

1 Giới thiệu bài toán Knapsack

Bài toán knapsack là một bài toán tối ưu tổ hợp, input là một tập hợp các túi chứa vật (có 2 chỉ số weight và value). Output của bài toán là số túi sao cho cân nặng nhỏ hơn hoặc bằng cân nặng cho phép và giá trị là lớn nhất. Bài toán cái túi đã được nghiên cứu trong hơn một thế kỷ, với các công trình đầu tiên có niên đại từ năm 1897. Tên gọi "bài toán knapsack" có từ những công trình đầu tiên của nhà toán học Tobias Dantzig (1884–1956), và đề cập đến vấn đề phổ biến là đóng gói những vật phẩm hữu ích hoặc có giá trị nhất mà không làm quá tải hành lý.

2 Thuật toán tiến hóa để giải bài toán knapsack

Với bài toán knapsack, ta sẽ khởi tạo các tham số ban đầu của bài toán như sau

- Initial Population: số cá thể ban đầu của bài toán
- Fitness Function: hàm số được chọn để ước lượng giá trị của cá thể
- Selection: lựa chọn cá thể để đột biến hoặc lai
- Crossover: tỉ lệ lai (thường rất cao để có thể đảm bảo có cá thể đời sau có chất lượng tốt sinh ra
- Mutation: tỉ lệ đột biến (thấp để đảm bảo tìm được ít bị ảnh hưởng)

Với bài toán Knapsack có số lượng đồ vật là n:

- Đầu tiên, ta sẽ khởi tạo kích thước quần thể (là số lời giải)
- Lựa chọn hàm số phù hợp cho bài toán Knapsack sao cho giá trị là lớn nhất với trọng lượng ở mức cho phép
- Lựa chọn các cá thể tốt nhất để tiến hành crossover
- lựa chọn crossover dựa trên one-point crossover
- với mutation ta sẽ sử dụng kĩ thuật lật bit (bit đã được đột biến là 1, ngược lại là 0)

Vì vậy các tham số bài toán được lựa chọn như sau

POPULATION SIZE \in (50,100,200,300)
CROSSOVER = 0.9
MUTATION = 0.1
MAX GENERATIONS = 50

Bảng viện phân tích thực nghiệm với thời gian giới hạn là 2 phút, ta có kết quả của các thuật toán tiến hóa như sau (chọn 3 loại đại diện group 0,6,12) và thay đổi POPULATION SIZE (việc thay đổi tỉ lệ đột biến cao hơn sẽ làm cho lời giải mang tính ngẫu nhiên hơn (việc có kết quả tốt hay xấu phụ thuộc vào việc phân tử đột biến theo chiều hướng xấu hay tốt (ta không thể khẳng định điều này)). Vì vậy ta chỉ cần giữ mức đột biến đủ nhỏ để chất lượng lời giải được đảm bảo sẽ tốt hơn.

1. Với POPULATION SIZE=50:

group	n	weight	value	item
00Uncorrelated	50	130491	179702	30
00Uncorrelated	100	239987	336484	61
00Uncorrelated	200	506172	719817	116
00Uncorrelated	500	1263000	1643058	276
00Uncorrelated	1000	2495109	2969864	538
00Uncorrelated	2000	4938073	5597142	1052

Result table for Uncorrelated group

06UncorrelatedWithSimilarWeights	50	2401258	17023	24
06UncorrelatedWithSimilarWeights	100	4902458	33541	49
06UncorrelatedWithSimilarWeights	200	9905108	64655	99
06UncorrelatedWithSimilarWeights	500	24712240	149788	247
06UncorrelatedWithSimilarWeights	1000	49525528	279611	495
06UncorrelatedWithSimilarWeights	2000	99048153	542488	990

Result table for UncorrelatedWithSimilarWeights group

12Circle	50	129609	8640156	27
12Circle	100	254769	16983726	48
12Circle	200	513048	34201438	103
12Circle	500	1230885	82054781	240
12Circle	1000	2443749	1.63E+08	479
12Circle	2000	4770600	3.18E+08	974

Result table for Circle group

2. Với POPULATION SIZE=100:

group	n	weight	value	item
00Uncorrelated	50	128449	178965	31
00Uncorrelated	100	240384	353461	60
00Uncorrelated	200	506371	715423	114
00Uncorrelated	500	1263086	1742890	286
00Uncorrelated	1000	2494684	3081184	554
00Uncorrelated	2000	4937715	5875698	1072

Result table for Uncorrelated group

06UncorrelatedWithSimilarWeights	50	2401236	17301	24
06UncorrelatedWithSimilarWeights	100	4902421	35413	49
06UncorrelatedWithSimilarWeights	200	9905186	69654	99
06UncorrelatedWithSimilarWeights	500	24711822	155074	247
06UncorrelatedWithSimilarWeights	1000	49524667	296781	495
06UncorrelatedWithSimilarWeights	2000	99049653	552641	990

Result table for UncorrelatedWithSimilarWeights group

12Circle	50	129609	8640159	25
12Circle	100	254769	16983726	47
12Circle	200	513049	34201506	98
12Circle	500	1230885	82054784	227
12Circle	1000	2443749	1.63E+08	476
12Circle	2000	4770600	3.18E+08	940

Result table for Circle group

3. Với POPULATION SIZE=200:

group	n	weight	value	item
00Uncorrelated	50	130854	179770	29
00Uncorrelated	100	240412	358988	61
00Uncorrelated	200	505159	782411	120
00Uncorrelated	500	1266225	1832476	283
00Uncorrelated	1000	2496513	3256394	562
00Uncorrelated	2000	4933654	6048001	1076

Result table for Uncorrelated group

06UncorrelatedWithSimilarWeights	50	2401378	17718	24
06UncorrelatedWithSimilarWeights	100	4902379	35672	49
06UncorrelatedWithSimilarWeights	200	9904946	71395	99
06UncorrelatedWithSimilarWeights	500	24712284	164205	247
06UncorrelatedWithSimilarWeights	1000	49524319	307833	495
06UncorrelatedWithSimilarWeights	2000	99049457	582313	990

Result table for Uncorrelated group

12Circle	50	129609	8640160	21
12Circle	100	254769	16983727	46
12Circle	200	513049	34201510	98
12Circle	500	1230885	82054787	226
12Circle	1000	2443749	1.63E+08	460
12Circle	2000	4770600	3.18E+08	936

Result table for Uncorrelated group

4. Với POPULATION SIZE=300:

group	n	weight	value	item
00Uncorrelated	50	130802	180893	31
00Uncorrelated	100	240589	360024	59
00Uncorrelated	200	506122	788146	118
00Uncorrelated	500	1265649	1884684	291
00Uncorrelated	1000	2495265	3302226	572
00Uncorrelated	2000	4937338	6267071	1112

Result table for Uncorrelated group

06UncorrelatedWithSimilarWeights	50	2401329	17809	24
06UncorrelatedWithSimilarWeights	100	4902393	35897	49
06UncorrelatedWithSimilarWeights	200	9905000	71470	99
06UncorrelatedWithSimilarWeights	500	24712081	165085	247
06UncorrelatedWithSimilarWeights	1000	49524451	314679	495
06UncorrelatedWithSimilarWeights	2000	99048162	576729	990

