

文章编号:1671-9352(2007)12-0019-05

基于翻转和删除形式的染色体完美重组

张海燕,李国君

(山东大学 数学与系统科学学院, 山东 济南 250100)

摘要:研究了基于翻转和删除形式的染色体完美重组问题,并给出了多项式算法。

关键词:翻转和删除;完美重组;断点图;填充

中图分类号:TP301 **文献标志码:**A

Perfect sorting by reversals and deletions

ZHANG Hai-yan, LI Guo-jun

(School of Mathematics & System Science, Shandong University, Jinan 250100, Shandong, China)

Abstract: The problem of perfect sorting by reversals and deletions was considered, and one kind of polynomial algorithm was given.

Key words: reversal and deletion; perfect sorting; break point graph; padding

0 引言

近来,基于翻转(reversal)形式的染色体完美(perfect)重组问题引起越来越多学者的注意。Bérard等^[1]和Sagot等^[2]分别对基于翻转形式的染色体完美重组问题进行了研究,并相应地给出多项式算法。

但是,他们的算法要求重组前后的两个染色体有相同的基因,这种情况在生物实验中很少出现。通常,两个染色体完美重组前后会有不同基因。因此,在这种情况下,我们需要删除(deletion)操作。本文研究这种基于翻转和删除形式的染色体完美重组问题,并给出多项式算法。

设 $\pi = (\pi_1 \cdots \pi_{i-1} \pi_i \cdots \pi_j \pi_{j+1} \cdots \pi_n)$ 是一条含有 n 个基因的染色体,其中 $1 \leq |\pi_i| \leq n (1 \leq i \leq n)$,且对任意 $i \neq j, |\pi_i| \neq |\pi_j|$ 。 π_i 前面的符号表示基因 π_i 的方向。染色体内部的翻转是基因进化的一种重要形式。染色体内部的翻转 $\rho(\pi, i, j)$ 是指把 π 中的 $[i, j]$ 子序列逆向排列,转化为 $(\pi_1 \cdots \pi_{i-1} - \pi_j \cdots - \pi_i \pi_{j+1} \cdots \pi_n)$ 。用 $\pi \cdot \rho$ 表示染色体 π 经过翻转 ρ 后得到的新染色体。

称由连续基因组成的部分 $(\pi_a \cdots \pi_b), 1 \leq a \leq b \leq n$ 是染色体 π 的一个基因部分,简记为 $S = (\pi_a \cdots \pi_b)$,其中 π_a, π_b 是这部分的端点。若翻转 $\rho(\pi, i, j)$ 中的区间 $[i, j]$ 与 S 中的区间 $[a, b]$ 满足下列关系 $[i, j] \subset [a, b]$ 或 $[i, j] \supset [a, b]$ 或 $[i, j] \cap [a, b] = \emptyset$,则称翻转 $\rho(\pi, i, j)$ 没有破坏 S ,否则称 $\rho(\pi, i, j)$ 破坏 S 。若翻转序列 ρ_1, \cdots, ρ_t 中至少有一个翻转 $\rho_i (1 \leq i \leq t)$ 破坏 S ,则称翻转序列 ρ_1, \cdots, ρ_t 破坏 S 。

在生物进化过程中,若一组同源基因(他们有共同的祖先)在公共祖先中连在一起,在以后进化中也不分开,则称这种基因为保守末端(conserved segment)。若翻转 $\rho(\pi, i, j)$ 不破坏染色体 π 中的任何保守末端,

收稿日期:2007-04-05

基金项目:国家自然科学基金资助项目(10271065;60373025)

作者简介:张海燕(1977-),女,博士研究生,研究方向:组合最优化与理论计算机科学。Email:wszhaiyan@sohu.com

则称翻转 $\rho(\pi, i, j)$ 是完美的。基于翻转形式的染色体 π 和 γ 的完美重组问题是指要找到一系列的完美翻转序列 ρ_1, \dots, ρ_t , 使 $\pi \cdot \rho_1 \cdots \rho_t = \gamma$, 且 t 最小。这时, 我们称 t 为 π, γ 的完美翻转距离, 记为 $d(\pi, \gamma)$ 。在通常研究模型中, 取 $\gamma = (\gamma_1 \cdots \gamma_n) = (1 \cdots n)$, 且 $d(\pi, \gamma)$ 简记为 $d(\pi)$ 。此时, 染色体 π, γ 的完美重组问题简称为染色体 π 的完美重组问题。

