

POPOP Problem

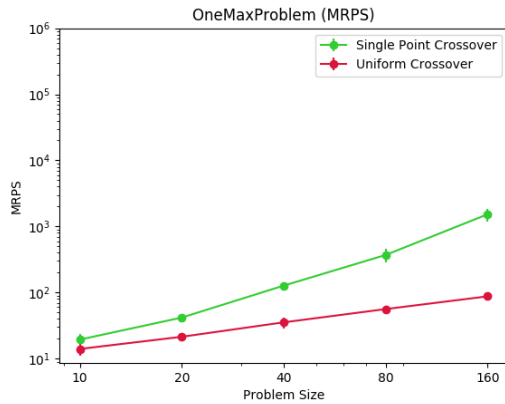
Khang Tran Dinh - MSSV: 18520072

October 2020

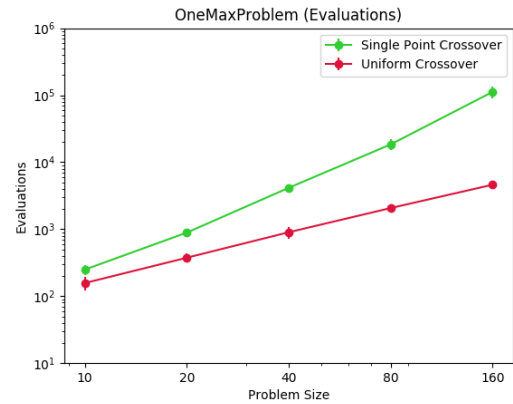
Bài toán OneMax là bài toán nhằm mục tiêu tối đa hóa số lượng số 1 trong quần thể.

1 OneMax Normal Problem

OneMax Normal Problem				
Problem Size	sGA-1X		sGA-UX	
	MRPS	Evaluations	MRPS	Evaluations
10	19.4 ± 4.1	249.2 ± 42.76	14.0 ± 2.97	156.8 ± 34.95
20	41.8 ± 6.23	888.0 ± 118.09	21.4 ± 3.1	374.8 ± 59.25
40	127.2 ± 16.35	4142.9 ± 417.31	35.4 ± 6.81	899.9 ± 170.86
80	370.6 ± 81.99	18437.5 ± 3452.81	56.2 ± 7.45	2066.5 ± 248.49
160	1529.8 ± 333.86	112219.9 ± 21341.02	87.6 ± 8.71	4611.8 ± 421.48



(a) OneMax Normal Problem with MRPS



(b) OneMax Normal Problem with Evaluations

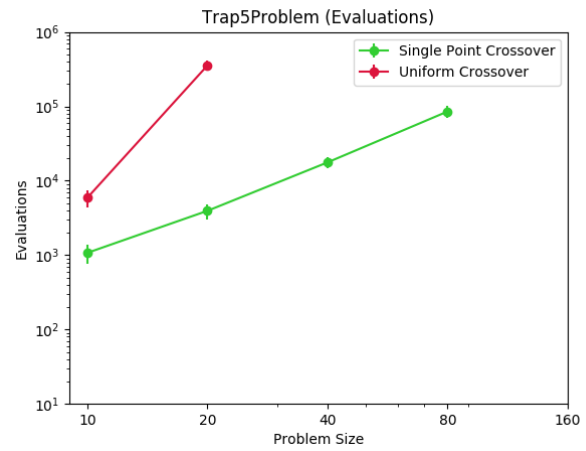
Hình 1: One Max Problem

2 OneMax Trap5 Problem

OneMax Trap5 Problem				
	sGA-1X		sGA-UX	
Problem Size	MRPS	Evaluations	MRPS	Evaluations
10	68.0 ± 20.12	1073.6 ± 319.96	216.0 ± 58.59	5914.7 ± 1566.82
20	168.2 ± 38.7	3941.4 ± 906.65	4502.8 ± 743.9	354791.3 ± 56929.76
40	499.4 ± 90.48	17647.7 ± 3004.76		
80	1611.6 ± 319.58	85309.4 ± 15879.51		
160				



(a) OneMax Trap5 Problem with MRPS



(b) OneMax Trap5 Problem with Evaluations

Hình 2: OneMax Trap5 Problem

3 Nhận xét

Việc sử dụng cả hai lai ghép *UX* và *1X* đều giải quyết hết tất cả 5 Problem Size trong *OneMax Normal Problem*, nhưng đối với vấn đề *OneMax Trap5 Problem* thì không thể giải trọn vẹn hết 5 Problem Size.

Đối với vấn đề *OneMax Normal Problem* việc sử dụng lai ghép *UX* tìm ra kết quả Evaluations và MRPS tốt hơn so với việc sử dụng lai ghép *1X*. Vì các đối với hàm *One Max Normal* thì việc thay đổi các vị trí độc lập có thể cho ra cùng một kết quả nên phép *UX* có ưu thế hơn.

Đối với vấn đề *OneMax Trap5 Problem*, việc sử dụng lai ghép *1X* tìm ra kết quả Evaluations và MRPS tốt hơn so với việc sử dụng lai ghép *UX*. Vì *OneMax Trap5 Problem* có cấu trúc từng các block liên kết với nhau, khi lai ghép *UX* xáo trộn nhiều một cách ngẫu nhiên sẽ phá vỡ cấu trúc các block nên việc tiến hóa rất tệ. Còn việc

sử dụng $1X$ đơn giản là vì khi chọn 1 điểm bất kì để cắt đôi chuỗi rồi hoán đổi, đơn giản là nó may mắn, lúc nó xáo trộn, khả năng chỉ phá nhiều nhất 1 block. Mặc dù vậy nhưng UX vẫn không thể giải hết được 5 Problem size.

Từ hai phân tích trên nhận thấy, đối với mỗi bài toán và hàm fitness khác nhau, cần khai thác được sự tương quan giữa các vị trí trong một cá thể. Từ đó có thể đưa ra phép lai ghép hợp lí mà có thể giảm đi sự phá vỡ cấu trúc có sẵn của bài toán.