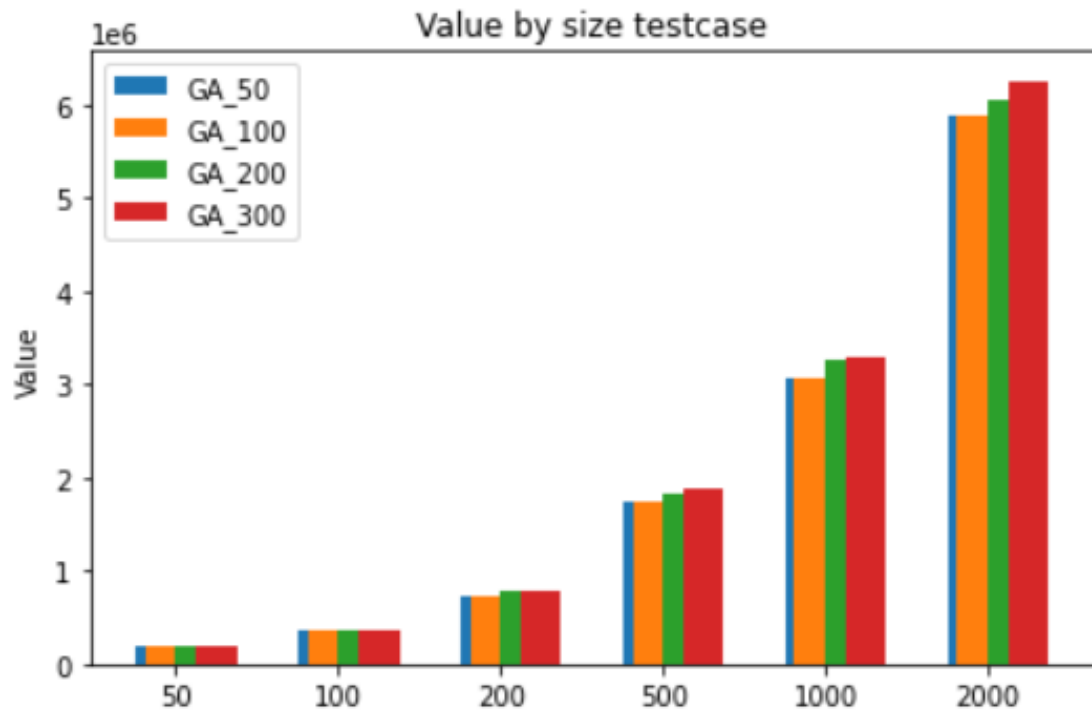
Result table for Uncorrelated group

12Circle	50	129609	8640160	22
12Circle	100	254769	16983730	47
12Circle	200	513049	34201513	87
12Circle	500	1230885	82054788	225
12Circle	1000	2443749	1.63E+08	457
12Circle	2000	4770600	3.18E+08	943