1 基本定义

方便起见, 将标号的排列转化为无标号的排列, 其操作如下: 符号为“+”的元素 π_i 用有序顶点对 (π_i^-, π_i^+) 代替, 符号为“-”的元素 π_i 用有序顶点对 (π_i^+, π_i^-) 代替。通常, 还要为 π 增加两个顶点 $\pi_0 = 0^+$, $\pi_{n+1} = (n+1)^-$ 。断点图 $G(\pi)$ 的构造如下: 顶点集 V 由 π_0, π_{n+1} 以及 $\pi_i (1 \leq i \leq n)$ 的有序顶点对构成, 共 $2n+2$ 个顶点; 边集 E 包含黑边集和灰边集两部分。相应于 π_i 的有序顶点对中右边顶点与相应于 π_{i+1} 的有序顶点对中左边顶点连一条黑边。如果顶点 u, v 在 $G(\pi)$ 中连一条黑边, 那么称 u, v 在 $G(\pi)$ 中相邻。如果顶点 u, v 在 γ 中相邻, 那么 u, v 之间连一条灰边。因本文取 $\gamma = (1 \cdots n)$, 故根据灰边定义, 即应在 i^+ 与 $(i+1)^-$ 之间连一条灰边。在断点图 $G(\pi)$ 中, 记黑边条数为 $b(\pi)$, 圈的个数为 $c(\pi)$, 且圈 C 中黑边的条数为圈的长度。显然 $b(\pi) = n+1$, 且经过任何翻转操作后 $b(\pi)$ 的值保持不变。

对于作用在与灰边 g 相关联的两条黑边上的翻转 ρ , 若 $c(\pi\rho) - c(\pi) = 1$, 则称灰边 g 是定向的, 否则称其为不定向的。一个圈上若至少有一条定向的灰边, 则称这个圈是定向的, 否则称其为不定向的。

在染色体 π 的一个基因部分 S 中, 如果存在具有不同符号的元素 $\pi_i, \pi_j, i, j \in [a, b]$, 那么称 S 是定向的 (oriented), 否则称 S 为不定向的 (unoriented)。对于任意的 $i \in [a, b-1]$, 若满足 $\pi_{i+1} = \pi_i + 1$, 则称 S 被排好 (sorted)。若 $\pi_a > 0$, 则称 S 正向排好; 若 $\pi_a < 0$, 则称 S 负向排好。设 $m = \min_{i \in [a, b]} |\pi_i|, M = \max_{i \in [a, b]} |\pi_i|$ 。若满足关系式 $M - m = b - a$, 则称 S 为保守末端。根据保守末端的定义, 易知染色体 π 内每个基因 $\pi_i (0 \leq i \leq n+1)$ 以及染色体 π 本身都是保守末端, 这样的保守末端被称为平凡保守末端, 其余保守末端被称为非平凡保守末端。若保守末端 S 满足 $\pi_a = m, \pi_b = M$ 或者 $\pi_a = -M, \pi_b = -m$, 则称 S 为孤立的 (isolated)。若孤立保守末端 S 不是它所包含的孤立保守末端的并集, 则称 S 为染色体 π 的一个分支 (component)。

染色体 π 中的一个障碍 (hurdle) 是指染色体 π 中满足下列条件 (1) 或 (2) 的一个不定向分支 S : (1) S 是 π 中极小不定向分支, 即 π 中不存在不定向分支 W , 使得 $[W_{\min}, W_{\max}] \subset [a, b]$; (2) S 是 π 中最大不定向分支, 即 π 中任意不定向分支 W , 都有 $[W_{\min}, W_{\max}] \subset [a, b]$, 并且 S 不分离 π 中的任意两个不定向分支。这里我们用到的符号, $W_{\min} = \min_{\pi_i \in W} i, W_{\max} = \max_{\pi_i \in W} i$ 。

