GIẢI THUẬT LAI CHO BÀI TOÁN SẮP HÀNG ĐA TRÌNH TỰ SINH HỌC

Nguyễn Ngọc Tú, Trần Văn Lăng
Phân viện Công nghệ thông tin tại Tp.HCM

1. GIỚI THIỆU

Từ những năm cuối thế kỷ 20, di truyền học và kỹ thuật gen đã phát triển nhanh chóng và đạt được nhiều thành tựu to lớn. Sự phát triển này giúp cho con người ngày càng hiểu rõ hơn cơ sở khoa học về sự sống. Và chính sự hiểu biết này đóng góp vai trò rất lớn đối với lĩnh vực chăm sóc và bảo vệ sức khoẻ con người. Chẳng hạn, việc chẩn đoán, dự phòng, trị liệu, v.v...Từ đó, nâng cao chất lượng cuộc sống và bảo vệ môi trường thiên nhiên. Đi kèm với sự phát triển của lĩnh vực sinh học, một vấn đề đặt ra là sự tham gia của các ngành khoa học khác, đặc biệt là ngành khoa học máy tính **[9]**. Ngành sinh học phân tử càng phát triển, càng đòi hỏi sự hỗ trợ rất lớn từ phía tin học, qua đó có thể giải quyết các bài toán lớn và phức tạp nhằm phục vụ cho những hiểu biết của con người về thế giới sinh vật, cũng như chính bản thân con người. Sự thành công của các dự án nghiên cứu về gen, cùng với sự hỗ trợ của các công cụ tin học, đã dẫn đến một sự thay đổi lớn trong việc nghiên cứu các vấn đề liên quan đến sinh học. Người ta chuyển dịch dần từ sự quan tâm cấu trúc của các đa phân tử sinh học sang sự phân tích các trình tự sinh học (*sequence analysis*) bằng các phương tiện tin học. Phương tiện tin học không phải chỉ dừng lại ở việc tạo ra các cơ sở dữ liệu lớn, mà còn tạo ra các công cụ hữu hiệu để phân tích và tìm hiểu bản chất của các các đa phân tử sinh học.

Chính vì vậy, trong quá trình nghiên cứu của các nhà sinh học, bước đầu tiên và cũng là bước quan trọng là quá trình phân tích trình tự. Để đảm bảo cho sự thành công và cho ra kết quả nhanh chóng, công cụ tin học đóng vai trò khá đắc lực. Với tốc độ gia tăng rất lớn về số lượng các trình tự sinh học được nghiên cứu nhằm chia sẻ thông tin chung trên toàn thế giới, dữ liệu về các trình tự sinh học có tại National Center for Biotechnology Information (NCBI), của Mỹ đã có tới hơn 120GB chứa khoảng 9 Gbase (*Gbase hay còn gọi là Giga base pairs, ở đó base pairs là một cặp bazờ gồm 2 nucleotid đối ngược nhau trong chuỗi xoắn kép* - National Center for Biotechnology Information, http://www.ncbi.nlm.nih.gov). Cùng với nó là sự phức tạp trong quá trình so sánh và tìm kiếm, dẫn tới những đòi hỏi ngày càng cao về các phương pháp, giải thuật tham gia. Đặc biệt, cần phải đảm bảo chất lượng của quá trình so sánh sao cho chấp nhận được, đồng thời thời gian đáp ứng cũng cần phải nhanh chóng.Trên thế giới đã có các nghiên cứu và có nhiều phương pháp được đưa ra, nhưng mỗi cách đều có những mặt mạnh và mặt yếu của nó khi giải quyết bài toán này **[9]**. Qua quá trình nghiên cứu, chúng tôi đề ra giải pháp kết hợp một số kỹ thuật để giải quyết bài toán.

Những kết quả được trình bày trong phần 4 về kết quả thử nghiệm, trong phần 2 được dùng để trình bày bài toán sắp hàng đa trình tự, phần 3 trình bày phương pháp lai do chúng tôi đề xuất để giải quyết bài toàn.

2. BÀI TOÁN SẮP HÀNG

Các dữ liệu thường được đem so sánh và phân tích bao gồm các chuỗi trình tự những nucleotide (*DNA*) và chuỗi trình tự những amino acid (*Protein*) **[1,4,9]**: DNA (*Deoxyribo Nucleic Acid*) và RNA (*Ribo Nucleic Acid*) là hai đại phân tử (đa phân tử) sinh học. Chúng là các nucleic acid vật chất mang thông tin di truyền từ các hệ thống sống. Ở đây, quá trình so sánh và tìm kiếm chỉ quan tâm nhiều tới DNA, nói đúng ra là một mạch đơn của chuỗi xoắn kép DNA. Mỗi mạch đơn DNA là một chuỗi các nucleotide sắp sếp kế tiếp nhau, nucleotide có 4 loại và được ký hiệu như sau: A (*Adenine*), G (*Guanine*), C (*Cytosine*), T (*Thymine*). Ta có bộ ký hiệu cho các nucleotide như sau: Nuc = {A, C, G, T}. Chẳng hạn, đoạn chuỗi trình tự Nucleotide của virus SARS có dạng: GATATTAGGTTTTTACCTACCCAGGAAAAGCCAACCAACC TCGATCTCTT. Protein là biểu biện của vật chất sống, nó tham gia vào hầu hết các quá trình

