DISEÑO Y ANÁLISIS DE ALGORITMOS
Ramificación y poda

Maximum diversity problem

Alberto Cruz Luis
alu0101217734@ull.edu.es

1. Introducción

En esta práctica se implementaran algoritmos heurísticos y de ramificación y poda para el *Maximum diversity problem*. El programa se implementara en C++.

2.Algoritmos usados para resolver el problema

Para ello usaremos 4 tipos de algoritmos que iré explicando más adelante. Greedy, GRASP y Branch Bound.

2.1 Greedy

```
Elem = S;
S = ∅;
Obtener s c = centro(Elem);
repeat
    Obtener el elemento s ∗ ∈ Elem más alejado de s c ;
    S = S ∪ {s ∗ };
    Elem = Elem − {s ∗ };
    Obtener s c = centro(S);
until (|S| = m)
Devolver S;
```

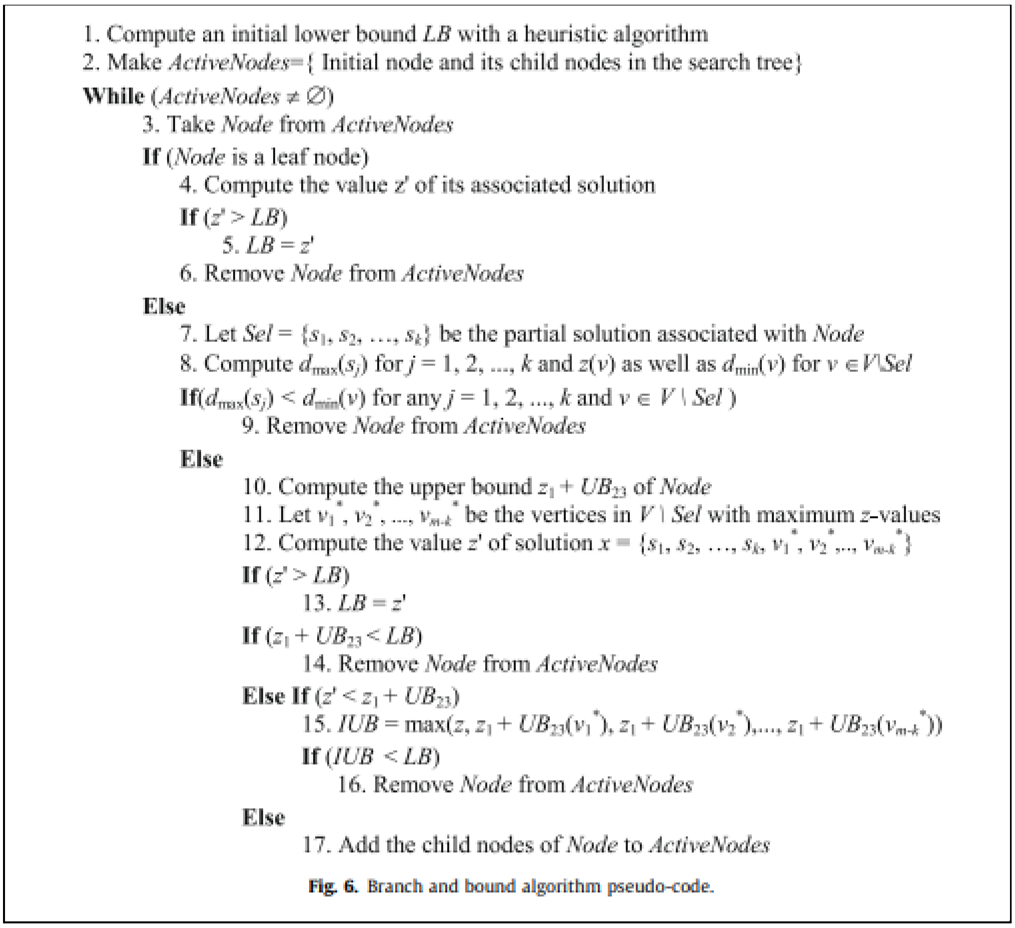
2.2 GRASP

```
procedure grasp()
InputInstance();
for GRASP stopping criterion not satisfied →
    ConstructGreedyRandomizedSolution(Solution);
    LocalSearch(Solution);
    UpdateSolution(Solution,BestSolutionFound);
rof;
return(BestSolutionFound)
end grasp;
```

```
procedure ConstructGreedyRandomizedSolution(Solution)
Solution = {};
```

```
for Solution construction not done →
MakeRCL(RCL);
s = SelectElementAtRandom(RCL);
Solution = Solution ∪ {s};
AdaptGreedyFunction(s);
rof;
end ConstructGreedyRandomizedSolution;
```