对于染色体重组问题, Figeac 等给出了一个基本引理^[3]。

引理 1^[3] 若存在一个翻转序列能重组染色体 π , 且不破坏染色体 π 的一个基因部分 S , 则一定存在一个具有相同个数的翻转序列, 该序列先排好 S 再排好染色体 π 中 S 外的部分。

在 [4] 中, Hannenhalli 首先给出了一个基础性的多项式算法。由 Hannenhalli 算法知, 求染色体的最小翻转序列时, 有些翻转不得不破坏一些孤立保守末端, 即 [4] 中所谓的“障碍合并 (hurdle merge)”。因此, Bergeron 等给出下面引理:

引理 2^[5] 一个染色体 π 不存在完美翻转序列当且仅当染色体 π 中有三个不定向分支 A, B, C 满足下列关系之一 (1) $A \subset B \subset C$; (2) $B \subset A, C \subset A$ 且 $B \cap C = \emptyset$; (3) $|C \cap B| \leq 1$ 且 A 包含于 B 或 C 内。

称满足引理 2 的染色体为傻的 (fool)。显然, 一个傻的染色体没有完美翻转序列。因此, 本文总是假设染色体 π 不是傻的。设 $\mu(\pi)$ 为染色体 π 中不定向分支个数, 由 Hannenhalli 定理和引理 2 得

定理 1^[4] 若染色体 π 不是傻的, 则 $d(\pi) = b(\pi) - c(\pi) + \mu(\pi)$ 。

2 基于翻转和删除形式的染色体重组

从本节开始, 设染色体 π 由基因 $\pi_1, \pi_2, \dots, \pi_n$ 和 A_π 组成, 其中 A_π 为染色体 π 中要被删除的基因集合, 且

A_π 中的基因可以在 π 中任意位置。现在,回到本文要解决的问题,即通过翻转和删除操作把染色体 π 完美重组为染色体 γ ,且 π 和 γ 有不同的基因。记 $\bar{\pi} = \pi \setminus A_\pi$ 。显然, $\bar{\pi}$ 与 γ 有相同的基因。因此,可以用 Hannenhalli 方法通过翻转操作把 $\bar{\pi}$ 转化为 γ 。

在染色体 π 中,给所有要被删除的基因作上标记,以便与 $\bar{\pi}$ 中基因区分开。方法如下:设 $S = (\pi_i \pi_{i+1} \cdots \pi_{j-1} \pi_j)$ ($1 \leq i < j \leq n$) 为染色体 π 的一基因部分,其中 π_i, π_j 属于 $\bar{\pi}$,而 π_k ($i+1 \leq k \leq j-1$) 属于 A_π 。可以用 $S' = (\pi_i \delta(\pi_i) \pi_j)$ 来代替 S ,其中 $\delta(\pi_i)$ 为夹在 π_i, π_j 中间的部分,且称 π_i, π_j 被 $\delta(\pi_i)$ 分开。相应地,断点图 $G(\pi)$ 也发生了变化,记为 $G'(\pi)$ 。 $G'(\pi)$ 构造如下:顶点集 V 包含染色体 $\bar{\pi}$ 中所有基因 π_i 对应的顶点 π_i^+ , π_i^- 以及按照惯例增加的两个顶点 $(\min_{\pi_i \in \bar{\pi}} |\pi_i| - 1)^+$ 和 $(\max_{\pi_i \in \bar{\pi}} |\pi_i| + 1)^-$;边集 E 包含黑边集和灰边集,黑边集又包含直接黑边和间接黑边两部分。在 π 中,直接相邻的顶点 u, v 所连的黑边为直接黑边,被 δ 分开的两个顶点所连的黑边为间接黑边。与 $G(\pi)$ 的构造相同,在 γ 中相邻的两个顶点连一条灰边。若圈 C 中含有间接黑边,则称圈 C 为间接的,否则称圈 C 为直接的。

