, 已经被成功地应用到各种组合优化问题中。基于 AB 非格模型, 该文将一种改进的禁忌搜索算法应用于蛋白质三维折叠结构预测。实验结果表明改进的禁忌算法求得的蛋白质三维最低能量构形的最低能量值比已有的算法求得的最低能量值要低, 同时三维构形中形成了一个疏水核, 被亲水残基包围, 反映了真实蛋白质的结构特征。该算法效率高, 可以有效地用于蛋白质三维折叠预测。

关键词: 禁忌搜索算法; 蛋白质三维折叠; AB 非格模型

Protein 3D Structure Prediction Based on Improved Tabu Search

ZHANG Xiao-long, CHENG Wen

(School of Computer Science and Technology, Wuhan University of Science and Technology, Wuhan 430065)

【Abstract】 Tabu Search(TS) algorithm is one of the global optimization search methods which has strong local search and is applied for a large number of combinatorial optimization problems. In the paper, an improved TS algorithm is proposed for protein Three-Dimensional(3D) folding structure prediction in AB off-lattice model. Experimental results show that the lowest energies computed by improved TS algorithm are better than those obtained by previous methods. Given a protein sequence, its lowest-energy conformation obtained by the improved TS forms a single hydrophobic core, which suggests that AB model in three dimensions appears to reflect real protein reasonably. Compared with previous heuristic approaches, improved TS algorithm has higher performance and can be effectively used to predict 3D structure prediction of proteins.

【Key words】 Tabu Search(TS) algorithm; protein 3D folding; AB off-lattice model

1 概述

蛋白质折叠结构的形状在很大程度上决定了其生物功能, 掌握蛋白质的结构信息对于研究蛋白质的功能和作用具有重要的意义, 然而蛋白质结构测定的速度远落后于其氨基酸序列预测的速度, 所以根据蛋白质的氨基酸序列预测其空间结构成为计算生物学领域的一个重要问题。

1961 年, Anfinsen 等提出了蛋白质的氨基酸序列及其环境决定其三维构象的著名论断, 并在 1973 年提出蛋白质天然构象对应能量最低的热力学假说, 这样使用理论计算方法从氨基酸序列预测蛋白质的天然结构成为可能。蛋白质折叠预测问题转化为 2 个难点问题: 一是如何归结出能更好地反映氨基酸残基间相互作用和环境等的数学模型; 二是根据热力学假设如何发展高效的搜索方法, 从蛋白质的势能表面上的多个局部极小值中搜索全局最小值。

由于真实蛋白质折叠结构的复杂性比较高, 理论研究者相继提出了许多简化模型, 其中以 Dill 等人根据疏水残基埋藏在蛋白质分子内部, 亲水残基暴露在蛋白质分子与水接触的表面这一蛋白质结构特点建立的 HP 格点模型最为著名, HP 格点模型把氨基酸分为疏水氨基酸(H)和亲水氨基酸(P) 2 类, 只考虑了空间相邻而序列不相连的 H 单体间的作用。1993 年 Stillinger 等人^[1]提出了 AB 非格点模型, 该模型不仅考虑了相邻 2 个键之间的作用, 而且还考虑了不相邻单体间的作用, 更能反映真实蛋白质的折叠问题。

禁忌搜索(Tabu Search, TS)的思想最早是由 Glover^[2]于 1986 年提出的, 是一种局部搜索能力很强的全局迭代寻优的智能搜索算法, 针对局部邻域搜索易陷入局部极小的特点,

禁忌搜索采用禁忌策略尽量避免迂回搜索, 从而保证对不同的有效搜索途径的探索, 跳出局部最优解。同时又采用藐视原则避免遗失优良状态, 激励对优良状态的局部搜索, 实现全局优化。