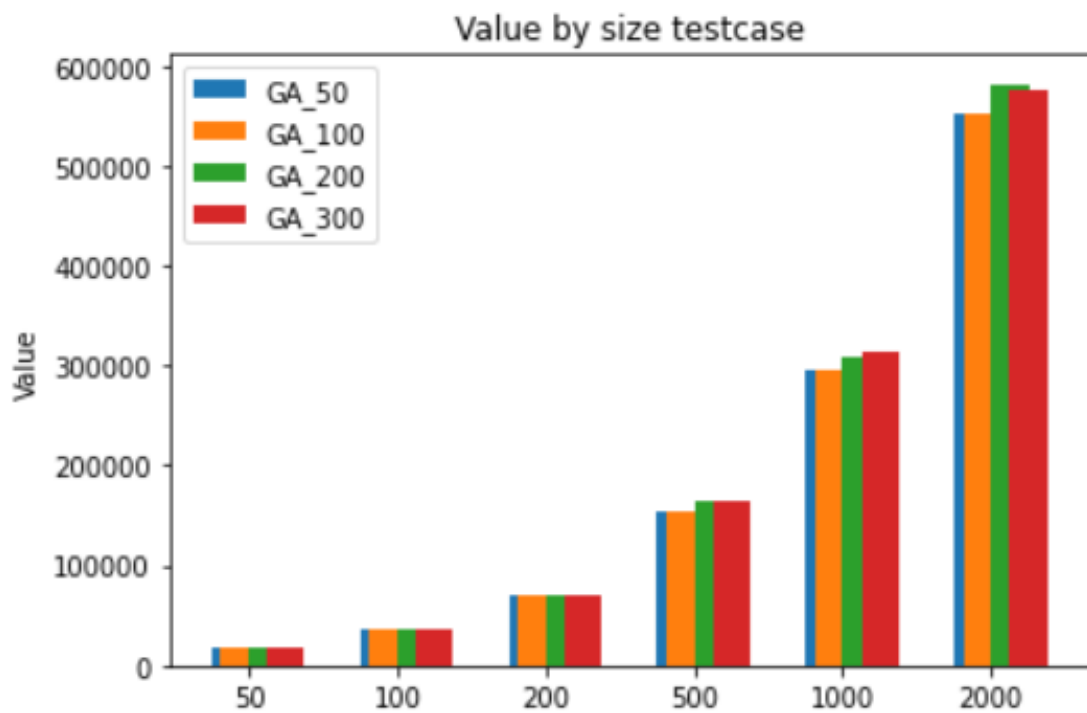
Result table for Uncorrelated group

Ta sẽ trực quan hóa kết quả của các test group test case để dễ hình dung N

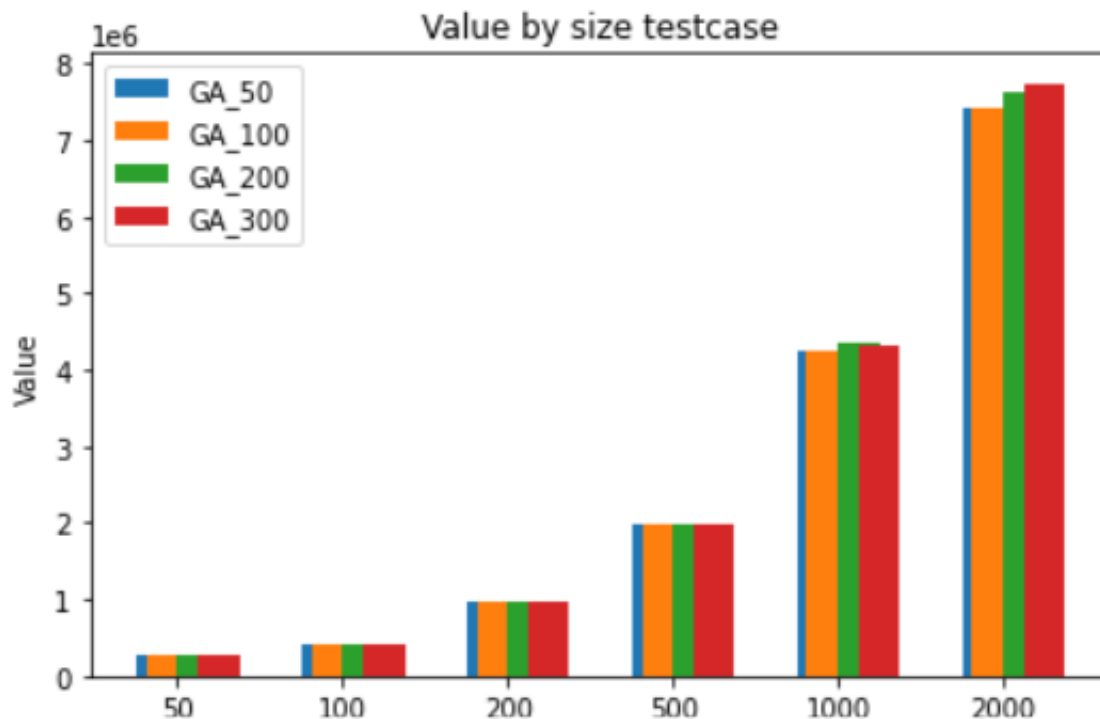
Group: 00Uncorrelated



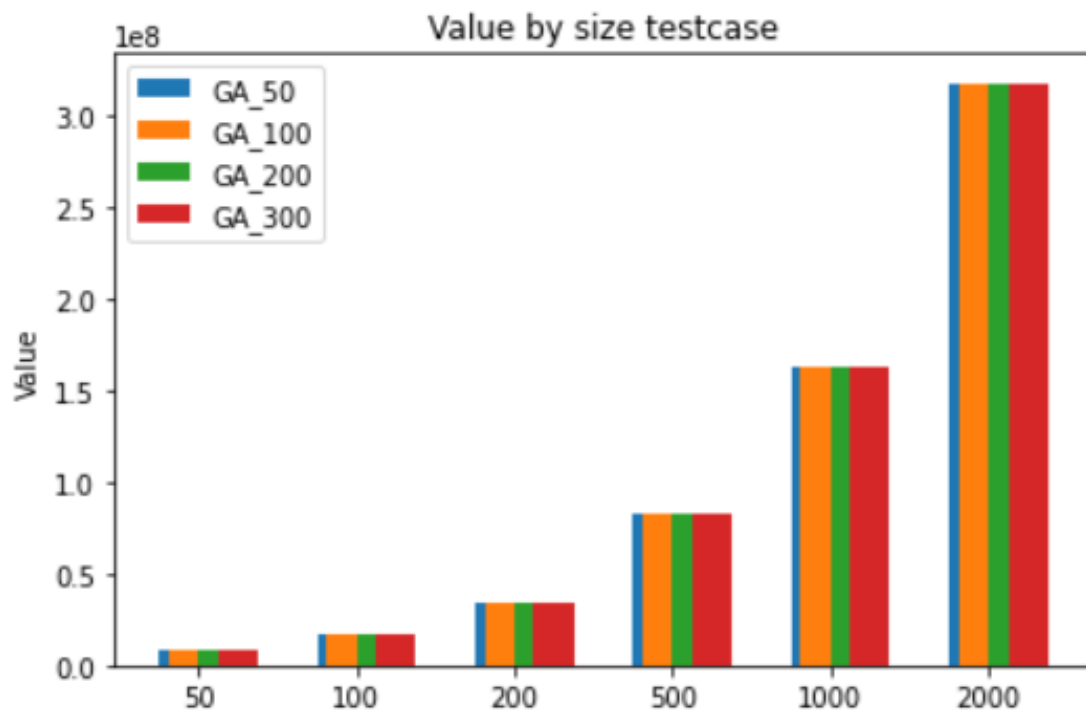
Group: 06UncorrelatedWithSimilarWeights



Group: 09SpannerStronglyCorrelated



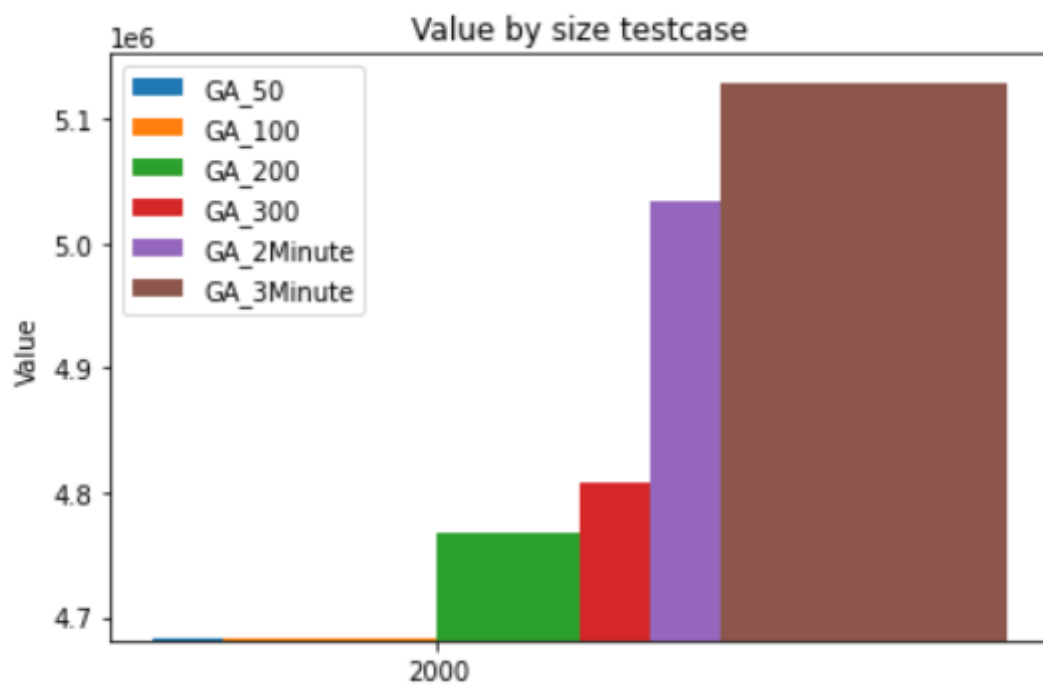
Group: 12Circle



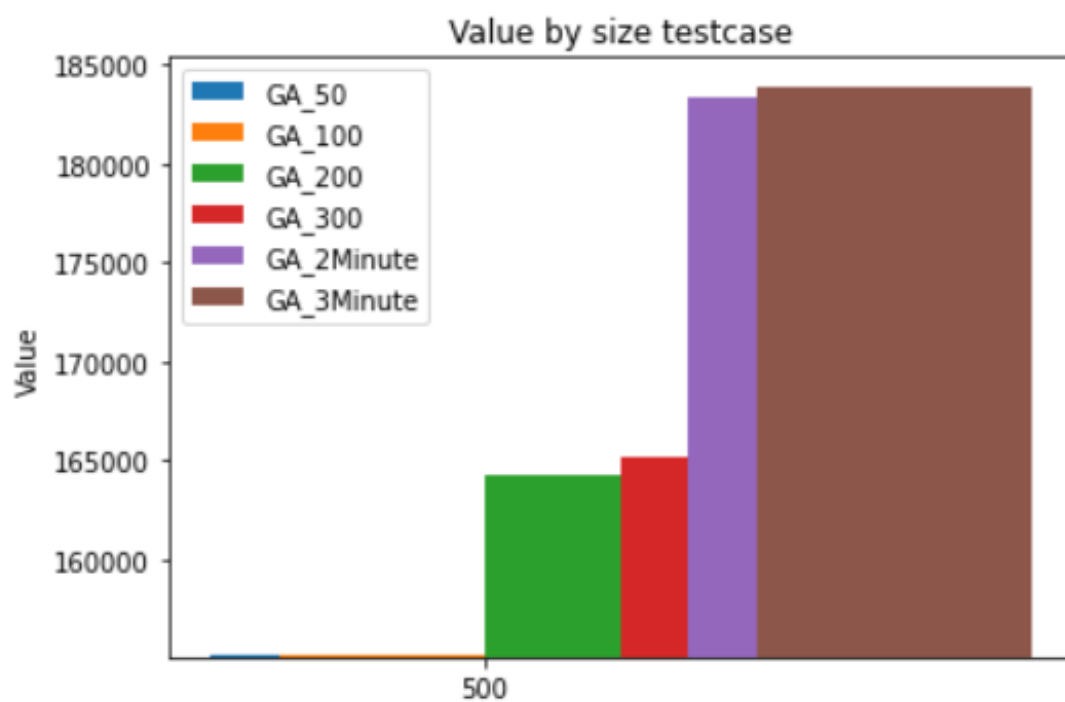
Từ đó ta nhận thấy rằng với population càng cao thì lời giải sẽ có xu hướng càng tốt (có một số trường hợp không tốt bởi vì có sự đột biến (dù tỉ lệ rất nhỏ) bù lại thời gian thực thi sẽ chậm hơn. Vì vậy ta sẽ chọn lời giải POPULATION SIZE=300 để so sánh với việc sử dụng OR tools.