下面定义由黑边和灰边确定的翻转 ρ 。设 $e = (\pi_i, \pi_j)$, $f = (\pi_s, \pi_t)$ ($i < j < s < t$) 为两条黑边。若黑边 e, f 都是间接的,则由 e, f 所确定的翻转 ρ 翻转 $[\pi_j, \delta(\pi_s)]$ 部分,即从 π_j 到与 π_t 相邻的部分;否则翻转 $[\pi_j, \pi_s]$ 部分。设 $g = (\pi_j, \pi_t)$ 为一条定向灰边,若黑边 e, f 为间接的,则由灰边 g 所确定的翻转 ρ 翻转 $[\pi_j, \delta(\pi_s)]$ 部分;否则翻转 $[\pi_j, \pi_s]$ 部分。由上面定义,易得下面引理:

引理 3^[6] 长度为 m 的定向圈 C ,利用 Hannenhalli 方法进行 $m-1$ 次翻转后,转化为 m 个长度为 1 的圈。若 C 为直接圈,则 m 个圈全都是直接圈,否则 m 个圈中仅有一个为间接圈。

由引理 3,对于定向圈 C ,通过 Hannenhalli 方法可以使要被删除的基因连在一起,即把断点图 $G'(\pi)$ 中多条间接黑边连成一条间接黑边,最后只需一次删除就可以把圈 C 中所有要被删除的基因删掉。对于不定向圈形成的障碍,进行障碍合并,即把障碍 H_1 中的不定向圈 C_1 和障碍 H_2 中的不定向圈 C_2 进行合并,这样就可以去掉两个障碍,产生一个定向圈。

根据黑边定义,障碍分为直接障碍和间接障碍。若障碍 H 中至少含有一条间接黑边,则称 H 为间接障碍,否则称 H 为直接障碍。对于障碍合并问题,Nadia 给出一个算法^[6]。把断点图 $G'(\pi)$ 的障碍分成直接障碍集 $Dh(\pi)$ 和间接障碍集 $Ih(\pi)$,即 $I_1, D_1, \cdots, I_p, D_p$,其中 I_i ($1 \leq i \leq p$) 为连续间接障碍的最大集,它以连续直接障碍最大集 D_{i-1} 和 D_i 为边界。为了使用尽可能少的删除操作,尽量选取间接圈进行障碍合并。Nadia 在[6]中给出的间接障碍合并算法如下:

算法 1

Step 1 对于所有 i ($1 \leq i \leq p$)

若 I_i 中至少有三个间接障碍 $I_{i1}, I_{i2}, \cdots, I_{ij}$ ($j \geq 3$),则

对于 $k \in \{1, 3, \cdots, j-1 - (j \bmod 2)\}$,合并 I_{ik} 和 $I_{i(k+1)}$ 。

(Step 1 结束后,还剩 p 个含有 1 个或 2 个间接障碍的障碍集 I'_i ($1 \leq i \leq p$))。

Step 2 如果 $p > 1$

设 I'_1, \cdots, I'_r 表示由 Step 1 得到的含有 1 个间接障碍的间接障碍集,

对于 $l \in \{1, 3, \cdots, r-1 - (r \bmod 2)\}$,合并 I'_l 和 I'_{l+1} 。

设 I''_1, \cdots, I''_s 表示由 Step 1 得到的含有 2 个间接障碍的间接障碍集,

对于 $1 \leq l \leq s$,分别在 I''_l 和 I''_{l+1} 中的任取一个障碍进行合并。

Step 3 否则 $p = 1$

如果 I_1 中含有 2 个间接障碍,则合并这两个障碍。

3 主要算法

由保守末端定义知,在染色体 π, γ 完美重组过程中,非平凡保守末端不会被删除。根据引理 1 知,我们可以先排好染色体 π 中的每个非平凡保守末端,然后再排好其余部分。

