

Trong bài tập này, chúng ta sẽ chạy thí nghiệm tìm kích thước quần thể nhỏ nhất cần thiết (minimally required population size - MRPS) để thuật toán di truyền đơn giản (simple Genetic Algorithm – sGA) với bản cài đặt POPOP cho 2 vấn đề OneMax và Trap k=5.

Các bạn cài đặt sGA theo bản cài đặt POPOP với tournament selection $s = 4$, xác suất lai ghép 100%, và không sử dụng đột biến.

Kích thước vấn đề (problem size) l là số biến của vấn đề. **Ví dụ:**

- OneMax $l = 40$ là vấn đề có 40 biến nhị phân.
- Trap k=5 với $l = 20$ là vấn đề có 20 biến nhị phân, được tạo nên bởi kết nối 4 hàm Trap k=5 lại với nhau.

Pseudo-code của phương pháp bisection tìm MRPS cho mỗi kích thước vấn đề l với 2 giai đoạn như sau:

Giai đoạn 1: Tìm cận trên (upper bound) của MRPS

$N^{upper} \leftarrow 4$

do

$N^{upper} \leftarrow 2 \times N^{upper}$

success \leftarrow Có thể chạy thành công sGA 10 lần (mỗi lần với 1 random seed khác nhau) với kích thước quần thể N^{upper} hay không?

while success = false

Giai đoạn 2: Tìm giá trị của MRPS

$N^{lower} \leftarrow N^{upper} / 2$

while $(N^{upper} - N^{lower}) / N^{upper} > 0.1$ do

$N = (N^{upper} + N^{lower}) / 2$

success \leftarrow Có thể chạy thành công sGA 10 lần (mỗi lần với 1 random seed khác nhau) với kích thước quần thể N hay không?

if success = true

$N^{upper} \leftarrow N$

else

$N^{lower} \leftarrow N$

if $(N^{upper} - N^{lower}) \leq 2$

break

Trả về $MRPS \leftarrow N^{upper}$ và average_number_of_evaluations (trung bình số lần gọi hàm fitness evaluations của quần thể có kích thước N^{upper} này, tính trung bình trên 10 lần chạy sGA).

Lưu ý 1: Dùng 10 random seed ở giai đoạn 2 là giống với 10 random seed ở giai đoạn 1.

Lưu ý 2: Thực hiện xong giai đoạn 1 và giai đoạn 2 như trên là **1 lần chạy bisection**. Mỗi lần chạy trả về một giá trị của MRPS. Chúng ta cần chạy **10 lần bisection** như vậy để lấy dữ liệu thống kê. Quy tắc sử dụng random seed như sau:

- Lần thứ 1: random seed từ MSSV+0 MSSV+9
- Lần thứ 2: random seed từ MSSV+10 MSSV+19
- Lần thứ 3: random seed từ MSSV+20 MSSV+29
-
- Lần thứ 10: random seed từ MSSV+90 MSSV+99

Đồ thị kết quả phải thể hiện được MRPS khi kích thước vấn đề tăng dần.

1. Trục hoành là kích thước vấn đề ($l = 10, 20, 40, 80, 160$).
2. Trục tung là MRPS trung bình (mean) của 10 lần chạy bisection **thành công** tương ứng với mỗi kích thước vấn đề.
3. Mỗi điểm kết quả cần thể hiện thanh lỗi (error bar) là ± 1 độ lệch chuẩn (standard deviation).
4. Sử dụng log scale để biểu diễn trục hoành và trục tung.

Đồ thị kết quả phải thể hiện được số lần gọi hàm fitness evaluations khi kích thước vấn đề tăng dần.

1. Trục hoành là kích thước vấn đề ($l = 10, 20, 40, 80, 160$).
2. Trục tung là số lần gọi hàm đánh giá trung bình (number_of_evaluations) của 10 lần chạy bisection **thành công** tương ứng với mỗi kích thước vấn đề. (**Note:** lấy mean của 10 giá trị average_number_of_evaluations của 10 lần bisection).
3. Mỗi điểm kết quả cần thể hiện thanh lỗi (error bar) là ± 1 độ lệch chuẩn (standard deviation).
4. Sử dụng log scale để biểu diễn trục hoành và trục tung.

Đồ thị 1 trong báo cáo cần chứa 2 đường kết quả như sau:

- Kết quả MRPS của sGA với phép lai một điểm (1X) cho OneMax.
- Kết quả MRPS của sGA với phép lai đồng nhất (UX) cho OneMax.