2 蛋白质结构预测的 AB 非格模型

Stillinger^[1]等人提出的 AB 非格模型将构成蛋白质的氨基酸根据疏水性和亲水性强弱分为 2 类: A(疏水性)和 B(亲水性), 即蛋白质序列简化为 2 个字符组成的序列。AB 非格模型最初被广泛应用于二维结构预测, 目前也被许多研究者用于蛋白质的三维结构预测, AB 非格模型作为三维结构预测的模型时, 除了考虑相邻键之间的键角外, 还需要考虑相邻 3 个键向量构成的 2 个平面间的扭转角。因此含 n 个残基的蛋白质的三维构型由 $n-2$ 个键角 $\theta_2, \theta_3, \dots, \theta_{n-1}$ 和 $n-3$ 个扭转角 $\beta_3, \beta_4, \dots, \beta_{n-1}$ 确定, 限定 θ_i, β_i 的范围分别为: $-\pi < \theta_i < \pi, -\pi < \beta_i < \pi$ 。 $\theta_i > 0, \beta_i > 0$ 表示逆时针旋转。

AB 非格模型中蛋白质折叠的能量包括 2 部分: 主链弯曲势能和不相邻残基间的 Lennard-Jones 形式的引力势能。前者与序列自身无关; 后者随着序列的变化而变化。 n 个残基的蛋白质序列的势能函数 E 表示如下:

$$E = \sum_{i=2}^{n-1} \frac{1}{4} (1 - \cos \theta_i) + \sum_{i=1}^{n-2} \sum_{j=i+2}^n 4 [r_{ij}^{-12} - C(\xi_i, \xi_j) r_{ij}^{-6}] \quad (1)$$

基金项目: 国家自然科学基金资助项目(60674115); 留学回国人员科研启动基金资助项目“生物信息处理中的机器学习算法研究”

作者简介: 张晓龙(1963-), 男, 教授、博士、博士生导师, 主研方向: 人工智能, 机器学习; 程 文, 硕士研究生

收稿日期: 2008-09-28 **E-mail:** cw1022@163.com

其中,

$$C(\xi_i, \xi_j) = \begin{cases} +1 & \xi_i = 1, \xi_j = 1 \\ +0.5 & \xi_i = -1, \xi_j = -1 \\ -0.5 & \xi_i \neq \xi_j \end{cases} \quad (2)$$

在三维空间中,相邻的氨基酸用单位长度的键相连, r_{ij} 是不相邻残基间的空间距离,与键角 θ 和扭转角 β 有关。 ξ_i 表示氨基酸单体的种类,若第 i 个残基属于 A,则 $\xi_i = 1$,若第 i 个残基属于 B,则 $\xi_i = -1$,从式(2)中可以看到,AA 对残基间有较强的引力,BB 对残基间有较弱的引力,AB 对残基间则有较弱的斥力,在一定程度上反映了真实蛋白质的特性。

AB 非格模型中,预测长度为 n 的蛋白质序列的三维结构,就是要找到 $n-2$ 个键角 $\theta_2, \theta_3, \dots, \theta_{n-1}$ 和 $n-3$ 个扭转角 $\beta_3, \beta_4, \dots, \beta_{n-1}$ 使得式(1)达到最小值,因此就转化为求目标函数 E 的约束最小值问题:

$$\min_{\theta_i, \beta_j \in (-\pi, \pi)} E(\theta_2, \theta_3, \dots, \theta_{n-1}; \beta_3, \beta_4, \dots, \beta_{n-1})$$

3 改进的禁忌算法

3.1 简单禁忌搜索的基本框架

禁忌算法的基本流程是:给定一个当前解和邻域结构,在当前解的邻域解中确定若干候选解;如果最佳候选解的目标函数值优于“best so far”状态,则忽视禁忌特性,用于代替当前解和“best so far”状态,并将其加入禁忌表中,同时修改禁忌表中各对象的任期,释放任期为 0 的对象;若不存在这样的候选解,则在候选解中选择非禁忌的最佳解为新的当前解,同时将其加入禁忌表中,并修改禁忌表中各对象的任期,释放任期为 0 的对象;如此重复上述过程,直至满足终止条件。

3.2 改进的策略

初始解、邻域函数、禁忌表、禁忌长度、候选解、藐视准则构成了禁忌搜索算法的关键因素。好的初始解可搜索到好的解,差的初始解会降低收敛速度。禁忌长度过短,所被禁忌的搜索方向就很有有限,可能造成循环搜索,禁忌长度过长,就会有搜索方向被禁忌,可能出现全禁忌的情况。候选集过小将容易造成早熟收敛,陷入局部最优。针对禁忌搜索的不足和蛋白质结构预测的特点,本文对禁忌搜索进行了改进,改进策略如下:

(1) 初始解 x_0 的选择

由于从不同的初始构形出发,得到的蛋白质最低能量值不同,为了获得比较好的能量值可能会选取多个初始解进行计算,考虑到随机性和计算时间问题,本文采用启发式方法产生初始解。在蛋白质的折叠结构中,疏水残基总是被亲水残基包围,形成一个稳定的疏水核,疏水核是蛋白质处于最低能量构形的关键因素,在此启发下,初始化时将疏水残基置于三维空间的中心位置,而将所有的亲水残基环绕着疏水残基,以此产生的构形作为初始构形。

(2) 解的表示形式

对于长度为 n 的蛋白质序列,势能函数 E 是关于 $(\theta_2, \theta_3, \dots, \theta_{n-1}; \beta_3, \beta_4, \dots, \beta_{n-1})$ 的最小约束值问题,所以每个解用一组键角值和扭转角表示,即表示为 $x(\theta_1, \theta_2, \dots, \theta_{n-2}; \beta_1, \beta_2, \dots, \beta_{n-3})$ 。用式(1)计算的解的能量表示为 $E(x)$ 。本文用 E_{\min} 表示全局最小能量值, x_{\min} 表示全局最小能量值对应的解(空间构象)。

(3) 邻域解产生

当前解的邻域解的产生和遗传算法中的变异操作相似,本文采用扰动变异的思想产生邻域解。对于当前解向量 $x(\theta_1,$

$\theta_2, \dots, \theta_{n-2}; \beta_1, \beta_2, \dots, \beta_{n-3})$,在算法搜索的前期为了保证有较好的多样性,采用多点变异,本文随机选取两位添加扰动产生邻域解;在算法搜索的后期采用单点变异,提高算法的收敛性。对于选择的随机位 $x^j(x^j$ 表示解向量的第 j 维)添加一个扰动值,为了保证邻域解有较好的多样性,添加了比例因子,新产生的位

$$x_{\text{new}}^j = x^j + f(r) \times 2\pi \times \text{random}(0 \cdots 1) \times \text{rate}^j$$

其中,系数 $f(r)$ 定义为

$$f(r) = \begin{cases} -1 & r < 0.5 \\ 1 & r \geq 0.5 \end{cases}, r \text{ 为 } [0,1] \text{ 之间的随机数}$$

rate 为比例因子,本文设置 rate 的值在 $[0.92, 0.96]$ 之间。 i 为迭代产生邻域解的迭代的次数,若邻域解的个数为 NL ,则每次产生邻域解集时, i 从 0 变化到 $NL-1$ 。

(4) 候选解集的选择

候选解集是当前邻域解的子集,本文设置候选集的长度(记为 CL)的范围为 $[4, 6]$ 的整数,依蛋白质序列的长度可能有所变化。选择候选集时,计算邻域解集中各个解的能量值,选择对应能量值最低的前 CL 个邻域解作为候选集。

(5) 禁忌表及其长度的设置

禁忌表是由当前解 x 之前的前 TL (禁忌表长度)个解构成的,禁忌长度是指禁忌对象在不考虑藐视准则的情况下不允许被选取的最大次数。本文设置了 2 个禁忌表:一个存放禁忌解向量 x ,另一个存放解向量对应的能量值 $E(x)$ 。为了提高算法的搜索性能,尽可能地避免搜索陷入死循环,本文设置禁忌表的长度是自适应动态变化的,设置禁忌表长度(TL)的变化区间为 $[8, 11]$ 。当前解附近极小解形成的“波谷”较深时,设置较大的禁忌表长度。

候选解被禁忌的准则设置如下:设候选解记为 $z(\theta_1, \theta_2, \dots, \theta_{n-2}; \beta_1, \beta_2, \dots, \beta_{n-3})$,对应的能量值为 $E(z)$,若禁忌表中存在一个解向量 $y(\theta'_1, \theta'_2, \dots, \theta'_{n-2}; \beta'_1, \beta'_2, \dots, \beta'_{n-3})$ (能量值为 $E(y)$),使得 $|E(y) - E(z)| \leq \psi$,并且两向量间的距离 $\|y - z\| \leq \eta$,则候选解 z 满足禁忌条件,被禁忌。依据实验情况,本文设置 $\psi = 0.08$, $\eta = 0.004$ 取得了比较好的结果。