sinh học và là cơ sở của sự đa dạng về cấu trúc và chức năng của tất cả các sinh vật. Trong sự sống, protein được tạo ra qua quá trình dịch mã từ đoạn gen biểu hiện chứa thông tin di truyền trong DNA. Protein là một chuỗi trình tự các amino acid nối kết với nhau bằng các liên kết tạo nên cấu trúc (*được chia ra làm nhiều dạng cấu trúc như bậc 1, bậc 2 và cấu trúc không gian bậc 3, bậc 4, bậc 5*). Amino acid gồm có 20 loại được ký hiệu tắt bởi các chữ cái. Mỗi Amino acid được mã hoá từ bộ 3 nucleotide. Tuy có 64 bộ mã hoá nhưng chỉ có 20 loại amino acid và một số mã làm tín hiệu cho việc dịch mã từ DNA. Bộ ký hiệu cho các amino acid: AA = {A, C, D, E, F, G, H, I, K, L, M, N, P, Q, R, S, T, V, W, Y}. Trình tự của protein là một chuỗi trình tự các amino acid, chẳng hạn với virus SARS có một đoạn amino acid như sau: MSDNGPQSNQRSAPRITFGGPTDSTDNNQNGGRNGARPKQRRPQ.

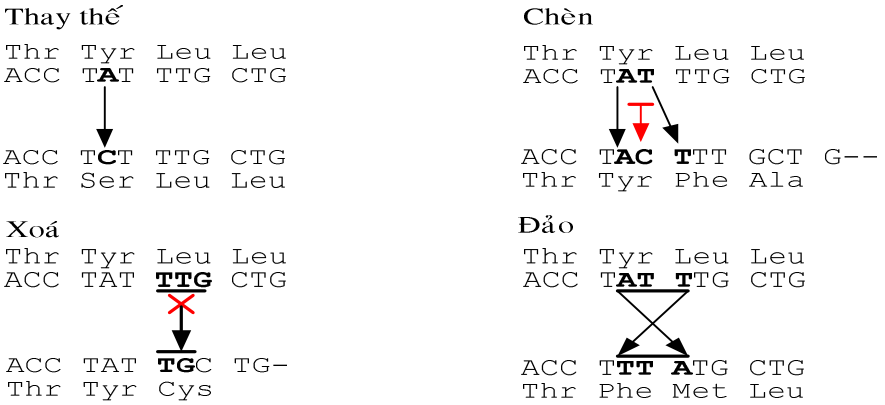
Khi đề cập tới việc xác định mức độ giống nhau của các trình tự có các vấn đề sau:

Để xác định giữa các trình tự hay các phần nào trong các trình tự giống nhau nhất và cũng là để xác định sự tương đồng, khi so sánh giữa các trình tự cần có hàm đánh giá mức độ giống nhau (*similarity measure*), hay còn gọi là đánh giá khoảng cách (*distance measure*) giữa các trình tự. Trong trường hợp đơn giản, hàm đánh giá này chỉ là đếm bao nhiêu phần tử giống nhau.

Trong quá trình tiến hoá, các trình tự có thể thêm hoặc bớt đi một số phần tử (*thường ký hiệu là InDel - insertions/deletions*) trong trình tự, cho nên các sinh vật có họ hàng gần nhau có thể các trình tự khác nhau ở phần thêm vào chen giữa trình tự. Bởi vậy khi chuyển sang việc so sánh trong mô hình toán học cần phải cho phép có quãng cách (*gap - được ký hiệu bằng dấu "-"*) để có thể tìm được các phần trình tự giống nhau nhất. Tuy nhiên, khả năng thêm hay bớt trong các trình tự là quá trình tiến hoá lâu dài vì vậy khi đánh giá các sinh vật nào gần nhau thì cũng có ít quãng cách hơn. Do đó trong mô hình toán học có đưa vào điểm phạt cho quãng cách (*gap penalties*) sao cho đáp ứng giống bài toán thực tế. Các loài gần sau sẽ có trình tự giống nhau ở các đoạn liên tục và dài cho nên các mô hình toán học còn thêm điểm phạt cho mỗi một đoạn quãng cách (*open gap penalties*).

Bên cạnh đó, trong quá trình tiến hoá cũng có trường hợp bị đột biến tại một số phần tử trong trình tự (*có thể hiểu đơn giản là nucleotide hay amino acid này được thay thế bằng phần tử khác*).

Trong hình 1 là bốn kiểu biến đổi chủ yếu trong quá trình tiến hoá các sinh vật trong tự nhiên:



Hình 1. Các trường hợp biến đổi chuỗi trình tự sinh học

Qua quá trình nghiên cứu lý thuyết và thực nghiệm, các nhà nghiên cứu đã đưa ra một số các bảng thống kê đánh giá mức độ đột biến từ phần tử này sang phần tử khác (*chẳng hạn như PAM - Point-Accepted-Mutation do Dayhoff đưa ra – hình 2, BOSUM - BLOcks Substitution Matrix do Henikoff và Henikoff đưa ra – hình 3, …*) để có thể phản ánh đúng ý nghĩa sinh học (xem **[6]**).