2.3 Branch & Bound



```
1. Compute an initial lower bound $LB$ with a heuristic algorithm
2. Make $ActiveNodes$={ Initial node and its child nodes in the search tree}
While ($ActiveNodes \neq \emptyset$)
        3. Take $Node$ from $ActiveNodes$
        If ($Node$ is a leaf node)
            4. Compute the value z' of its associated solution
            If ($z' > LB$)
                5. $LB = z'$
            6. Remove $Node$ from $ActiveNodes$
        Else
            7. Let $Sel = \{s_1, s_2, \dots, s_k\}$ be the partial solution associated with $Node$
            8. Compute $d_{\max}(s_j)$ for $j = 1, 2, \dots, k$ and $z(v)$ as well as $d_{\min}(v)$ for $v \in V \backslash Sel$
            If($d_{\max}(s_j) < d_{\min}(v)$ for any $j = 1, 2, \dots, k$ and $v \in V \setminus Sel$ )
                    9. Remove $Node$ from $ActiveNodes$
            Else
                    10. Compute the upper bound $z_1 + UB_{23}$ of $Node$
                    11. Let $v_1^*, v_2^*, \dots, v_{m-k}^*$ be the vertices in $V \setminus Sel$ with maximum $z$-values
                    12. Compute the value $z'$ of solution $x = \{s_1, s_2, \dots, s_k, v_1^*, v_2^*, \dots, v_{m-k}^*\}$
                    If ($z' > LB$)
                            13. $LB = z'$
                    If ($z_1 + UB_{23} < LB$)
                            14. Remove $Node$ from $ActiveNodes$
                    Else If ($z' < z_1 + UB_{23}$)
                            15. $IUB = \max(z, z_1 + UB_{23}(v_1^*), z_1 + UB_{23}(v_2^*), \dots, z_1 + UB_{23}(v_{m-k}^*))$
                            If ($IUB < LB$)
                                16. Remove $Node$ from $ActiveNodes$
                    Else
                            17. Add the child nodes of $Node$ to $ActiveNodes$
```

Fig. 6. Branch and bound algorithm pseudo-code.

4. Resultados obtenidos

En este apartado estaremos ejecutando los diferentes problemas con los algoritmos implementados.

Greedy

Problema	n	K	m	z	S	CPU
max_div_15_2.txt	15	2	5	73.5619	[8 6 3 10 1]	1.69827ms
max_div_15_2.txt	15	2	4	48.4139	[8 6 3 10]	1.20948ms
max_div_15_3.txt	15	3	5	94.7487	[11 8 4 10 13]	1.5821ms
max_div_15_3.txt	15	3	4	59.7638	[11 8 4 10]	1.26959ms
max_div_20_2.txt	20	2	5	61.2393	[17 18 8 2 12]	0.487891ms
max_div_20_2.txt	20	2	4	39.5682	[17 18 8 2]	0.392543ms
max_div_20_3.txt	20	3	5	92.8298	[12 13 7 2 16]	0.521287ms
max_div_20_3.txt	20	3	4	56.5347	[12 13 7 2]	0.418575ms
max_div_30_2.txt	30	2	5	80.9102	[8 27 1 10 12]	1.19503ms

max_div_30_2.txt	30	2	4	52.7712	[8 27 1 10]	0.987367ms
max_div_30_3.txt	30	3	5	99.5088	[16 6 23 13 14]	4.71543ms
max_div_30_3.txt	30	3	4	63.5184	[16 6 23 13]	5.05585ms

GRASP

Problema	n	K	m	Iter	LRC	z	S	CPU
max_div_15_2.txt	15	2	2	100	2	11.8592	[6 8]	9.93703 ms
max_div_15_2.txt	15	2	2	100	3	11.8592	[6 8]	12.2435 ms
max_div_15_2.txt	15	2	3	100	2	27.2313	[1 6 0]	24.5311 ms
max_div_15_2.txt	15	2	3	100	3	27.2313	[1 0 6]	18.4703 ms
max_div_15_2.txt	15	2	4	100	2	49.8268	[6 0 8 5]	39.5399 ms

max_div_15_2.txt	15	2	4	100	3	49.7633	[9 0 6 8]	41.8851 ms
max_div_15_2.txt	15	2	5	100	2	79.1295	[6 8 0 5 3]	79.7236 ms
max_div_15_2.txt	15	2	5	100	3	78.9346	[6 5 1 0 3]	84.8978 ms
max_div_15_3.txt	15	3	2	100	2	13.2732	[11 8]	7.88421 ms
max_div_15_3.txt	15	3	2	100	3	13.2732	[11 8]	7.33629 ms
max_div_15_3.txt	15	3	3	100	2	31.8685	[11 4 6]	20.4185 ms
max_div_15_3.txt	15	3	3	100	3	31.8685	[11 4 6]	19.722ms
max_div_15_3.txt	15	3	4	100	2	59.7638	[8 10 4 11]	50.086ms
max_div_15_3.txt	15	3	4	100	3	59.7638	[4 11 8 10]	50.3891 ms
max_div_15_3.txt	15	3	5	100	2	96.0858	[8 11 3 4 13]	108.502 ms
max_div_15_3.txt	15	3	5	100	3	96.0858	[13 3 8 11 4]	103.105 ms