(6) 全禁忌的处理方式

若候选解均被禁忌,且不存在优于“best so far”解的候选解,则称为全禁忌,此时下一个解就不能从当前候选集中获得,继续迭代也是徒劳的。针对此情况,采取了 2 种措施:其一是将全局最小能量值对应的解向量随机选择一位添加扰动,将新产生的解向量作为下一个解,这样既减少了不错失优良解的可能性,又提高算法的收敛性。其二是对候选解中的最佳候选解进行解禁,作为下一个解。这 2 种方法都取得了较好的实验效果。

3.3 改进禁忌算法的求解步骤

改进禁忌算法的求解步骤如下:

Step1 用 3.2 节中启发式方法产生初始解 $x_0(\theta_1, \theta_2, \dots, \theta_{n-2}; \beta_1, \beta_2, \dots, \beta_{n-3})$,用式(1)计算 x_0 的能量值 $E(x_0)$,令当前解 $x := x_0$,全局最优解 $x_{\min} := x_0$, $E(x) := E(x_0)$, $E_{\min} := E(x_0)$,初始化迭代次数 $k := 0$,最大迭代次数 L_{\max} ,邻域解个数 NL ,候选解个数 CL ,禁忌表长度 TL ,将禁忌表置空(2 个禁忌表分别用来存放禁忌解和解对应的能量值)。

Step2 若 $k > L_{\max}$,则结束算法并输出最优解 E_{\min} ;否则继续以下步骤。

Step3 若 $k < L_{\max} \times \delta$,其中 $0 < \delta < 1$,本文选取 $\delta = 0.85$,

则用 3.2 节介绍的两点变异的方法构造 NL 个当前解 x 的邻域解 x_i ($1 \leq i \leq NL$), 否则用单点变异的方法构造邻域解。

Step4 利用式(1)计算各个邻域解的能量值 $E(x_i)$ ($1 \leq i \leq NL$), 按照能量值从小到大的顺序重新排序 x_i 和 $E(x_i)$, 从排序后的邻域解集中选取前 CL 个解(对应能量值最低)作为候选解。从候选集中选择最佳解 x_1 , 对应能量值为 $E(x_1)$ 。

Step5 若 $E(x_1) < E_{\min}$, 则令 $x_{\min} := x_1, E_{\min} := E(x_1)$ (更新全局最优解)。

Step6 若 $E(x_1) < E(x)$, 则新的当前解 $x := x_1, E(x) := E(x_1)$, 然后转到 Step11; 否则转步骤 Step7。

Step7 令 $l := 1$, l 用来记录候选解的下标。

Step8 若 $l \leq CL$ (候选解个数), 则转 Step9; 否则转 Step10。

Step9 若 x_l 不满足 3.2 节中设置的禁忌条件, 则令新的当前解 $x := x_l, E(x) := E(x_l)$, 然后转 Step11; 否则计算 $l = l + 1$, 转到 Step8。

Step10 因出现全禁忌的情况, 按照 3.2 节中全禁忌的解决方法, 将全局最优解 x_{\min} 随机选择一位产生扰动, 产生新的解向量 x_{new} , 并计算 $E(x_{\text{new}})$, 然后令新的当前解 $x := x_{\text{new}}, E(x) := E(x_{\text{new}})$, 如果 $E(x_{\text{new}}) < E_{\min}$, 则 $x_{\min} := x_{\text{new}}, E_{\min} := E(x_{\text{new}})$ 。

Step11 将新的当前解 x 添加到存放禁忌解向量的禁忌表 Tabu list1 中, 对应的能量值 $E(x)$ 添加到存放能量值的禁忌表 Tabu list2 中。若禁忌表中的禁忌对象的个数超过了禁忌表长度, 则将最先进入禁忌表 1 的解向量移去, 同时移去禁忌表 2 中对应的能量值。