设 $S_1 = (\pi_{a_1} \cdots \pi_{b_1})$, $S_2 = (\pi_{a_2} \cdots \pi_{b_2})$ 为两个保守末端(非平凡),若区间 $[a_1, b_1]$, $[a_2, b_2]$ 满足 $[a_1, b_1] \subset [a_2, b_2]$ 或 $[a_1, b_1] \supset [a_2, b_2]$ 或 $[a_1, b_1] \cap [a_2, b_2] = \emptyset$, 则称这两个保守末端没有交叠(overlap)部分。如果一个保守末端与其它任何保守末端没有交叠部分,则称这个保守末端是强的(strong)。根据强保守末端定义,可设保守末端 $S = S_1 \cup S_2 \cup \cdots \cup S_k$, 其中 $S_i (1 \leq i \leq k)$ 为强保守末端。设强保守末端 S_i, S_j 为保守末端 S 的任意两个子集,因为 S_i, S_j 都是保守末端,所以 S_i 中所有元素的绝对值都大于(或都小于) S_j 中所有元素的绝对值,我们将其记为 $S_i > S_j$ (或 $S_i < S_j$)。根据 $S_i (1 \leq i \leq k)$ 的大小关系,可以对强保守末端 S_i 排序。每个子集 S_i 都对应一个次序号 $c_i (1 \leq c_i \leq k)$, 因此可以得到一个次序序列 (c_1, c_2, \cdots, c_k) 。例如若 S_i 是所有子集中最大(小)一个,则其次序号 $c_i = k (c_i = 1)$ 。若强保守末端 S_i 不包含其它的非平凡保守末端,则称 S_i 是最小强保守末端。显然,强保守末端 $S_i (1 \leq i \leq k)$ 还可以继续划分成若干个最小强保守末端以及平凡保守末端的并集,即设 $S_i = S_{i1} \cup S_{i2} \cup \cdots \cup S_{il}$, 其中 $S_{ij} (1 \leq j \leq l)$ 为最小强保守末端或含有单个基因的平凡保守末端。

为了找到强保守末端 S_i 的完美翻转序列,首先通过填充(padding)操作把 S_i 转化为孤立强保守末端。设强保守末端 $S_i = (\pi_i \cdots \pi_j) (1 \leq i < j \leq n)$, S_i 的填充就是给 S_i 增加一个元素 π_k , 且满足 $u < k < u + 1 (i - 1 \leq u \leq j)$, $v < |\pi_k| < v + 1$, 其中 u, v 为已知正整数。若经过填充得到的染色体 π' 比染色体 π 增加一个圈,且没有新的不定向分支生成,即满足 $d(\pi) = d(\pi')$, 则称这个填充为安全的(safe)。设 $m = \min_{\pi_i \in S_i} |\pi_i|$, $M = \max_{\pi_i \in S_i} |\pi_i|$, 如果可以用 $i - 1 < k_1 < i$, $m - 1 < \pi_{k_1} < m$ 和 $j < k_2 < j + 1$, $M < \pi_{k_2} < M + 1$ 同时对 S_i 进行安全填充,那么称 S_i 有正向填充(positive padding)。如果可以用 $i - 1 < k_1 < i$, $M < -\pi_{k_1} < M + 1$ 和 $j < k_2 < j + 1$, $m - 1 < -\pi_{k_2} < m$ 同时对 S_i 进行安全填充,那么称 S_i 有负向填充(negative padding)。关于强保守末端的填充,有下面重要引理:

引理 4^[2] 对染色体 π 中的强保守末端 S_i 进行填充,只有两种可能情况:(1) S_i 存在惟一正向或负向填充;(2) S_i 不存在正向且不存在负向填充。若(2)成立,则 π 没有完美翻转序列。