	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V	B	Z	X	*
A	2	-2	0	0	-2	0	0	1	-1	-1	-2	-1	-1	-3	1	1	1	-6	-3	0	0	0	0	-8
R	-2	6	0	-1	-4	1	-1	-3	2	-2	-3	3	0	-4	0	0	-1	2	-4	-2	-1	0	-1	-8
N	0	0	2	2	-4	1	1	0	2	-2	-3	1	-2	-3	0	1	0	-4	-2	-2	2	1	0	-8
D	0	-1	2	4	-5	2	3	1	1	-2	-4	0	-3	-6	-1	0	0	-7	-4	-2	3	3	-1	-8
C	-2	-4	-4	-5	12	-5	-5	-3	-3	-2	-6	-5	-5	-4	-3	0	-2	-8	0	-2	-4	-5	-3	-8
Q	0	1	1	2	-5	4	2	-1	3	-2	-2	1	-1	-5	0	-1	-1	-5	-4	-2	1	3	-1	-8
E	0	-1	1	3	-5	2	4	0	1	-2	-3	0	-2	-5	-1	0	0	-7	-4	-2	3	3	-1	-8
G	1	-3	0	1	-3	-1	0	5	-2	-3	-4	-2	-3	-5	0	1	0	-7	-5	-1	0	0	-1	-8
H	-1	2	2	1	-3	3	1	-2	6	-2	-2	0	-2	-2	0	-1	-1	-3	0	-2	1	2	-1	-8
I	-1	-2	-2	-2	-2	-2	-2	-3	-2	5	2	-2	2	1	-2	-1	0	-5	-1	4	-2	-2	-1	-8
L	-2	-3	-3	-4	-6	-2	-3	-4	-2	2	6	-3	4	2	-3	-3	-2	-2	-1	2	-3	-3	-1	-8
K	-1	3	1	0	-5	1	0	-2	0	-2	-3	5	0	-5	-1	0	0	-3	-4	-2	1	0	-1	-8
M	-1	0	-2	-3	-5	-1	-2	-3	-2	2	4	0	6	0	-2	-2	-1	-4	-2	2	-2	-2	-1	-8
F	-3	-4	-3	-6	-4	-5	-5	-5	-2	1	2	-5	0	9	-5	-3	-3	0	7	-1	-4	-5	-2	-8
P	1	0	0	-1	-3	0	-1	0	0	-2	-3	-1	-2	-5	6	1	0	-6	-5	-1	-1	0	-1	-8
S	1	0	1	0	0	-1	0	1	-1	-1	-3	0	-2	-3	1	2	1	-2	-3	-1	0	0	0	-8
T	1	-1	0	0	-2	-1	0	0	-1	0	-2	0	-1	-3	0	1	3	-5	-3	0	0	-1	0	-8
W	-6	2	-4	-7	-8	-5	-7	-7	-3	-5	-2	-3	-4	0	-6	-2	-5	17	0	-6	-5	-6	-4	-8
Y	-3	-4	-2	-4	0	-4	-4	-5	0	-1	-1	-4	-2	7	-5	-3	-3	0	10	-2	-3	-4	-2	-8
V	0	-2	-2	-2	-2	-2	-2	-1	-2	4	2	-2	2	-1	-1	-1	0	-6	-2	4	-2	-2	-1	-8
B	0	-1	2	3	-4	1	3	0	1	-2	-3	1	-2	-4	-1	0	0	-5	-3	-2	3	2	-1	-8
Z	0	0	1	3	-5	3	3	0	2	-2	-3	0	-2	-5	0	0	-1	-6	-4	-2	2	3	-1	-8
X	0	-1	0	-1	-3	-1	-1	-1	-1	-1	-1	-1	-1	-2	-1	0	0	-4	-2	-1	-1	-1	-1	-8
*	-8	-8	-8	-8	-8	-8	-8	-8	-8	-8	-8	-8	-8	-8	-8	-8	-8	-8	-8	-8	-8	-8	-8	1

Hình 2. Ma trận biến đổi amino acid PAM

Theo các nghiên cứu, các thay đổi dạng chèn và xoá bớt ký tự trong trình tự xuất hiện rất ít so với trường hợp do đột biến. Do đó, trong mô hình so sánh các trình tự không quan tâm tới việc chèn hay xoá thêm các ký tự mà chỉ xét thêm các quãng cách (*gaps*) trong việc so sánh để đảm bảo phản ánh chính xác của loại thay đổi này. Quãng cách được hiểu đơn giản khi nhìn trong trình tự là phần trống, không có ký tự để so sánh với ký tự của chuỗi khác. Chẳng hạn, với chuỗi sau 'AAG-AT-A' có hai quãng cách, mỗi quãng có một chỗ trống. Còn với chuỗi 'AA--GATA' có một quãng cách với hai khoảng trống. Khi tính điểm so sánh phải tính thêm điểm phạt (*gap panelty*) do quãng cách này gây ra vì càng nhiều quãng cách, khoảng trống thì các trình tự đem so sánh càng ít giống nhau.

