Report of Assignment – 3

Performance o MAX_CUT Using GRASP

Roll: 1905120

The maximum cut (MAX-CUT) problem

Given an undirected graph, G = (V, U), where V is the set of vertices and U is the set of edges, and weights w_{uv} associated with each edge (u, v) ∈ U, the *maximum cut* (MAX-CUT) problem consists in finding a nonempty proper subset of vertices S ⊂ V (S ≠ ∅), such that the weight of the cut $(S, \bar{S})$, given by $w(S, \bar{S}) = \sum_{u \in S, v \in \bar{S}} w_{uv}$, is maximized.

To find **MAX_CUT** of a graph, I used these 8 algorithms:

Constructive Algorithm:

- Random
- Greedy
- Semi-greedy

Local Search:

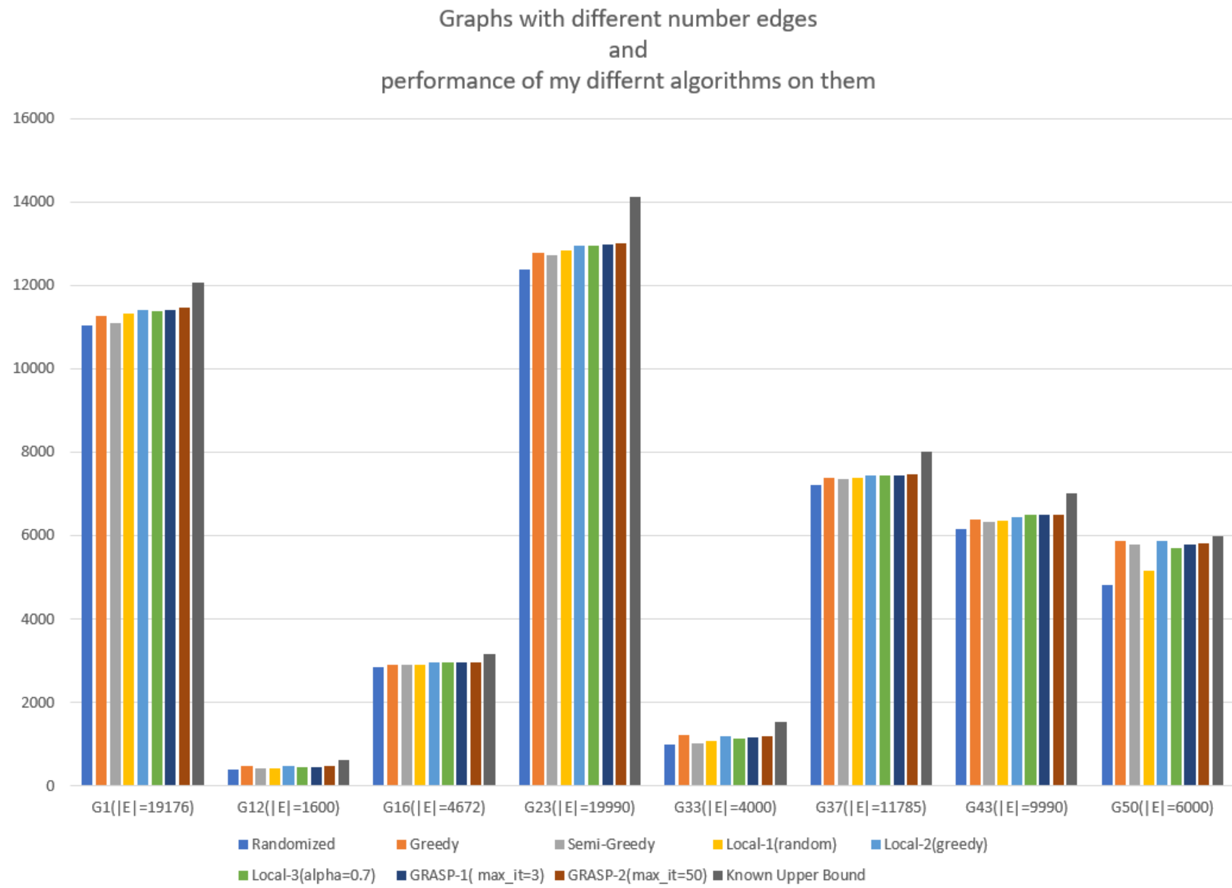
- LS:1 with random initialization
- LS:2 with greedy initialization,
- LS:3 with semi-greedy (α = 0.7)