Step12 令 $k := k + 1$, 然后转 Step2。

4 实验结果与分析

为了验证在 AB 非格模型下, 本文采用的改进禁忌搜索算法是否能有效地搜索到最低能量构形, 采用许多文献中(文献[3-6])广泛使用的斐波纳契序列来预测, 其定义如下:

$$S_0 = A, S_1 = B, S_{i+1} = S_{i-1} * S_i$$

其中, “*” 是连接运算符, 前几条斐波纳契序列如 $S_2 = AB, S_3 = BAB, S_4 = ABBAB$ 等。由 $S_{i+1} = S_{i-1} * S_i$ 递推得出: 疏水残基(A)在序列中被孤立; 而亲水残基(B)在序列中或成对出现、或单独出现。

表 1 显示了在 AB 非格模型下, 本文采用的改进禁忌算法(TS)计算出的 4 条斐波纳契序列的三维最低能量值 E_{\min}^{TS} , 同时显示了文献 [3] 中 PERM(Pruned Enriched Rosenbluth Method)算法得到的最低能量值 E_{\min}^{PERM} 和 nPERM 算法的最低能量值 E_{\min}^{nPERM} , 文献[4]中 SA(Simulated Annealing)算法得到的最低能量值 E_{\min}^{SA} , 文献[5]中 ELP(Energy Landscape Paving Minimizer)算法得到的最低能量值 E_{\min}^{ELP} , 文献 [6] 中 CSA(Conformational Space Annealing)算法得到的最低能量值 E_{\min}^{CSA} 。本文采用 Java 语言实现算法, 表 1 中 N 代表序列的长度。从表 1 中可以看出, 对于 4 条序列, 本文采用改进的 TS 算法求得的最低能量值比 PERM, nPERM, SA 算法求得的最低能量值要低很多, 同时和目前用 ELP, CSA 算法计算出来的最好能量值相比, 对于前 3 个序列, 即 $N=13, N=21, N=34$ 的序列, 本文 TS 算法求得的最低能量值要优于这 2 种算法。除了 $N=55$ 的序列的最低能量值略差于 ELP, CSA 的最低能量值, 对于其他序列的计算结果都优于其他方法计算的结果, 特别是 $N=13, N=34$ 的序列, 本文计算出的最低能量值比 ELP 算法的结果都要优得多。所以本文提出的改进 TS 算法能有

效地预测蛋白质的三维结构。

表 1 AB 模型下不同算法求得的三维最低能量值

N	SEQUENCE	E_{\min}^{PERM}	E_{\min}^{nPERM}	E_{\min}^{SA}	E_{\min}^{ELP}	E_{\min}^{CSA}	E_{\min}^{TS}
13	ABBABBA BABBAB	-3.973 0	-4.961 6	-4.974 6	-4.967	-4.974 6	-6.568 7
21	BABABBA BABBABBA BABBAB	-7.685 7	-11.523 8	-12.061 7	-12.316	-12.326 6	-13.415 1
34	ABBABBABAB BABBABABBABA BBABBABABBAB	-12.860 1	-21.567 8	-23.044 1	-25.476	-25.511 3	-27.990 3
55	BABABBABAB BABBABABBAB BABBABABBAB BABABBABBA BBABBABBA	-20.107 0	-32.884 3	-38.197 7	-42.428	-42.341 8	-41.509 8

图 1~图 4 分别给出了表 1 中 4 条序列在 AB 非格模型下的三维最低能量构形。图中黑球表示疏水氨基酸(A), 白球表示亲水氨基酸(B)。从图中可以看出, 所有构形都形成了一个疏水核, 亲水残基在外面包围着, 这与真实蛋白质分子的结构性质是一致的。本文得出的三维最低构形与 PERM, nPERM, SA 得出的构形相比, 疏水残基更紧密地聚合在一起, 形成疏水核。另一方面, 对于序列 4 与 ELP 算法得出的最低能量构形基本一致。