	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V
A	**5**	−2	−1	−2	−1	−1	−1	0	−2	−1	−2	−1	−1	−3	−1	1	0	−3	−2	0
R	−2	**7**	−1	−2	−4	1	0	−3	0	−4	−3	3	−2	−3	−3	−1	−1	−3	−1	−3
N	−1	−1	**7**	2	−2	0	0	0	1	−3	−4	0	−2	−4	−2	1	0	−4	−2	−3
D	−2	−2	2	**8**	−4	0	2	−1	−1	−4	−4	−1	−4	−5	−1	0	−1	−5	−3	−4
C	−1	−4	−2	−4	**13**	−3	−3	−3	−3	−2	−2	−3	−2	−2	−4	−1	−1	−5	−3	−1
Q	−1	1	0	0	−3	**7**	2	−2	1	−3	−2	2	0	−4	−1	0	−1	−1	−1	−3
E	−1	0	0	2	−3	2	**6**	−3	0	−4	−3	1	−2	−3	−1	−1	−1	−3	−2	−3
G	0	−3	0	−1	−3	−2	−3	**8**	−2	−4	−4	−2	−3	−4	−2	0	−2	−3	−3	−4
H	−2	0	1	−1	−3	1	0	−2	**10**	−4	−3	0	−1	−1	−2	−1	−2	−3	2	−4
I	−1	−4	−3	−4	−2	−3	−4	−4	−4	**5**	2	−3	2	0	−3	−3	−1	−3	−1	4
L	−2	−3	−4	−4	−2	−2	−3	−4	−3	2	**5**	−3	3	1	−4	−3	−1	−2	−1	1
K	−1	3	0	−1	−3	2	1	−2	0	−3	−3	**6**	−2	−4	−1	0	−1	−3	−2	−3
M	−1	−2	−2	−4	−2	0	−2	−3	−1	2	3	−2	**7**	0	−3	−2	−1	−1	0	1
F	−3	−3	−4	−5	−2	−4	−3	−4	−1	0	1	−4	0	**8**	−4	−3	−2	1	4	−1
P	−1	−3	−2	−1	−4	−1	−1	−2	−2	−3	−4	−1	−3	−4	**10**	−1	−1	−4	−3	−3
S	1	−1	1	0	−1	0	−1	0	−1	−3	−3	0	−2	−3	−1	**5**	2	−4	−2	−2
T	0	−1	0	−1	−1	−1	−1	−2	−2	−1	−1	−1	−1	−2	−1	2	**5**	−3	−2	0
W	−3	−3	−4	−5	−5	−1	−3	−3	−3	−3	−2	−3	−1	1	−4	−4	−3	**15**	2	−3
Y	−2	−1	−2	−3	−3	−1	−2	−3	2	−1	−1	−2	0	4	−3	−2	−2	2	**8**	−1
V	0	−3	−3	−4	−1	−3	−3	−4	−4	4	1	−3	1	−1	−3	−2	0	−3	−1	**5**

Hình 3. Ma trận biến đổi BLOSUM

Có hai cách tính điểm phạt do quãng cách gây ra như sau:

Cách tính tuyến tính: $\gamma(g) = -gd$ (1)

Tính có sự ảnh hưởng khác nhau giữa khoảng trống đầu và khoảng trống mở rộng thêm : $\gamma(g) = -d - (g-1)e$. (2)

Trong đó g là số khoảng trống, d là điểm phạt cho một khoảng trống mở đầu, e là điểm phạt cho mỗi khoảng trống mở rộng thêm trong một quãng cách.

Trong quá trình tìm sự tương đồng, trường hợp nào thấy tương đồng nhất (*có điểm tính cao nhất*) sẽ được chọn. Thông thường có hai cách so sánh các trình tự **[7]**:

- So sánh tương đồng toàn cục: trường hợp này xét tương đồng trên toàn chuỗi, công thức tính cho việc so sánh như sau (*so sánh 2 chuỗi*):

$$F(i,\ j) = \max \begin{cases} F(i-1,\ j-1) + s(x_i, y_j) \\ F(i-1,\ j) - d \\ F(i,\ j-1) - d \end{cases} \quad (3)$$

- So sánh tương đồng cục bộ: tìm phần giống nhau nhất giữa hai chuỗi trình tự, công thức tính như sau:

$$F(i,\ j) = \max \begin{cases} 0 \\ F(i-1,\ j-1) + s(x_i, y_j) \\ F(i-1,\ j) - d \\ F(i,\ j-1) - d \end{cases} \quad (4)$$

Với $F(i, j)$ là điểm số tương đồng tích luỹ dần khi so sánh hai chuỗi trình tự tới vị trí i của chuỗi 1 và j của chuỗi 2. Và s là hàm tính toán sự tương đồng từng ký hiệu đơn của hai chuỗi dựa trên các bảng đánh giá như PAM, BLOSUM. Với cách tính trên, kết quả của vị trí so sánh cuối cùng $F(n_1,n_2)$ là số điểm tính sự tương đồng giữa các trình tự.

Khi so sánh nhiều trình tự ta có cách tính tổng số điểm tương đồng (*SP – Sum of Pairs*) là tổng điểm tương đồng của từng cặp như sau:

$$S = \sum_i S(m_i) = \sum_i \sum_{j<=k} s(m_i^j, m_i^k) \quad (5)$$

Trong đó $S(m_i)$ là điểm tương đồng tính tại một vị trí i của toàn bộ các trình tự, $s(m_i^j, m_i^k)$ là điểm tương đồng của cặp trình tự j và k tại vị trí i. Từ trên ta thấy ở mức độ đơn giản, chỉ so sánh giữa hai trình tự với nhau (*PSA – sắp hàng cặp trình tự*), ta ký hiệu hai chuỗi trình tự là S_1, S_2. Trong đó S_1 có độ dài là n_1 và S_2 có độ dài là n_2 thì phương pháp so trùng tìm sự tương đồng tối ưu có độ phức tạp là $O(n_1 n_2)$.