Ngoài ra với Max Generation=50 thì ở GA thời gian thực thi vẫn chưa phù hợp với thời gian giới hạn 2 phút. Vì thế ta sẽ set lại Max Generation là một số lớn để bài toán có thể chọn lọc ra lời giải tối ưu hơn. Chúng ta sẽ trực quan kết quả của các loại GA (2 loại cuối là khi đã tune điều chỉnh max generation hợp lý ứng với thời gian giới hạn là 2 và 3 phút) để xem việc tăng max generation hiệu quả không.

Group: 08SpannerWeaklyCorrelated
Size test: 2000
2000



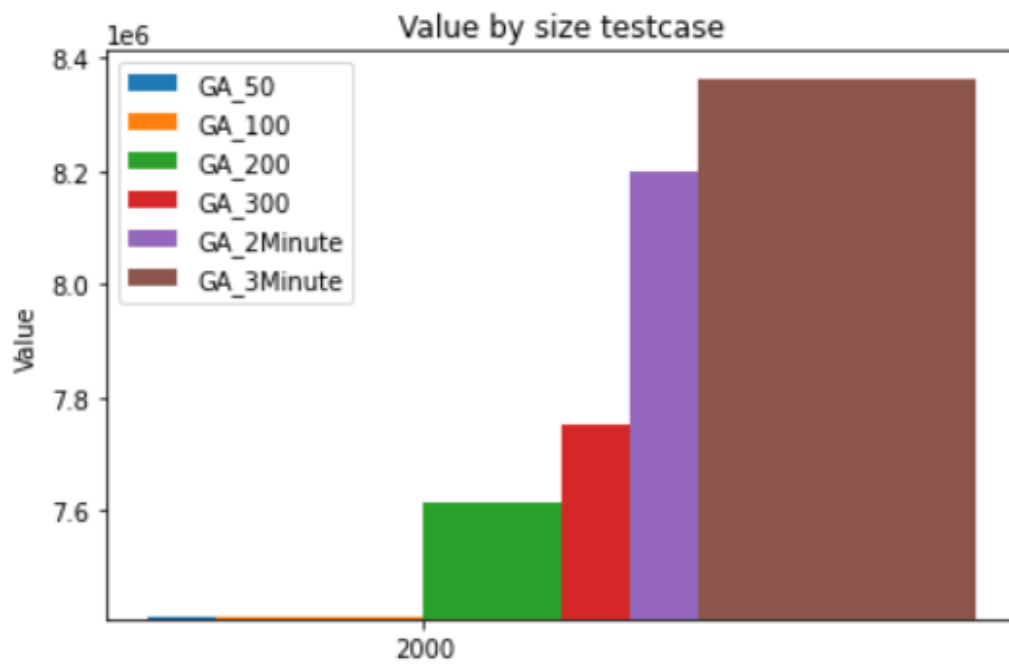
Group: 06UncorrelatedWithSimilarWeights
Size test: 500
500