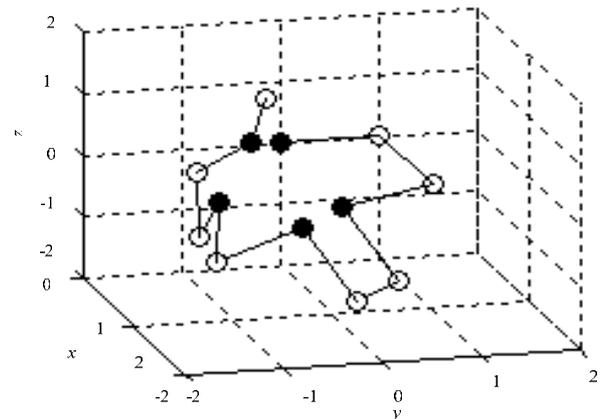


图 1 $N=13$ 的最低能量构形

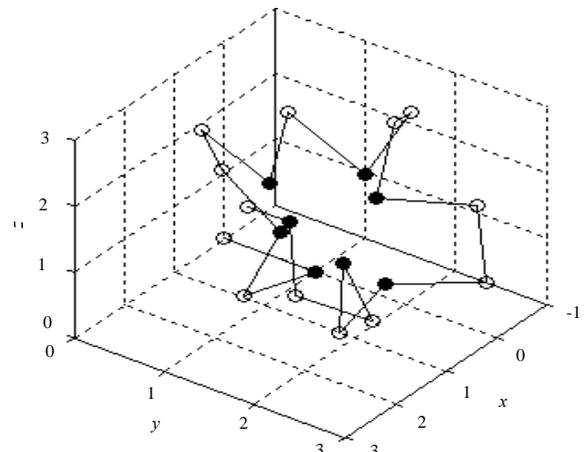


图 2 $N=21$ 的最低能量构形

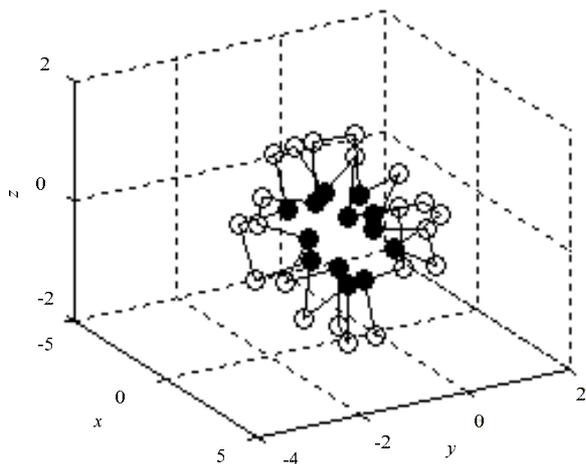


图3 N=34的最低能量构形

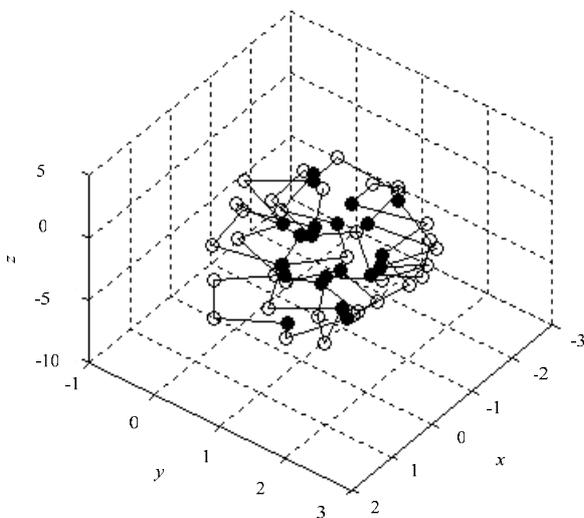


图4 N=55的最低能量构形

(上接第30页)

(5)机构类和职位类包含的 browse(浏览)方法均属于 EQ, 均包含 2 个 DET 和 1 个 FTR。同样, 工作地点类包含的 browse 方法也属于 EQ, 包含 4 个 DET 和 1 个 FTR。