Với k chuỗi có độ dài n, khi áp dụng quy hoạch động thì độ phức tạp vẫn rất lớn: $O\left(\left(2^k - 1\right)n^k\right)$ **[10]**. Vấn đề càng phức tạp hơn khi có xu hướng so sánh toàn bộ hệ gen (*lên tới cả tỷ ký tự*) chứ không chỉ là một đoạn trình tự (*thường chỉ vài trăm đến vài ngàn ký tự*).

Sau đây là một vài ví dụ về việc so sánh các trình tự: Với hai hai trình tự: $S1$: ACGACA; $S2$: AGCAC, ta có một số kết quả so sánh như sau:

	(1)	(2)	(3)
S1’	ACGACA-	ACG-ACA	A-CGACA
S2’	A-G-CAC	A-GCAC-	AGC-AC-

Còn với ba trình tự: S1: ATTCGAC; S2: TTCCGTC; S3: ATCGTC

Ta có kết quả gióng cột:

S1’	ATT-CGA-C
S2’	-TTCCG-TC
S3’	A-T-CG-TC

3. GIẢI THUẬT LAI

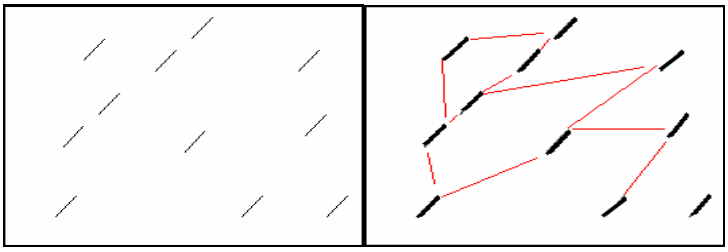
Với bài toán sắp hàng đa trình tự, giải pháp giải quyết bài toán dựa trên sự kết hợp giải thuật di truyền và kỹ thuật luyện kim – chúng tôi xem xét như thuật giải lai (GA-SA) - được thực hiện

qua hai giai đoạn sau: phân tích các trình tự tạo các thông tin hỗ trợ cho quá trình chính; thực hiện quá trình tìm kết quả tương đồng tốt nhất nhờ giải thuật di truyền và kỹ thuật luyện kim.

3.1 Tạo thông tin hỗ trợ

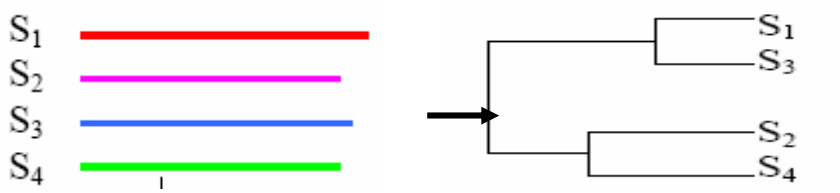
Ý tưởng chính của phần này là thực hiện đánh giá sơ bộ ban đầu các phần giống nhau giữa các trình tự để có thông tin về các vùng tương tự, từ đó giảm thiểu các trường hợp cần xét tới. Để thực hiện nhanh cho việc tìm kiếm, các bộ từ ban đầu (*word, k-tuple: là một chuỗi trình tự con, khoảng 5-10 ký tự cho DNA và 2-3 ký tự cho Protein*) sẽ được thiết lập như là bảng băm (*hash*) và duyệt trong các trình tự. Từ các thông tin này sẽ tìm ra các vùng giống nhau giữa các trình tự, cũng có nghĩa là điểm tương đồng giữa hai phần này trong chuỗi trình tự là tối ưu.

Với kết quả tìm được, các thông tin về các đoạn tương đồng sẽ được sắp xếp theo thứ tự và được đánh dấu ưu tiên trong việc bảo toàn các đoạn này trong việc đi tìm các giải pháp tốt hơn cho toàn bộ các trình tự (hình 4).



Hình 4. Các đoạn trình tự giống nhau và thứ tự sắp xếp

Tiếp theo, giải thuật thực hiện phân tích từng cặp trình tự để tạo cây phân loài giúp xác định các trình tự nào gần nhau nhất hướng dẫn cho giải thuật ưu tiên các phần giống nhau giữa một số trình tự, chỉ thay đổi các phần không giống nhau nhiều (hình 5) **[2]**.