GRASP:

- GRASP-1 with maximum iteration = 3
- GRASP-2 with maximum iteration = 50

	Problem			Constructive Algorithm			Local search						GRASP			My Best Value	Upper Bound
Name	|V|	|E|	Randomized	Greedy	Semi-Greedy	Local-1(random)		Local-2(greedy)		Local-3(alpha=0.7)		GRASP-1(max_it=3)		GRASP-2(max_it=50)			
						Iteration	Best Value	Iteration	Best value	Iteration	Best value	Iteration	Best value	Iteration	Best value		
G1	800	19176	11032	11268	11108	168	11316	60	11417	82	11382	286	11421	4572	11466	11466	12078
G2	800	19176	11091	11292	11203	154	11344	61	11365	121	11426	282	11454	4097	11463	11463	12084
G3	800	19176	11049	11295	11229	148	11316	108	11447	121	11388	281	11418	4404	11469	11469	12077
G4	800	19176	10974	11231	11199	167	11391	69	11400	113	11421	261	11456	4726	11489	11489	
G5	800	19176	11070	11264	11277	166	11400	64	11407	138	11426	258	11378	4101	11468	11468	
G6	800	19176	1501	1744	1722	174	1851	121	1910	111	1924	334	1951	5322	2009	2009	
G7	800	19176	1413	1601	1502	154	1708	83	1797	131	1721	324	1780	5216	1854	1854	
G8	800	19176	1410	1609	1585	195	1736	92	1801	158	1787	274	1808	4969	1868	1868	
G9	800	19176	1471	1631	1515	180	1783	92	1807	120	1800	334	1925	5100	1904	1925	
G10	800	19176	1247	1599	1561	146	1746	87	1759	123	1809	270	1785	5033	1841	1841	
G11	800	1600	390	480	428	11	426	6	480	19	470	10	478	392	494	494	627
G12	800	1600	406	474	434	10	418	7	494	7	452	21	470	407	482	494	621
G13	800	1600	434	510	470	17	460	3	508	10	472	19	500	422	506	510	645
G14	800	4694	2865	2949	2944	42	2921	21	2968	24	2973	70	2973	1210	2988	2988	3187
G15	800	4661	2842	2915	2918	52	2921	18	2932	15	2944	57	2956	1248	2976	2976	3169
G16	800	4672	2855	2914	2923	60	2917	29	2955	24	2960	70	2967	1246	2980	2980	3172
G17	800	4667	2875	2925	2917	46	2933	22	2948	21	2947	63	2948	1263	2972	2972	
G18	800	4694	700	816	744	70	782	51	866	64	840	135	859	2169	911	911	
G19	800	4661	587	724	668	89	778	20	798	48	784	123	789	2073	824	824	
G20	800	4672	628	796	751	64	755	43	797	53	748	117	828	2082	858	858	
G21	800	4667	622	748	716	119	813	40	824	38	766	119	797	2168	849	849	
G22	2000	19990	12460	12810	12784	246	12839	121	12968	89	12911	362	12984	5441	13016	13016	14123
G23	2000	19990	12396	12790	12729	238	12852	126	12965	132	12958	311	12990	5499	13005	13005	14129
G24	2000	19990	12290	12820	12719	259	12877	91	12906	108	12882	348	12934	5679	13023	13023	14131
G25	2000	19990	12311	12773	12768	256	12865	100	12943	143	12980	306	12963	5500	13013	13013	
G26	2000	19990	12332	12772	12698	250	12801	115	12953	125	12947	339	12959	5438	12990	12990	
G27	2000	19990	2305	2712	2604	309	2782	133	3005	196	2839	447	2921	7323	2944	3005	
G28	2000	19990	2281	2670	2474	305	2766	106	2861	165	2818	503	2884	7535	2938	2938	
G29	2000	19990	2314	2734	2614	293	2928	103	2962	138	2881	475	2994	7830	3045	3045	
G30	2000	19990	2390	2808	2599	246	2815	139	2950	171	2974	542	3007	7549	3037	3037	
G31	2000	19990	2340	2647	2599	290	2844	103	2759	170	2812	488	2898	7688	2982	2982	
G32	2000	4000	1032	1196	1112	33	1086	16	1218	28	1132	54	1198	916	1230	1230	1560
G33	2000	4000	998	1214	1024	52	1074	14	1194	32	1128	39	1182	922	1202	1214	1537
G34	2000	4000	998	1176	986	47	1072	20	1190	24	1144	51	1180	945	1192	1192	1541
G35	2000	11778	7202	7373	7366	115	7362	51	7450	69	7466	176	7463	2823	7476	7476	8000
G36	2000	11766	7170	7385	7371	131	7351	54	7427	65	7454	181	7432	2805	7468	7468	7996
G37	2000	11785	7209	7380	7351	138	7382	63	7454	75	7458	179	7457	2777	7479	7479	8009
G38	2000	11779	7196	7369	7364	125	7349	53	7460	60	7428	155	7443	2954	7480	7480	
G39	2000	11778	1705	2002	1832	206	2038	80	2147	125	2060	318	2143	5724	2171	2171	
G40	2000	11766	1715	2005	1884	194	1995	98	2130	118	2052	311	2137	5185	2149	2149	
G41	2000	11785	1679	1934	1816	198	1987	73	2136	125	2031	293	2080	4746	2143	2143	
G42	2000	11779	1773	2129	1801	208	2067	48	2184	109	2113	315	2184	5194	2238	2238	
G43	1000	9990	6170	6392	6347	102	6361	74	6461	83	6492	172	6505	2922	6511	6511	7027
G44	1000	9990	6116	6421	6343	140	6400	52	6455	53	6406	191	6471	3109	6514	6514	7022
G45	1000	9990	6196	6380	6359	183	6410	42	6446	85	6464	133	6483	2912	6512	6512	7020
G46	1000	9990	6205	6389	6385	129	6411	35	6425	44	6452	175	6483	2997	6494	6494	
G47	1000	9990	6134	6391	6375	95	6389	49	6493	54	6460	162	6472	3019	6508	6508	
G48	3000	6000	5004	6000	5816	70	5096	1	6000	2	5742	6	5690	143	6000	6000	6000
G49	3000	6000	4952	6000	5810	68	5148	1	6000	4	5738	4	5694	121	5930	6000	6000
G50	3000	6000	4828	5880	5798	87	5156	1	5880	5	5716	9	5788	120	5856	5880	5988
G51	1000	5909	3609	3679	3679	62	3671	29	3725	26	3712	94	3751	1568	3748	3751	
G52	1000	5916	3627	3711	3690	79	3708	24	3743	23	3734	83	3738	1509	3761	3761	
G53	1000	5914	3623	3685	3677	56	3688	25	3740	37	3732	86	3742	1391	3757	3757	
G54	1000	5916	3615	3706	3690	58	3675	20	3733	27	3724	88	3740	1461	3754	3754	