Group: 09SpannerStronglyCorrelated

Size test: 2000

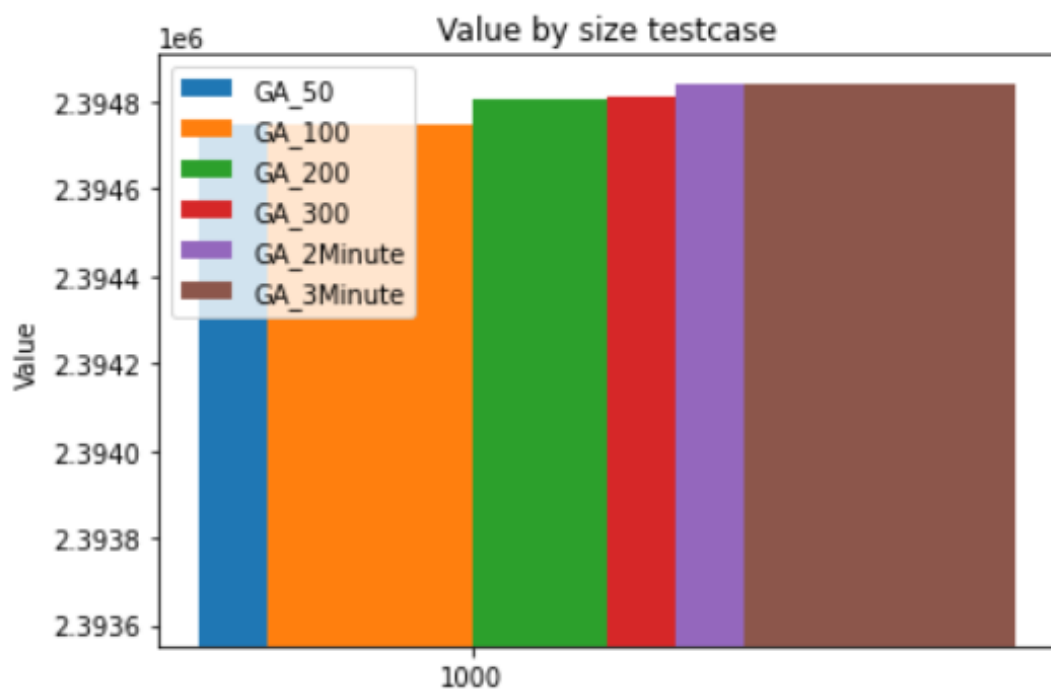
2000



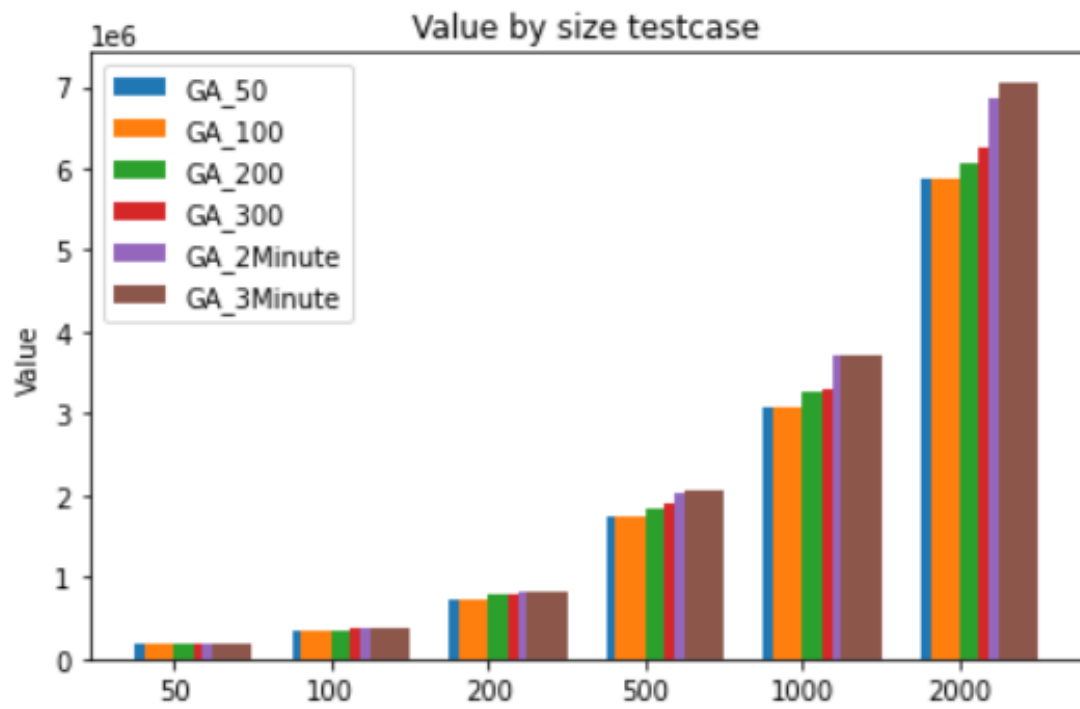
Group: 11ProfitCeiling

Size test: 1000

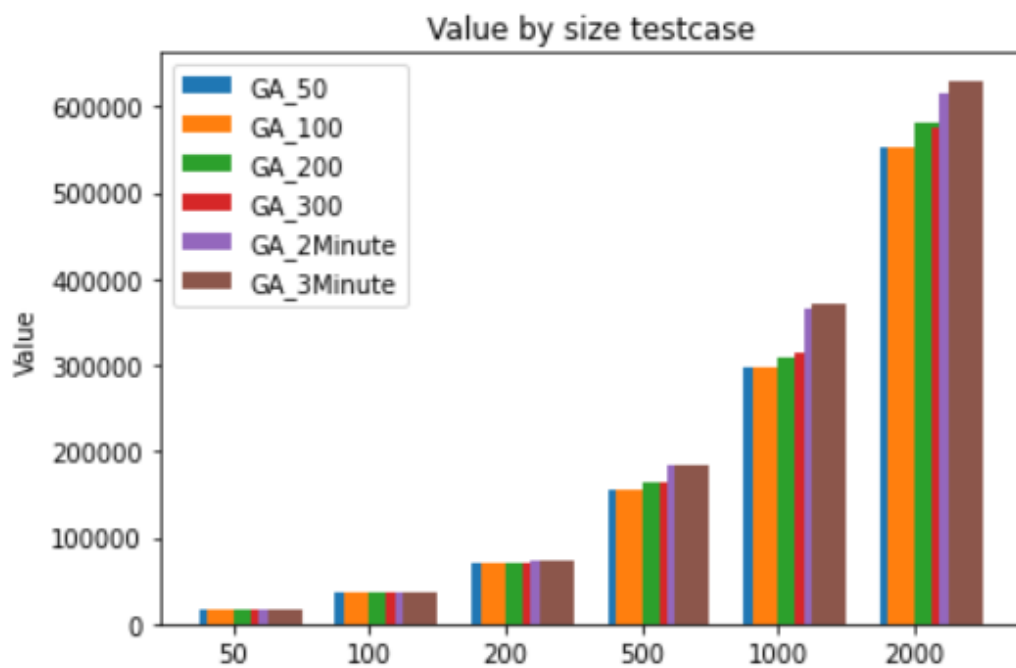
1000



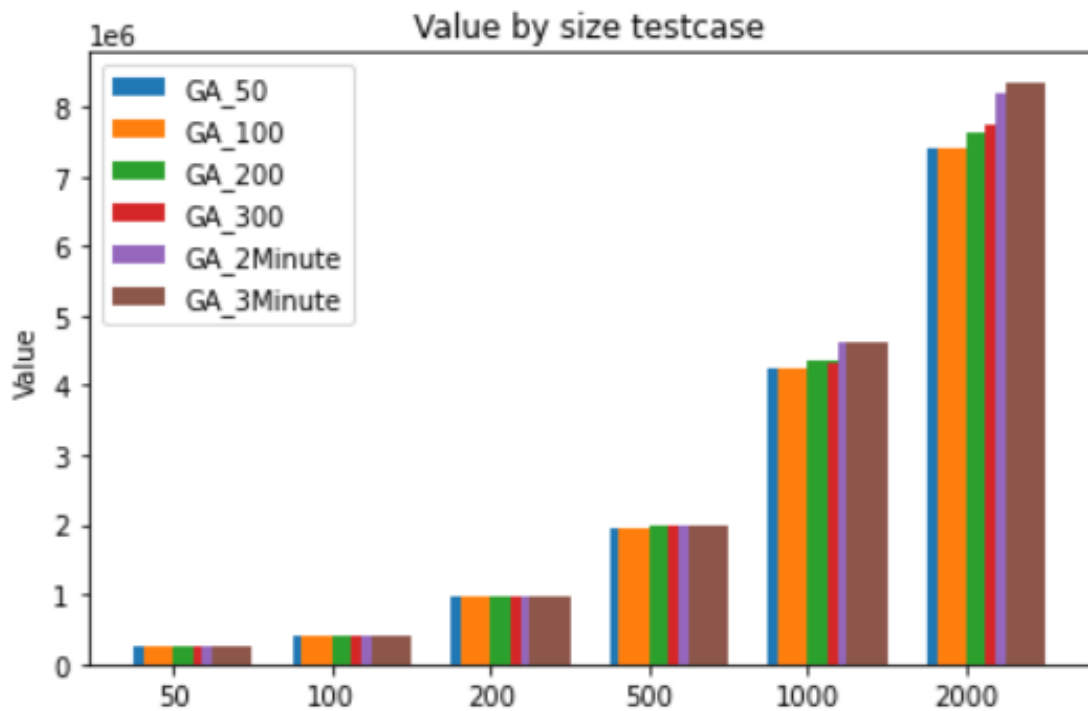
Group: 00Uncorrelated



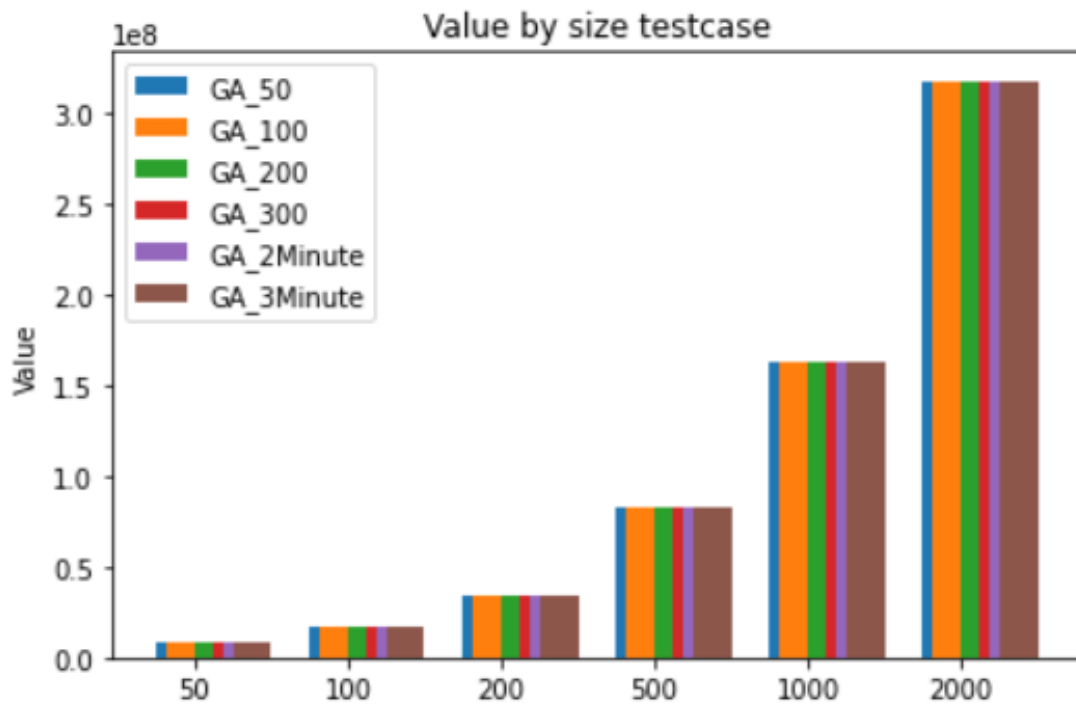
Group: 06UncorrelatedWithSimilarWeights



Group: 09SpannerStronglyCorrelated



Group: 12Circle



Từ ta có thể thấy ở hầu hết các group thì kết quả của việc tối ưu maxgeneration là sẽ tốt hơn(việc để maxgeneration thấp sẽ khiến không tận dụng hết lượng populationsize lớn vì với số generation ít thì việc chọn lọc ra các lời giải tốt sẽ chưa có kết quả cao(kết quả ngày một cải thiện qua mỗi đời nhưng chưa cao)