Đồ thị 2 trong báo cáo cần chứa 2 đường kết quả như sau:

- Kết quả số lần gọi hàm đánh giá (number_of_evaluations) của sGA với phép lai một điểm (1X) cho OneMax.
- Kết quả số lần gọi hàm đánh giá (number_of_evaluations) của sGA với phép lai đồng nhất (UX) cho OneMax.

Bảng kết quả 1 chứa các thông tin cho OneMax (từ bảng này có thể vẽ được đồ thị 1 và 2)

Problem size	sGA-1X		sGA-UX	
	MRPS	# Evaluations	MRPS	# Evaluations
10	Mean (std)	Mean (std)	Mean (std)	Mean (std)
20	Mean (std)	Mean (std)	Mean (std)	Mean (std)
40	Mean (std)	Mean (std)	Mean (std)	Mean (std)
80	Mean (std)	Mean (std)	Mean (std)	Mean (std)
160	Mean (std)	Mean (std)	Mean (std)	Mean (std)

Đồ thị 3 trong báo cáo cần chứa 2 đường kết quả như sau:

- Kết quả MRPS của sGA với phép lai một điểm (1X) cho Trap k=5.
- Kết quả MRPS của sGA với phép lai đồng nhất (UX) cho Trap k=5.

Đồ thị 4 trong báo cáo cần chứa 2 đường kết quả như sau:

- Kết quả số lần gọi hàm đánh giá (number_of_evaluations) của sGA với phép lai một điểm (1X) cho Trap k=5.
- Kết quả số lần gọi hàm đánh giá (number_of_evaluations) của sGA với phép lai đồng nhất (UX) cho Trap k=5.

Bảng kết quả 2 chứa các thông tin cho Trap k=5 (từ bảng này có thể vẽ được đồ thị 3 và 4)

Problem size	sGA-1X		sGA-UX	
	MRPS	# Evaluations	MRPS	# Evaluations
10	Mean (std)	Mean (std)	Mean (std)	Mean (std)
20	Mean (std)	Mean (std)	Mean (std)	Mean (std)
40	Mean (std)	Mean (std)	Mean (std)	Mean (std)
80	Mean (std)	Mean (std)	Mean (std)	Mean (std)
160	Mean (std)	Mean (std)	Mean (std)	Mean (std)

Lưu ý 3:

Khi giải các vấn đề có problem size lớn, có thể kích thước quần thể N^{upper} sẽ tăng rất lớn. Trong báo cáo này, nếu N^{upper} tăng đến 8192 mà vẫn không thể chạy thành công sGA 10 lần thì chúng ta KHÔNG chạy tiếp trường hợp đó nữa (không cần tiếp tục tăng kích thước quần thể trong Giai đoạn 1 lên nữa). Trong trường hợp này thì ta KHÔNG vẽ điểm dữ liệu tương ứng vào trong Đồ thị kết quả và KHÔNG có ô kết quả tương ứng trong Bảng kết quả.

Lưu ý 4:

Đối với bảng kết quả 1 và bảng kết quả 2, trong mỗi hàng cần chạy t-test để kiểm định ý nghĩa thống kê (so sánh p-value với significance level $\alpha=0.05$) sự khác biệt giữa 1X và UX. Nếu kết quả nào có ý nghĩa thống kê thì **tô đậm** kết quả tốt lên (ví dụ như trong 2 bảng bên trên). (Hint: có thể xem thử hàm `scipy.stats.ttest_ind`).

Nộp file zip duy nhất chứa các file khác như sau:

1. Source codes (C/C++/Python).
2. Files dữ liệu ghi dữ liệu kết quả (10 giá trị MRPS và 10 giá trị average number_of_evaluations tìm được trong 10 lần bisection thành công cho mỗi kích thước vấn đề, và 10 random seeds tương ứng của mỗi lần chạy bisection). Đây là thông tin cần thiết để kiểm tra kết quả thí nghiệm mà các bạn đã thực hiện (nếu cần).
3. Trong file báo cáo các bạn vẽ 4 đồ thị và 2 bảng kết quả như yêu cầu ở trên và ghi lại nhận xét/ý kiến của bản thân về hiệu suất của sGA khi giải OneMax và Trap với những phép lai ghép khác nhau.