Lab Report 1 Submission (2019)

Sylvain Lapeyrade, Reda Bourakkadi (sylla801, redbo196)

December 1, 2019

Answers

1) Exercise 1: Coalition structure generation

You should plot and discuss your results in the report—e.g., what is the solution quality of the different algorithms, and what type of problems are they suitable for?

After implementing both the random and greedy search algorithms, we obtained the results detailed in Figure 1. We can see at first view that the **Greedy Search** algorithm does not find the best utility value for each problem and for each number of agents which is respectively 15 for problems 1 and 2, and 20 for problems 3 and 4.

Indeed, for the problem 1, the **Greedy Search** is a bit lower than the **Random Search** value with 10 000 and 100 000 evaluations (i.e. $7.8364 > 8.22902 > 8.65678$). However, for the problem 2, the tendency is inverted (i.e. Random Search with 100 000 evals: $9.11907 < 10.0136$ for the Greedy Search) while the number of agents remain the same.

	CSG-15-NPD-1	CSG-15-NPD-2	CSG-20-NPD-1	CSG-20-NPD-2
Random Search:				
100 evaluations	6.84732	7.34286	8.57318	8.3931
1000 evaluations	7.63916	8.07568	9.44656	8.90211
10000 evaluations	8.22902	8.55365	9.61898	9.81114
100000 evaluations	8.65678	9.11907	10.5505	10.639
Greedy Search	7.8364	10.0136	12.1601	11.7127

Figure 1: Data obtained from the coalition structure generation exercise.

The data plotted from these experiments in Figure 2 shows the global tendency from the results of the two algorithms. It can be very well seen that for the **Random Search** algorithm, the more evaluations we have, the better the utility value. It seems logical since there is more chance to have a better result. As for the **Greedy Search**, overall it seems to outperform the **Random Search**, but it looks like the more agents there are, the better it is in comparison. Which is also logical since, as the complexity increases (i.e. the number of solutions), one can expect a random algorithm to have greater difficulties to find a very good solution than a structured algorithm.

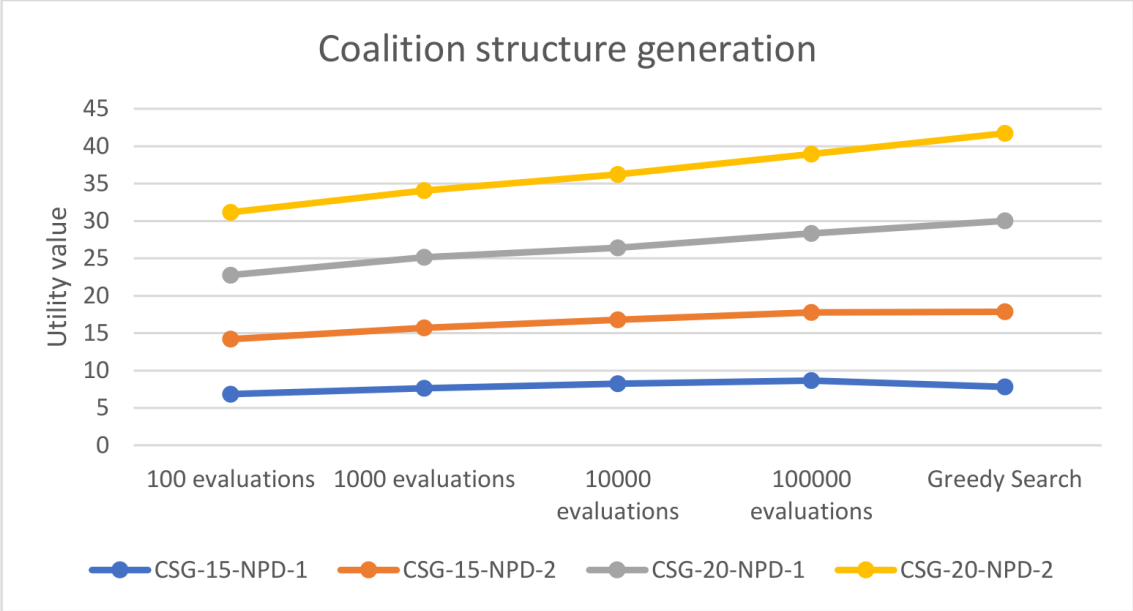

Figure 2: Utility values plotted with their relative algorithms and number of evaluations.