3 Sử dụng OR Tool để giải Knapsack

OR Tools là một công cụ mạnh mẽ được Google phát triển để giải quyết một số vấn đề về các bài toán tối ưu tổ hợp Kết quả về độ chính xác của OR tools là rất mạnh mẽ(về độ chính xác thì không kém các cách giải truyền thống như QHĐ, quay lui nhưng thời gian thực thi tốt hơn Ta sẽ xem kết quả bài toán ứng với 3 loại group test 4 loại group test đầu tiên(set timelimit là 2 phút)

group	n	weight	value	item
00Uncorrelated	50	130802	180893	31
00Uncorrelated	100	240546	363530	60
00Uncorrelated	200	506796	814631	118
00Uncorrelated	500	1266192	2078380	302
00Uncorrelated	1000	2496944	4030476	622
00Uncorrelated	2000	4938250	8143634	1247
01WeaklyCorrelated	50	131960	147878	23
01WeaklyCorrelated	100	247799	273154	51
01WeaklyCorrelated	200	491651	548022	107
01WeaklyCorrelated	500	1269755	1408203	264
01WeaklyCorrelated	1000	2512518	2754816	498
01WeaklyCorrelated	2000	4859858	5376960	1019
02StronglyCorrelated	50	116210	151210	35
02StronglyCorrelated	100	237227	308227	71
02StronglyCorrelated	200	475215	616215	141
02StronglyCorrelated	500	1224367	1577367	353
02StronglyCorrelated	1000	2488338	3188338	700
02StronglyCorrelated	2000	4962178	6370178	1408
03InverseStronglyCorrelated	50	164700	147700	17
03InverseStronglyCorrelated	100	299600	267600	32
03InverseStronglyCorrelated	200	613588	549588	64
03InverseStronglyCorrelated	500	1498585	1339585	159
03InverseStronglyCorrelated	1000	2880139	2576139	304
03InverseStronglyCorrelated	2000	6022477	5386477	636

4 So sánh kết quả của OR Tools và Genetic Algorithm

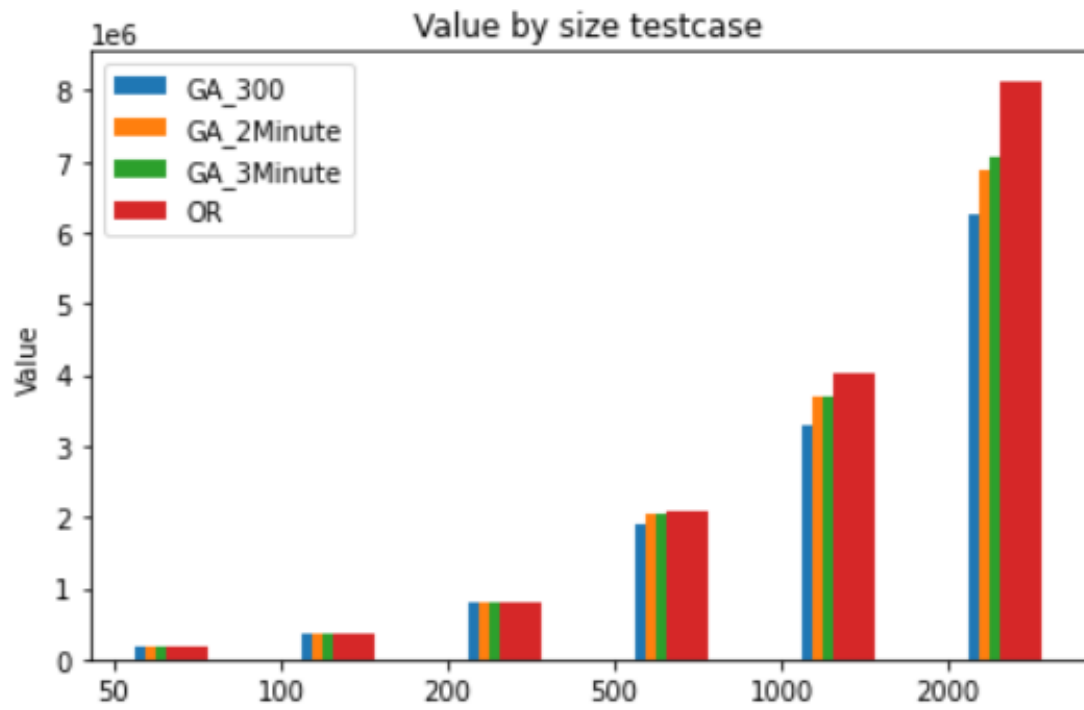
Với cùng khoảng thời gian limit là 2 phút ta có kết quả như sau giữa lời giải Genetic(generation=50 và poplation=300) và OR :

group	OR_Tools	GA	MSE
00Uncorrelated	180893	180893	0
00Uncorrelated	363530	360024	-3506
00Uncorrelated	814631	788146	-26485
00Uncorrelated	2078380	1884684	-193696
00Uncorrelated	4030476	3302226	-728250
00Uncorrelated	8143634	6267071	-1876563
01WeaklyCorrelated	147878	147110	-768
01WeaklyCorrelated	273154	269945	-3209
01WeaklyCorrelated	548022	544401	-3621
01WeaklyCorrelated	1408203	1373491	-34712
01WeaklyCorrelated	2754816	2638240	-116576
01WeaklyCorrelated	5376960	5082111	-294849
02StronglyCorrelated	151210	151210	0
02StronglyCorrelated	308227	306224	-2003
02StronglyCorrelated	616215	610203	-6012
02StronglyCorrelated	1577367	1544235	-33132
02StronglyCorrelated	3188338	3093333	-95005
02StronglyCorrelated	6370178	6112023	-258155
03InverseStronglyCorrelated	147700	147652	-48
03InverseStronglyCorrelated	267600	265538	-2062
03InverseStronglyCorrelated	549588	546406	-3182
03InverseStronglyCorrelated	1339585	1319578	-20007
03InverseStronglyCorrelated	2576139	2484019	-92120
03InverseStronglyCorrelated	5386477	5172588	-213889