综合以上过程, 并根据复杂度等级划分标准, 一共有 4 个 ILF, 功能复杂度均为 low, 功能点数为 $4 \times 7 = 28$; EIF 有 1 个, 功能复杂度也为 low, 对应功能点数为 5; 有 2 个 EI 的功能复杂度为 average, 对应功能点数为 $2 \times 4 = 8$; 另外, 9 个 EI 的功能复杂度为 low, 对应功能点数为 $9 \times 3 = 27$; 有 1 个 EO 的功能复杂度为 low, 功能点数为 4; 有 4 个 EQ 的功能复杂度为 low, 对应功能点数为 $3 \times 4 = 12$; 另有 1 个 EQ 的功能复杂度为 high, 对应功能点数为 6。因此, 可以得到总的功能点数为 90, 当然这是未经调整的功能点数。

5 结束语

功能点分析方法是用户能够理解的度量方法, 它可以在项目开发的早期较准确地度量出软件规模, 使客户和软件开发商能达成确保项目成功的共识和期望。

在软件项目开发中为节省维护阶段的成本和工作量, 本文通过分析 IFPUG FPA 方法的度量元素, 给出从 Java 源代

5 结束语

针对蛋白质结构预测模型多变量多极值的特点, 本文将禁忌搜索算法进行改进, 采取了一些改进策略: 用启发式的方法产生初始解, 用扰动的思想对当前解进行变异产生邻域解, 算法搜索前期和后期分别采用两点变异和单点变异, 保证搜索的多样性和收敛性, 采取自适应的动态禁忌表方式, 提高算法的搜索性能和鲁棒性。通过对 4 条斐波纳契序列进行预测, 改进禁忌算法得到的能量值优于 PERM, nPERM, SA 算法求得的最低能量值, 通过模拟 4 条序列的三维最低能量构形, 发现所有的构形中都形成了一个疏水核, 被亲水残基包围, 说明 AB 非格模型虽然简单, 却一定程度上反映了真实蛋白质的特性。

由于禁忌算法中迭代搜索过程是串行的, 仅是单一状态的移动, 因此降低了求解全局最优解的概率。下一步的工作是利用遗传算法并行搜索的特点, 将两者结合起来提高算法的搜索效率。

参考文献

- [1] Stillinger F H. Toy Model for Protein Folding[J]. Phys. Rev., 1993, 48(2): 1469-1477.
- [2] Glover F. Future Paths for Integer Programming and Links to Artificial Intelligence[J]. Computers and Operations Research, 1986, 13(5): 533-549.
- [3] Hsu H P, Mehra V, Grassberger P. Structure Optimization in an Off-lattice Protein Model[J]. Phys. Rev., 2003, 68(3): 31-34.
- [4] 陈 矛, 黄文奇. 求解蛋白质折叠问题的模拟退火算法[J]. 小型微型计算机系统, 2007, 28(1): 75-78.
- [5] Bachmann M, Arkin H, Janke W. Multicanonical Study of Coarse-grained Off-lattice Models for Folding Heteropolymers[J]. Phys. Rev., 2005, 71(3): 1-15.
- [6] Kim S Y, Lee S B, Lee Jooyoung. Structure Optimization by Conformational Space Annealing in an Off-lattice Protein Model[J]. Phys. Rev., 2005, 72(1): 61-66.

码中抽取数据功能点和事务功能点的计算规则, 并用实例给出具体的度量过程。这样, 软件项目管理人员可以在软件的整个生命周期内对软件项目进行跟踪和管理, 同时也为自动化度量工具的实现奠定坚实的理论基础。

参考文献

- [1] 吴 际, 汤铭端. 扩展功能点[J]. 软件学报, 2001, 12(2): 309-316.
- [2] Rossi P, Fernandez G. Definition and Validation of Design Metrics for Distributed Applications[C]//Proc. of the 9th Int'l Software Metrics Symposium. [S. l.]: IEEE Computer Society, 2003.
- [3] Richard L. A Model for Estimating the Size of a Formal Communication Protocol Specification and Its Implementation[J]. IEEE Trans. on Software Engineering, 2003, 29(1): 46-62.
- [4] Abran A, Silva I, Primera L. Fields Studies Using Functional Size Measurement in Building Estimation Models for Software Maintenance[J]. Journal of Software Maintenance and Evolution, 2002, 14(1): 31-64.