Thus, for a small problem, the **Random Search** can be more suitable complexity and computationally wise. Whereas the **Greedy Search** should be privileged for bigger problems.

2) Exercise 2: Coordinating coalitions

You should plot and discuss your results in the lab report from benchmarking the algorithms with the following problem sets.

This problem is a bit different since it is not the number of agents which increases from problems 1 & 2 to problems 3 & 4 but the number of *task*. The results of our implementation of the two algorithms are presented in Figure 3. Looking at the results, we can observe that **Greedy Search** is systematically better than **Random Search** for as much as 100 000 evaluations.

	SCSGA-15-10-NPD-1	SCSGA-15-10-NPD-2	SCSGA-15-30-NPD-1	SCSGA-15-30-NPD-2
Random Search:				
100 evaluations	7.33926	7.23491	17.5223	18.803
1000 evaluations	7.81732	7.68351	18.7312	19.9256
10000 evaluations	8.16512	7.99161	19.0879	19.9256
100000 evaluations	8.41165	8.38352	20.2034	21.0484
Greedy Search:	8.98147	8.83272	23.0142	24.1854

Figure 3: Data obtained from the Coordinating coalitions exercise.

By plotting the results in Figure 4, we can find the same tendency as for **Exercise 1**. Indeed, the **Greedy Search** seems to produced results a bit better than **Random Search** for smalls numbers of tasks, whereas the results difference is greater for a greater number of tasks.

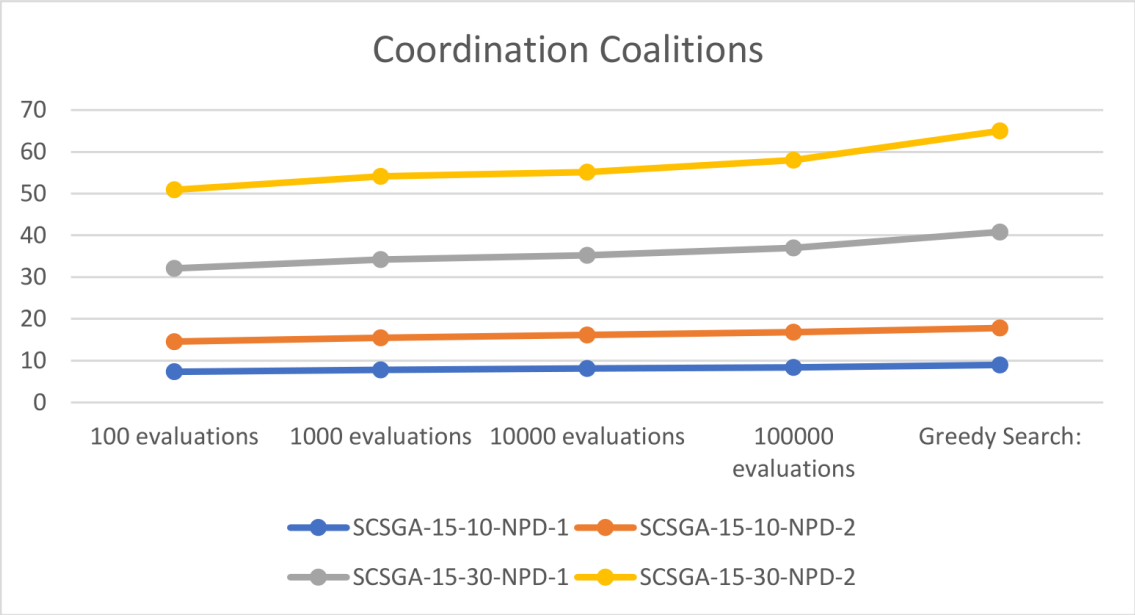

Figure 4: Utility values plotted with their relative algorithms and number of evaluations.

In conclusion, for smalls numbers of tasks, the **Random Search** seems to be good enough but not the best, it can be again chosen for complexity or computational reasons. For larger number of tasks though, the **Greedy Search** will be better and better, hence should be privileged.