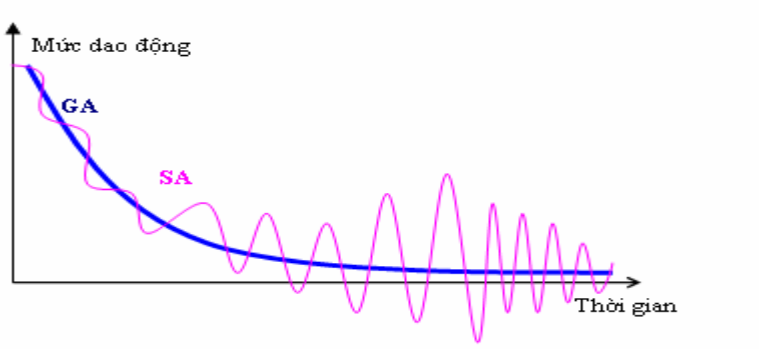


Hình 5. Các trình tự và quan hệ tiến hoá trong cây phân loài

3.2 Áp dụng GA-SA

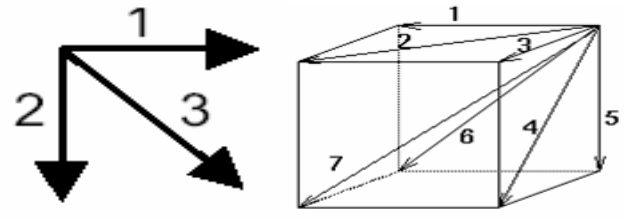
Kết hợp tạo giải thuật lai giữa hai giải thuật chính là giải thuật di truyền và kỹ thuật Simulated Annealing (SA). Trong đó tận dụng ý tưởng là tìm kiếm dựa trên quần thể như của giải thuật di truyền và biến đổi trạng thái như của SA trên các ”cá thể” nhưng quá trình phát sinh các cá thể không chỉ hoàn toàn là ngẫu nhiên mà còn dùng các phép toán lai tạo, đột biến, chọn lọc của giải thuật di truyền để có thể kế thừa được các giải pháp tốt. Bên cạnh đó nhờ có bước thực hiện ban đầu mà có thể tạo được quần thể ban đầu tương đối tốt và làm cơ sở cho việc xét các bước chuyển trạng thái mới.

Quá trình sẽ thực hiện một phần song song giữa GA (Genetic Algorithms) và SA, một số phần tử tốt nhất của GA chuyển sang cho thực hiện các biến đổi theo giải thuật SA. Trong quá trình thực hiện từ SA, một số trạng thái phát sinh có độ thích nghi tốt sẽ chuyển sang GA để thực hiện lai ghép với các các thể khác. Kết hợp hai giải thuật còn nhằm mục đích ”phá vỡ” một phần sự cứng nhắc và ít biến đổi khi giải thuật GA thực hiện giai đoạn cuối, SA được thiết lập giúp tăng khả năng chọn lựa vào các vùng không gian nghiệm rộng hơn. Hình sau mô tả mức dao động của không gian vùng nghiệm khi kết hợp GA-SA:



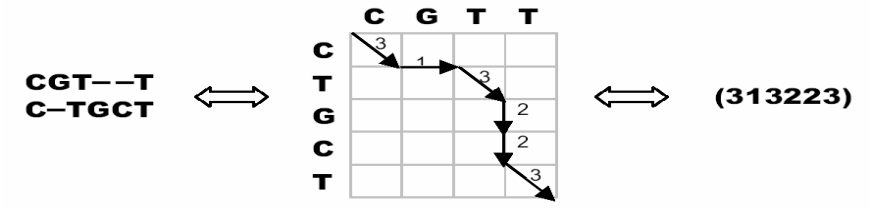
Hình 7. Biểu đồ mô tả mức độ ổn định của nghiệm được chọn

Xuất phát từ quá trình tìm kiếm giải pháp tốt nhất, mỗi bước xét từ một vị trí so trùng đi tới bước kế tiếp thì có 3 chọn lựa khi so sánh 2 trình tự. Tương tự ta có $(2^k - 1)$ chọn lựa cho việc xét cách so trùng kế tiếp, bởi với mỗi ký tự của một chuỗi trình tự ta xét có 2 trường hợp xảy ra, giữ nguyên ký tự hay chèn thêm khoảng trống vào chuỗi. Chúng tôi đánh mã tương ứng cho mỗi chọn lựa là $1..(2^k - 1)$ như hình 6.



Hình 7. Hướng chọn lựa từ một điểm so trùng khi so sánh 2,3 trình tự

Với bài toán này giải thuật GA được áp dụng với các cá thể có nhiễm sắc thể với độ dài khác nhau. Và mỗi phần tử trong nhiễm sắc thể đại diện cho một hướng chọn lựa đi từ vị trí trước nó. Chẳng hạn, với cặp trình tự so sánh sau ta có hình minh hoạ hướng và nhiễm sắc thể tương ứng **[5,6,8]**:



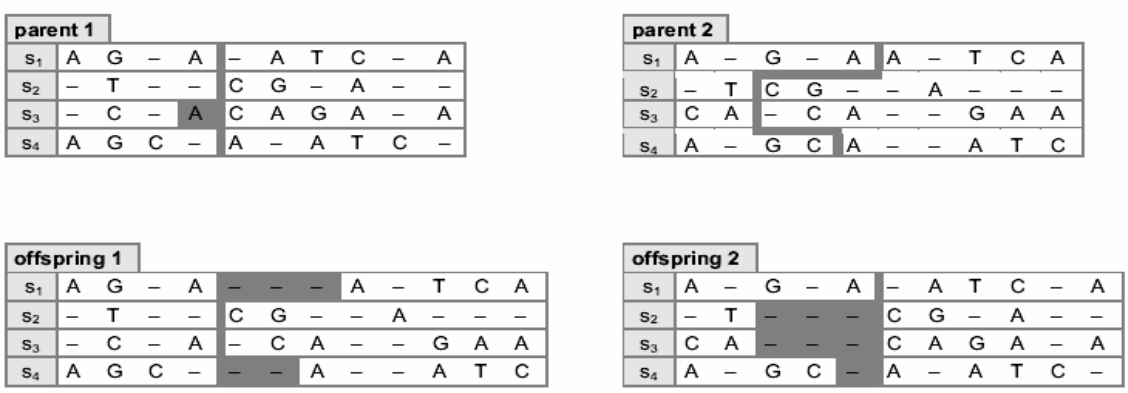
Hình 8. Mối quan hệ giữa gióng cột đa trình tự và cách biểu diễn một cá thể