下面给出通过翻转操作完美重组保守末端 $S = S_1 \cup S_2 \cup \cdots \cup S_k$ 的算法。

算法 2

Step 1 对所有 $i (1 \leq i \leq k)$, 若强保守末端 S_i 没有被排好,则进行下面操作

(1) 若 S_i 为不定向的,则在不破坏 S_i 内任何保守末端的情况下,选择一条灰边进行翻转,使 S_i 成为定向的。否则转入(2)。

(2) 对 S_i 按如下进行划分。在 S_i 中,把交叠的圈看成一部分,且按每一部分最左边的顶点次序排号。不失一般性,设将 S_i 共分为 $S_{i1}, S_{i2}, \cdots, S_{il} (1 \leq l \leq n)$ l 部分。转入(3)。

(3) 在 S_i 的 l 部分中,任取一个非平凡极小保守末端 $S_{ij} (1 \leq j \leq l)$ 。若 S_{ij} 被排好,则转到(5)。否则进行下面的(4)。

(4) 对 S_{ij} 进行填充。若 S_{ij} 不存在正向且不存在负向填充,则染色体 π 不存在完美翻转序列。否则,把 S_{ij} 填充为孤立保守末端,利用 Hannenhalli 方法排好 S_i 。转入(5)。

(5) 从 S_{ij} 出发,分别向两侧寻找 $S_{i(j-p)} \cup \cdots \cup S_{i(j-1)} \cup S_{ij} \cup S_{i(j+1)} \cdots \cup S_{i(j+t)}$, 使 $S_{i(j-p)} \cup \cdots \cup S_{i(j-1)} \cup S_{ij} \cup S_{i(j+1)} \cdots \cup S_{i(j+t)}$ 为包含 S_{ij} 最小的一个强保守末端。转入(6)。

(6) 若 $S_{i(j-p)}, \cdots, S_{i(j-1)}, S_{ij}, S_{i(j+1)}, \cdots, S_{i(j+t)}$ 中有非平凡最小强保守末端,则分别对其中的非平凡最小强保守末端,转入(4)进行填充,然后转入(7)。否则,直接转入(7)。

(7) 给每个 $S_{ik} (k = j - p, \cdots, j - 1, j, j + 1, \cdots, j + t)$ 一个次序号 $c_{ik} (1 \leq c_{ik} \leq p + t + 1)$ 。若 S_{ik} 正向排好,则在 c_{ik} 前加 +; 若 S_{ik} 负向排好,则在 c_{ik} 前加 -; 若 S_{ik} 为平凡强保守末端,则取其本身符号。以带有 + 或 - 的次序号 c_{ik} 为顶点,建立断点图,利用 Hannenhalli 方法将 $S_{i(j-p)} \cup \cdots \cup S_{i(j-1)} \cup S_{ij} \cup S_{i(j+1)} \cdots \cup S_{i(j+t)}$ 排好。

(8) 以 $S_{i(j-p)} \cup \cdots \cup S_{i(j-1)} \cup S_{ij} \cup S_{i(j+1)} \cdots \cup S_{i(j+t)}$ 为起点,重复(5)~(7)步,直到 $j - p = 1, j + t = l$ 。

Step 2 若 S_i 排好,以 S_i 为起点,重复 Step 1 中的(5)~(8),排好保守末端 S 。

下面我们给出主要算法。

主要算法

Step 1 在断点图 $G'(\pi)$ 中,若任一保守末端 S 中的黑边被包含在长度大于 1 的圈中,执行算法 2。否则转入 Step 2。

Step 2 若 $G'(\pi)$ 中含有定向分支,则在定向分支中取一条定向灰边进行翻转,直到每个定向分支中每个圈的长度为 1。否则转入 Step 3。

Step 3 对于间接障碍集 Ih ,执行算法 1;对于直接障碍集 Dh ,尽可能选取不相邻的障碍进行合并。否则,若剩余一个直接障碍和一个间接障碍,则合并这两个障碍;若剩余一个直接或间接障碍,则在其内部进行翻转,使其转化为定向分支。