Figure: The resultant table



Observation:

1. From the table it is clear, that the best solutions among all these algorithms are almost near Upper Bounds.

   In fact, 2 cases (G48 and G49) were able to achieve the upper bound.

Name	|V|	|E|	My Best Value	Upper Bound
G48	3000	6000	6000	6000
G49	3000	6000	6000	6000

2. In 46 cases out of 54 cases the best Maximum Cut value among all the algorithms came from **GRASP-2 with maximum iteration=50** which is colorless section in **MY BEST VALUE** column.

3. In 8 cases out of 54 cases, are highlighted using red color in the **MY BEST VALUE** column. The best value came from either Greedy or LS-2 with greedy initialization.

The common characteristic between these cases: They have lower edge count than other cases.

4. The most noticeable trend in most cases is the improvement of Maximum Cut value as we go from left side of the table to right side (few except: greedy > semi-greedy and LS-2 > LS-3) of the table.

So, our most efficient algo sequence:

1. GRASP-2 with maximum iteration = 50
2. GRASP-1 with maximum iteration = 3
3. LS-2 with greedy initialization
4. LS-3 with semi-greedy ($\alpha = 0.7$)
5. LS-1 with random initialization
6. Greedy
7. Semi-greedy
8. Random

Explanation:

This result is predictable as constructive algorithm always gives the best solution in one iteration.

Local search is an improvement over constructive algorithm as it always takes us to local optimum.

GRASP is further improvement of local search as it random initialize and start local search a few times iteratively.

Greedy performs better than Semi-greedy because greedy aims at acquiring the best optimum always where semi-greedy will randomly select from Restricted Candidate list which may not always be the best optimum value.

Conclusion:

GRASP > Local Search > Greedy > Semi-Greedy > Random