Để biểu diễn các phép toán sử dụng trong GA được đơn giản, chúng tôi mô tả tương ứng với các trường hợp là phần sắp hàng của nó. Hàm thích nghi của giải thuật di truyền:

$$Fitness = SymbolScore - GapScore \quad (6)$$

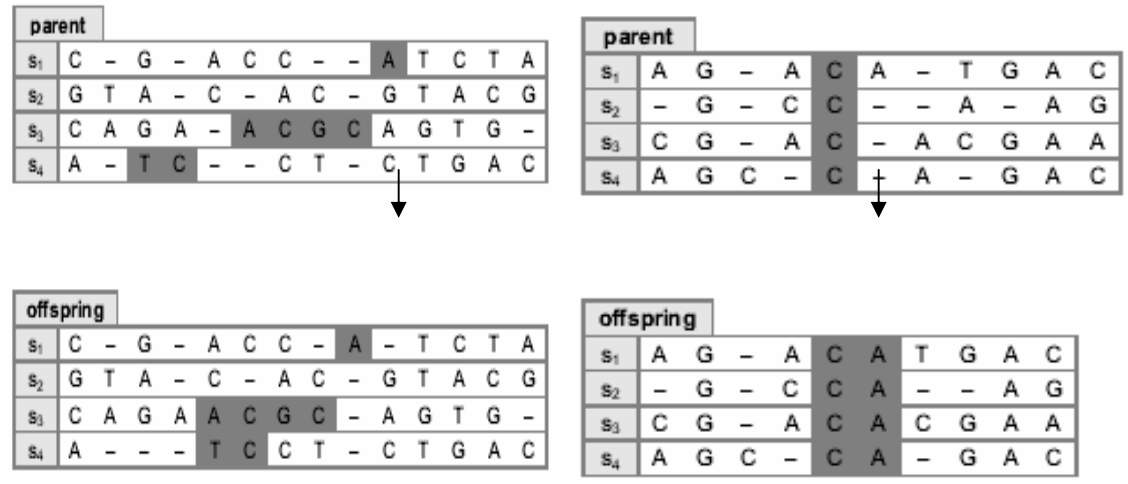
Với SymbolScore là điểm tương đồng giữa các ký hiệu tại tất cả các vị trí, còn Gapscore là điểm khác biệt tạo thành các khoảng trống tại tất cả các vị trí xuất hiện.

Phép toán lai ghép một điểm: chọn một điểm ngẫu nhiên trên các vùng ít tương đồng của cá thể cha thứ 1, tìm các vị trí tương ứng trên cá thể cha thứ 2 và ghép các phần với nhau tạo nên hai cá thể con (hình 8):



Hình 9. Lai ghép hai cá thể cha

Các phép đột biến như chèn thêm các khoảng trống vào, chuyển đổi vùng trống từ vị trí này sang vị trí khác trong một chuỗi, hoặc dịch hẳn một khối nhỏ nếu như vùng này thuộc các vùng đã xác định là giống nhau giữa một số trình tự. Với mỗi phép đột biến này, một số ký tự sẽ được chuyển sao cho cùng vị trí với các ký tự giống nó trên các chuỗi khác (hình 9).



Hình 10. Một phép đột biến trong các cá thể

Cùng thực hiện đồng thời và tác động tới quần thể của giải thuật GA là việc áp dụng ”kỹ thuật luyện kim” trên một số cá thể ”thích nghi” được của GA. Giải thuật của SA được áp dụng như hình 10 **[3,7]**:

Chọn lời giải ban đầu s0;
Chọn nhiệt độ khởi đầu t0 > 0;
Chọn hàm thu giảm nhiệt độ α;
Repeat
  Repeat
    Chọn ngẫu nhiên một lời giải lân cận s của lời giải hiện tại s0;
    $\delta = f(s) - f(s0)$; */* sự thay đổi sự tương đồng, mức thích nghi */*
    if ($\delta > 0$) **then** *// lời giải sau tốt hơn lời giải trước*
      s0 = s;
    else
      sinh ngẫu nhiên một số $x \in [0,1]$;

```
                    if ( x < e^(δ/t) ) then
                        s0 = s;
                    endif
                endif
            unltil số-lần-lặp = nrep;
            t = α(t);
        until Điều-kiện-dừng = TRUE;
```

Hình 11. Giải thuật áp dụng mô phỏng luyện kim

Các trạng thái lân cận được xem xét như là các cá thể mới sinh ra trong quá trình đột biến của một cá thể của giải thuật GA.