max_div_20_2.txt	20	2	2	100	2	8.51033	[17 18]	38.6212 ms
max_div_20_2.txt	20	2	2	100	3	8.51033	[17 18]	14.9052 ms
max_div_20_2.txt	20	2	3	100	2	21.9961	[17 8 18]	24.7779 ms
max_div_20_2.txt	20	2	3	100	3	21.9961	[17 18 8]	23.6166 ms
max_div_20_2.txt	20	2	4	100	2	40.0023	[1 18 2 8]	49.1495 ms
max_div_20_2.txt	20	2	4	100	3	40.0023	[8 2 18 1]	52.1648 ms
max_div_20_2.txt	20	2	5	100	2	63.6517	[17 18 8 1 13]	101.987 ms
max_div_20_2.txt	20	2	5	100	3	63.6517	[13 8 18 1 17]	105.998 ms
max_div_20_3.txt	20	3	2	100	2	11.8003	[13 12]	10.5786 ms
max_div_20_3.txt	20	3	2	100	3	11.8003	[12 13]	9.78571 ms
max_div_20_3.txt	20	3	3	100	2	30.8727	[13 12 7]	25.4742 ms

max_div_20_3.txt	20	3	3	100	3	30.8727	[13 12 7]	27.9734 ms
max_div_20_3.txt	20	3	4	100	2	56.6903	[12 13 16 2]	65.1883 ms
max_div_20_3.txt	20	3	4	100	3	56.6903	[2 12 13 16]	64.176ms
max_div_20_3.txt	20	3	5	100	2	92.8298	[12 2 16 13 7]	119.965 ms
max_div_20_3.txt	20	3	5	100	3	90.8648	[16 2 12 13 8]	118.153 ms
max_div_30_2.txt	30	2	2	100	2	11.6571	[8 27]	63.0623 ms
max_div_30_2.txt	30	2	2	100	3	11.6571	[8 27]	51.0468 ms
max_div_30_2.txt	30	2	3	100	2	28.9443	[8 27 1]	44.6917 ms
max_div_30_2.txt	30	2	3	100	3	27.5943	[8 27 14]	32.8112 ms
max_div_30_2.txt	30	2	4	100	2	52.7712	[8 1 10 27]	72.825ms
max_div_30_2.txt	30	2	4	100	3	52.7712	[8 1 10 27]	69.4888 ms

max_div_30_2.txt	30	2	5	100	2	80.9102	[12 8 10 1 27]	129.38ms
max_div_30_2.txt	30	2	5	100	3	80.0675	[17 10 8 27 1]	121.755 ms
max_div_30_3.txt	30	3	2	100	2	13.0737	[16 6]	67.4642 ms
max_div_30_3.txt	30	3	2	100	3	13.0737	[6 16]	48.2569 ms
max_div_30_3.txt	30	3	3	100	2	34.2905	[5 16 23]	47.9031 ms
max_div_30_3.txt	30	3	3	100	3	34.2905	[5 16 23]	37.7643 ms
max_div_30_3.txt	30	3	4	100	2	63.702	[23 5 16 13]	87.3297 ms
max_div_30_3.txt	30	3	4	100	3	63.702	[23 5 16 13]	83.374ms
max_div_30_3.txt	30	3	5	100	2	99.592	[5 13 14 23 16]	149.611 ms
max_div_30_3.txt	30	3	5	100	3	99.5089	[6 13 23 16 14]	153.469 ms

Branch & Bound

Problema	n	K	m	Strategy	z	S	CPU
max_div_15_2.txt	15	2	4	Minimum Level	49.8268	[0 5 6 8]	876.604 ms
max_div_15_3.txt	15	3	4	Minimum Level	59.7638	[4 8 10 11]	333.978 ms
max_div_20_2.txt	20	2	4	Minimum Level	40.0023	[1 2 8 18]	1535.62 ms
max_div_20_3.txt	20	3	4	Minimum Level	56.6903	[2 12 13 16]	776.42 ms
max_div_30_2.txt	30	2	4	Minimum Level	52.7712	[1 8 10 27]	5176.96 ms
max_div_30_3.txt	30	3	4	Minimum Level	63.702	[5 13 16 23]	5937.57 ms