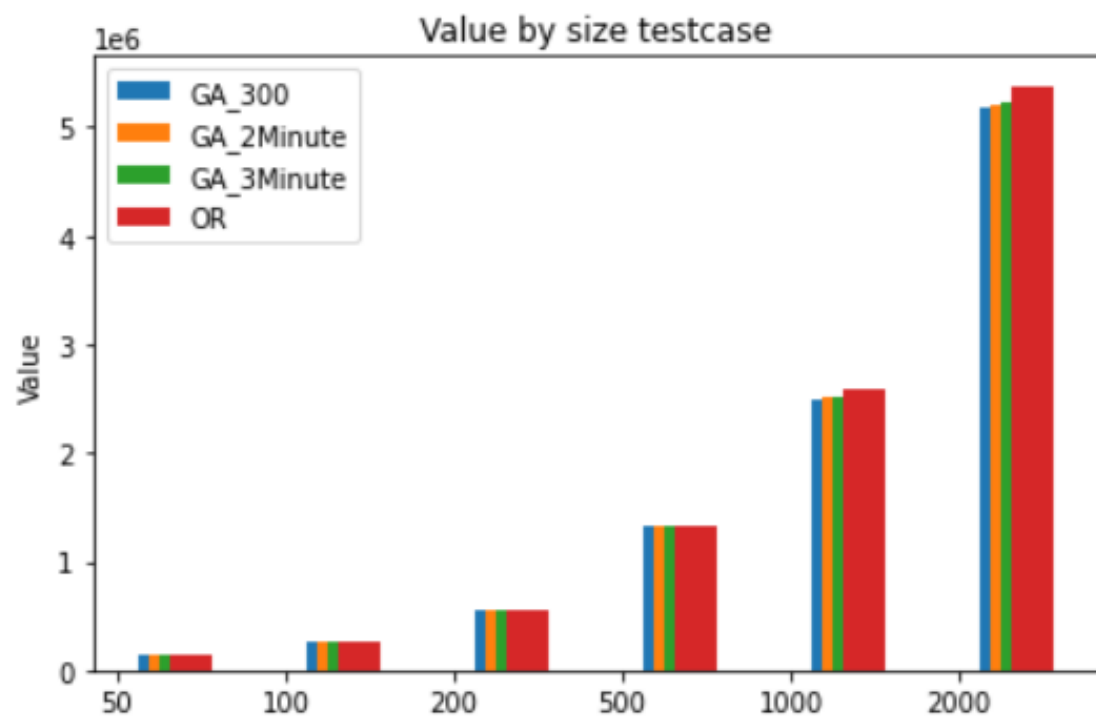
Từ đó ta thấy rằng với các test case nhỏ thì kết quả bài toán sẽ ít chênh lệch và khi số lượng vật ngày càng nhiều hơn thì sẽ có sự chênh lệch càng lớn(mặc dù thời gian chạy của các test OR chậm hơn)

Ta sẽ xem kết quả trực quan sau khi tối ưu GA(so sánh GA300,GA2minute,GA3minute) và OR:

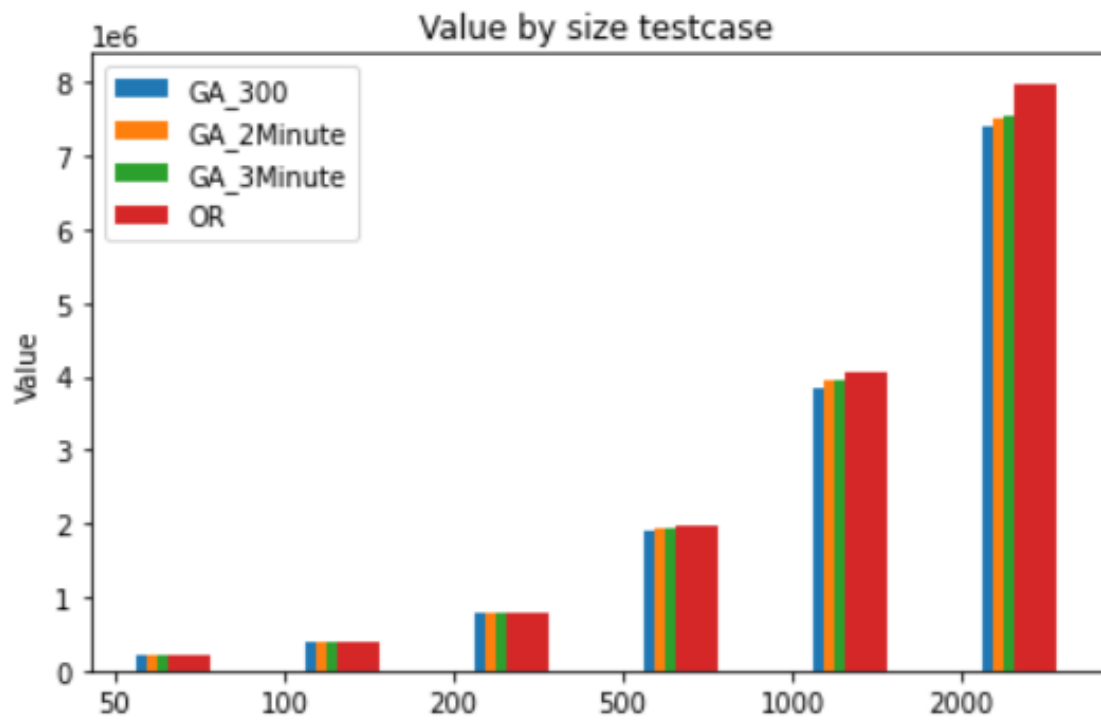
Group: 00Uncorrelated



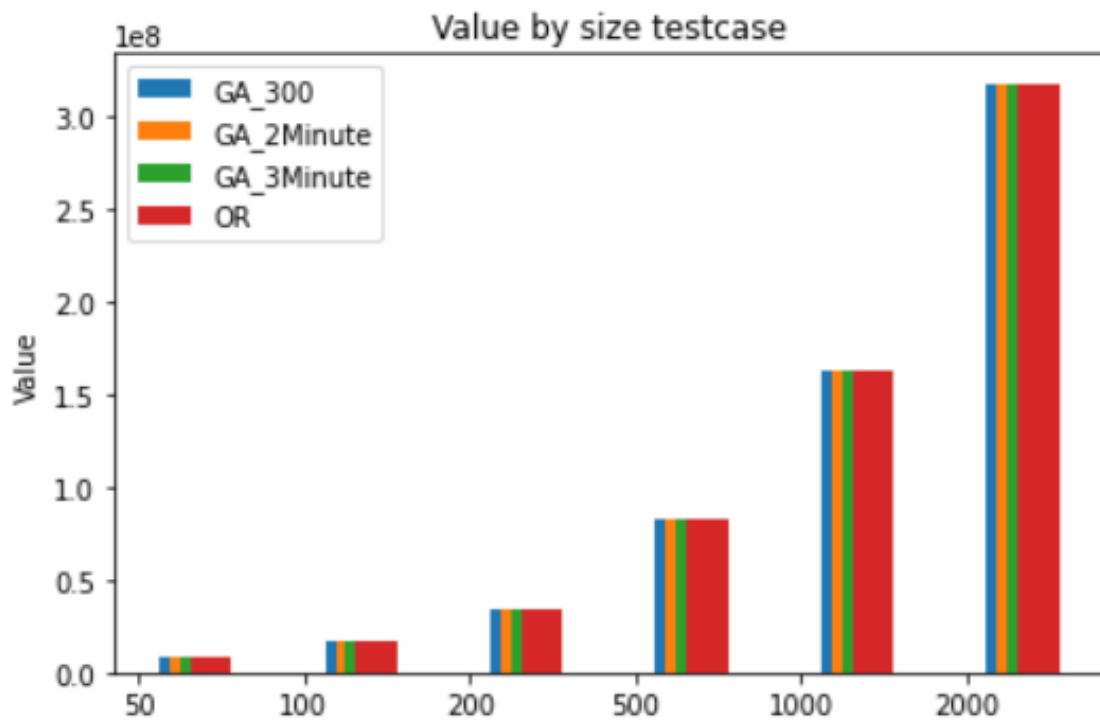
Group: 03InverseStronglyCorrelated



Group: 10MultipleStronglyCorrelated



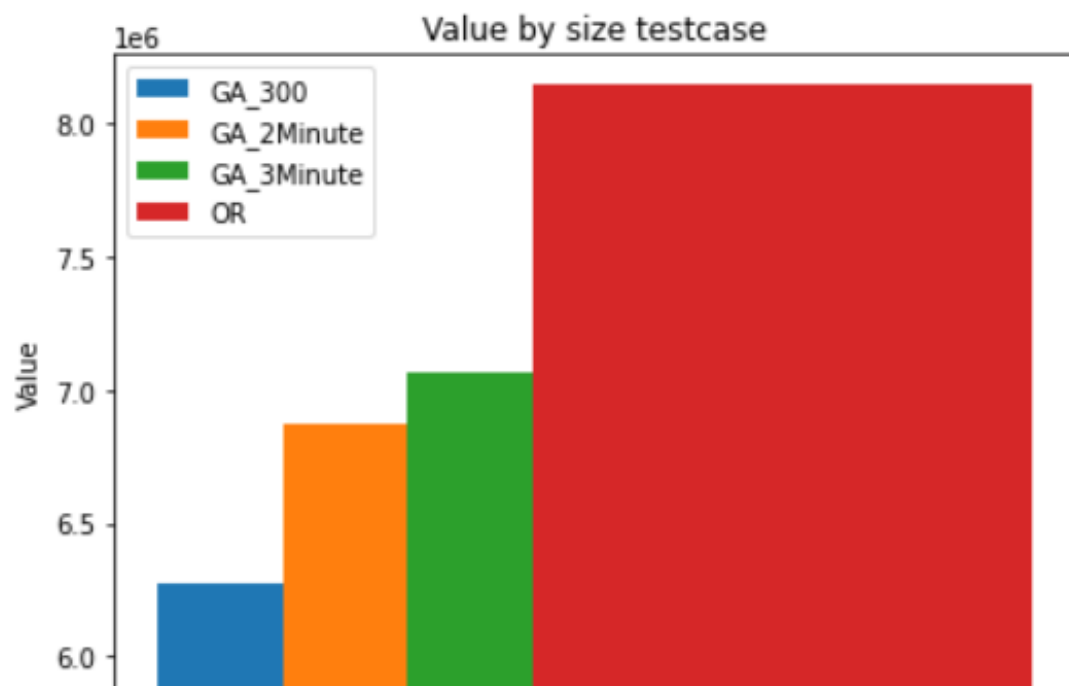
Group: 12Circle



Group: 00Uncorrelated

Size test: 2000

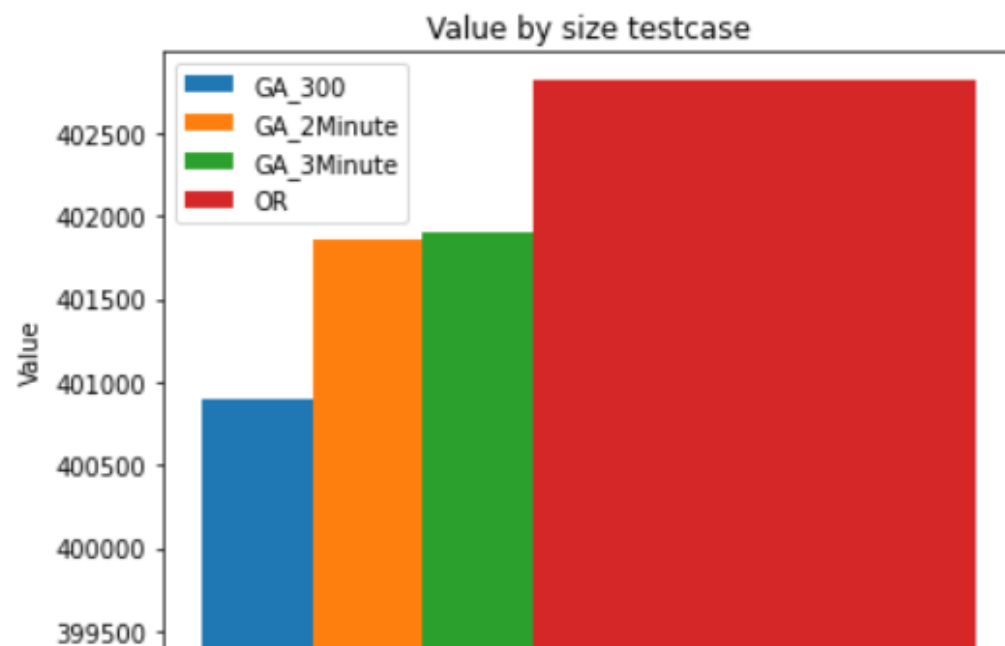
2000



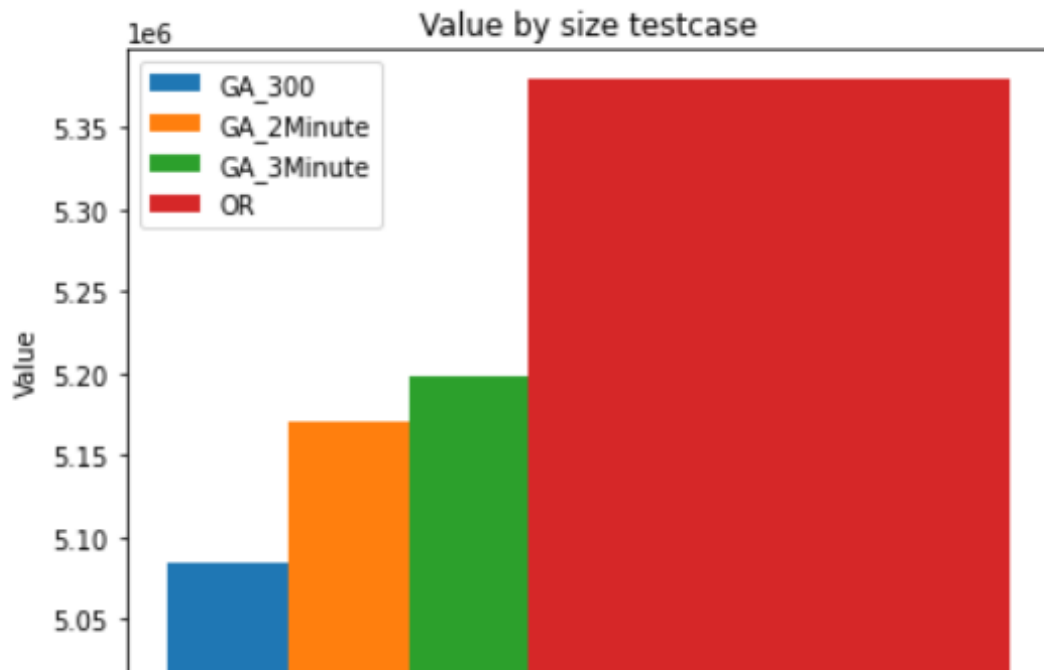
Group: 10MultipleStronglyCorrelated

Size test: 100

100



Group: 01WeaklyCorrelated
Size test: 2000
2000



Từ đó ta thấy rằng thuật toán OR hoạt động tốt hơn GA với $\text{population} \leq 300$, tuy nhiên nếu ta tăng population size càng cao thì liệu lời giải của GA có thể sẽ càng tốt (Kiểm chứng bằng thực nghiệm). Vì vậy để cải thiện chất lượng lời giải ta có thể tăng giá trị populationsize để có thể có nhiều lời giải ban đầu nhằm dễ dàng chọn lọc và cải thiện chất lượng hơn qua mỗi đời.

5 Tổng kết

Kết quả thuật toán Genetic là chưa tối ưu vì số lượng generation vẫn còn ít và POPULATIONSIZE vẫn còn nhỏ, và nếu ta có thể tăng thời gian giới hạn và POPULATIONSIZE đủ lớn thì sẽ tìm ra được lời giải tối ưu, riêng trong tình huống thời gian và kích thước giới hạn như trên thì lời giải của OR tool sẽ tốt hơn một chút so với Genetic. Tuy vậy việc tối ưu population size đã nói ở trên vẫn có một vấn đề rằng liệu với một mức populationsize nào đó thì chất lượng lời giải có cải thiện nữa không. Cần phải thực nghiệm thêm nhiều lần để đưa ra câu trả lời chính xác cho vấn đề này.