Step 4 (删除操作)对于每个间接黑边 $e = (a, b)$,删除中间标有 $\delta(a)$ 的部分。

设染色体 π 中所有非平凡保守末端组成的集合为 π_S ,且 $\pi \setminus \pi_S, \gamma \setminus \pi_S$ 分别表示 π, γ 去掉 π_S 中基因所剩染色体部分。设在算法 1 中执行障碍合并的操作数为 $\alpha(Ih(\pi \setminus \pi_S))$,算法 2 中排好所有非平凡保守末端所需要的翻转数为 $\beta(\pi_S)$,在主要算法中所需要的删除操作数为 $d(A_\pi)$ 。我们用 $d(\pi \setminus \pi_S)$ 表示 $\pi \setminus \pi_S$ 转化到 $\gamma \setminus \pi_S$ 所需要最小的翻转数。我们有下面定理

定理 2 完美重组染色体 π 所需要最小的操作数为 $d(\pi) = \beta(\pi_S) + d(\pi \setminus \pi_S) + d(A_\pi)$ 。

证明 在利用算法 2 排好染色体 π 中所有的非平凡保守末端过程中以及在主要算法 Step2 ~ Step4 中,我们的翻转和删除操作都没有破坏任何非平凡保守末端,故找到的重组序列为完美序列。

在算法 2 中,首先将非平凡保守末端填充为孤立保守末端,然后利用 Hannenhalli 方法进行翻转,故算法 2 中所需要的翻转数 $\beta(\pi_S)$ 为最小。其次,在把 $\pi \setminus \pi_S$ 重组为 $\gamma \setminus \pi_S$ 的过程中,把要删除的基因放在间接黑边上,故不影响翻转操作的次数。利用 Hannenhalli 方法把 $\pi \setminus \pi_S$ 转化为 $\gamma \setminus \pi_S$,其翻转数 $d(\pi \setminus \pi_S)$ 是最小的。由引理 3 知,所有的间接黑边通过翻转后连在一起,这样保证了使用尽可能少的删除操作,即 $d(A_\pi)$ 为最小的。故定理得证。

定理 3 完美重组染色体 π 可在多项式时间 $O(n^4)$ 内完成,其中 n 为染色体 π 中的基因个数。

证明 由定理 2 知, $d(\pi) = \beta(\pi_S) + d(\pi \setminus \pi_S) + d(A_\pi)$ 。根据 [5] 知, $\beta(\pi_S), d(\pi \setminus \pi_S)$ 都是在多项式时间 $O(n^4)$ 内完成。若设染色体 $\pi \setminus \pi_S$ 中定向分支个数为 $v(\pi \setminus \pi_S)$,非定向分支个数为 $\mu(\pi \setminus \pi_S)$,则由引理 3 和算法 1 知,最后进行的删除操作数 $d(\pi) = v(\pi \setminus \pi_S) + \mu(\pi \setminus \pi_S) - \alpha(Ih(\pi \setminus \pi_S))$ 。故定理得证。

参考文献:

- [1] BÉRARD S, BERGERON A, CHAUVE C. Perfect sorting by reversals is not always difficult[C]// Proceeding of WABI 2005, Lecture Notes in Computer Science(3692). Berlin: Springer, 2005: 228-237.
- [2] SAGOT M F, TANNIER E. Perfect sorting by reversals[C]// Proceedings of COCOON 2005, Lecture Note in Computer Science(3595). Berlin: Springer, 2005: 42-51.
- [3] FIGEAC M, VARRÉ J S. Sorting by reversals with common intervals[C]// Proceedings of WABI 2004, Lecture Notes in Computer Science(3240). Berlin: Springer, 2004: 26-37.
- [4] HANNENHALLI S, PEVZNER P. Transforming cabbage into turnip (polynomial algorithm for sorting signed permutations by reversals [C]// Proceedings of the 27th ACM Symposium on Theory of Computing. New York: ACM Press, 1995: 178-189.
- [5] BERGERON A, MIXTACKI J, STOYE J. The inversion distance problem[C]// Mathematics of Evolution and Phylogeny. Oxford: Oxford University Press, 2005.
- [6] MABROUK N EL. Genome rearrangement by reversals and insertions/deletions of contiguous segments[C]// Proc 11th Annual Symposium on Combinatorial Pattern Matching (CPM'00). London: Springer, 2000: 222-234.

(编辑:李晓红)