Hai hàm thu giảm nhiệt độ được sử dụng đến trong hiện thực:

$$\alpha(t) = t / k; \qquad \alpha(t) = t \times \beta^k, \beta \in [0,1] \tag{7}$$

4. MỘT SỐ KẾT QUẢ THỰC NGHIỆM

Sau đây là một số kết quả thực thi giải thuật trên một số tập mẫu:

Bảng 1. Tính toán sắp hàng nhiều trình tự trên các mẫu thử nghiệm khác nhau

Tên mẫu	Số trình tự	Độ dài các trình tự		Điểm đánh giá (SPS)	Thời gian thực thi (ms)
		Min	Max		
aab_ref1	4	67	79	0.762	940
hpi_ref1	4	70	81	0.677	940
idy	4	70	81	0.766	4,530
dox_ref1	4	91	94	0.928	1,100
fmb_ref1	4	94	104	0.873	940
ar5A_ref1	4	192	203	0.873	780
ad2_ref1	4	203	213	0.830	1,250
aym3_ref1	4	219	244	0.803	1,090
gdoA_ref1	4	234	265	0.641	1,250
csp_ref1	5	66	70	0.932	1,250
csy_ref1	5	100	104	0.767	1,090
fkj_ref1	5	98	110	0.912	1,090
hfh_ref1	5	116	132	0.710	1,090
amk_ref1	5	242	250	0.931	1,410
ped_ref3	21	324	388	0.587	26,250
lvl_ref2	23	426	470	0.739	47,500
acr_ref7	43	1000	1136	0.703	855,940

Bảng 2. So sánh thời gian thực thi thuật toán tuần tự và song song

Số trình tự	Kích cỡ trung bình của trình tự	Tuần tự (ms)	P = 2		P = 3	
			T (ms)	Speedup	T (ms)	Speedup
4	100	940	587	1,6	469	2
4	200	1250	787	1,58	634	1,97

8	200	2810	1780	1,57	1447	1,95
16	200	7030	4352	1,6	3476	2
20	300	26250	16020	1,6	12653	2
128	300	1684370	1012983	1,6	789576	2,1

Với kết quả tính toán trong 2 bảng trên, ta nhận thấy : kết quả rất khả quan chính xác từ 70% - 93% so với dữ liệu thực nghiệm của sinh học trong ngân hành dữ liệu NCBI như đã trên. Trong khi đó, với Clustal, độ chính xác trung bình là 85%.

Thuật toán song song tương đối có hiệu quả, đặc biệt với số tiến trình P = 2, thuật toán có Speedup so với tuần tự đạt được 1,6.

5. ĐÁNH GIÁ VÀ KẾT LUẬN

Qua nghiên cứu các các giải thuật trước, chúng tôi chọn hướng giải quyết kết hợp giải thuật di truyền và kỹ thuật luyện kim nhưng có sự hướng dẫn của một số thông tin phân tích ban đầu để giúp giải quyết bài toán trong không gian giới hạn và có các bước chuyển tốt hơn. Tuy nhiên, dù có những kết quả thực nghiệm khá tốt nhưng qua đó cho thấy cần phải kết hợp với những ý nghĩa sinh học về sự tương đồng giữa các trình tự sinh học sâu rộng hơn mới có được những heuristic và phương pháp so sánh có kết quả giống như thực tế xảy ra trong sinh học. Bên cạnh đó, chúng tôi chuyển sang thực thi song song giải thuật để giảm tối đa thời gian xử lý và tăng khả năng tìm kiếm nghiệm cho bài toán này.

TÀI LIỆU THAM KHẢO

[1]. Hồ Huỳnh Thuỳ Dương, *Sinh học phân tử: khái niệm, phương pháp, ứng dụng*. Nxb Giáo Dục, (2002).

[2]. Hogeweg P, Hesper, The alignment of sets of sequences and the construction of phylogenetic trees. An integrated method. J. Mol. E. vol. 20, p175-186 (1984).

[3]. Lâm Kim Hoà, *Xếp lịch thi học kỳ bằng cách kết hợp lập trình ràng buộc và giải thuật mô phỏng luyện kim*, Luận văn Thạc sĩ, Đại Học Bách Khoa TPHCM, (2003).

[4]. Lê Đức Trình, *Sinh học phân tử của tế bào*. Nxb. Khoa Học Kỹ Thuật, (2001).

[5]. Notredame C, Higgins D. G., SAGA: sequence alignment by genetic algorithm. *Nucleic Acids Res.*, Vol. 24, p1515 - 1524 (1996).

[6]. R. Durbin, S. R. Eddy, A. Krogh, G. Mitchison. *Biological Sequence analysis: probabilistic models of proteins and nucleic acids*. Cambridge University Press, (2001)

[7]. S. Shen, J. Yang, A. Yao, P. Hwang, Super Pairwise Alignment, *J. Comp. Biol.* Vol. 9, p477 - 486 (2002)

[8]. Stefan Leopold, *An Alignment Graph based Evolutionary Algorithm for the Multiple Sequence Alignment Problem*, TUWIEN – Vienna University of Technology, (2001)

[9]. Yong Yang. *Comparative Analysis of Methods for Multiple Sequence Alignment*. Stanford University, (2001)

[10]. Wang L, Jiang T., On the complexity of multiple sequence alignment, *J. Comput. Biol.* T. 1, Vol. 4, p337 – 348 (1994).