

Tarea 3

Metaheurísticas Poblacional para el MISP

Integrantes: Sebastián González M.

Pedro Palacios

Profesor: Pedro Pinacho D.

Metaheurística Poblacional ACO

El algoritmo propuesto para metaheurística poblacional es el de un Ant Colony Optimization. El ACO está compuesto por dos estructuras base **Ant** y **PheromoneTree**, para que funcione la estructura de **AntColony**.

Por un lado, tenemos a **PheromoneTree**, esta estructura se encarga de administrar las feromonas que poseen los nodos a través de un árbol donde sus hojas representan a los nodos del grafo y poseen el nivel de feromonas de estos, en cambio, los nodos internos y raíz poseen la acumulación de su decendencia. La administración de estas feromonas consiste en el depósito, evaporación, eliminación y búsqueda de estas a través del árbol. Al contar con la acumulación de feromonas en nodos internos, la modificación de las hojas conlleva una propagación del cambio a través de los nodos internos hasta la raíz. La evaporación igualmente posee esta propagación, con la diferencia de que se elimina un cierto porcentaje de las feromonas de cada nodo hoja, hasta un tope mínimo de 1. La búsqueda implementada consiste en una elección aleatoria de un nodo hijo con una ponderación de las feromonas del padre.

Como segunda estructura tenemos **Ant**, la cual se encarga de construir las soluciones del MISP. Esta estructura puede acceder a un árbol de feromonas general y también posee su propio árbol, para poder ir descartando los nodos que ya no puede usar. La construcción de la solución consiste en buscar un nodo a través del árbol que posee, agregarlo a la solución e invalidar el nodo y a su vecindad, repitiendo el ciclo hasta que no queden hojas por insertar. La estructura posee 2 formas de depositar feromonas en el árbol general, una es agregando una cierta cantidad a las que ya hay y la otra es cambiando el valor existente.

Por último, **AntColony**, esta estructura se encarga de la búsqueda de soluciones a través de **Ant** y posee el **PheromoneTree** general. El funcionamiento de la colonia consiste en un ciclo limitado por tiempo de ejecución, en el cual la hormiga crea una solución y esta deposita feromonas en los nodos que componen su solución, luego guarda la solución si es mejor que la mejor anterior, borra la solución de la hormiga y evapora una parte de las feromonas del árbol general dando paso a una nueva iteración.

Pseudo código

Algoritmo 1: Algoritmo Colonia de Hormigas para el MISP

Input: n : número de nodos, nl : lista de vecindades, t : tiempo máximo
(s), e : tasa de evaporación, d : cantidad de depósito

Output: $bestSize$: tamaño del mejor conjunto independiente
encontrado

$bestSize \leftarrow 0$;

$start \leftarrow tiempoActual()$;

$F[0..n-1] \leftarrow 1$;

while $tiempoActual() - start < t$ **do**

 /* Construcción de la solución */

while exista al menos un nodo u con $F[u] > 0$ **do**

$v \leftarrow$ selección aleatoria ponderada por F ;

 agregar v a sol ;

$F[v] \leftarrow 0$;

foreach u en $nl(v)$ **do**

$F[u] \leftarrow 0$;

 /* Depósito de feromonas */

foreach u en sol **do**

$F[u] \leftarrow F[u] + d * frac|sol|/bestSize$;

 /* Actualización de la mejor solución */

if $|sol| > bestSize$ **then**

$bestSize \leftarrow |sol|$;

 /* Evaporación de feromonas */

for $i \leftarrow 0$ **to** $n-1$ **do**

$F[i] \leftarrow \lfloor F[i] \cdot (1 - e) \rfloor$;

$F[i] \leftarrow \max(1, F[i])$;

return $bestSize$

Ajuste de parámetros usando IRace

Los parámetros ajustables de nuestro algoritmo Ant Colony son Depósito: “ d ” y Tasa de evaporación: “ e ”

Depósito d : Corresponde a una constante que influye en la cantidad de feromonas depositada por las hormigas al encontrar un camino válido. La cantidad real depositada se calcula según el tamaño de la solución encontrada y el tamaño de la mejor solución encontrada como: $d * \frac{currentSize}{bestSize}$

Tasa de evaporación e : Corresponde a la fracción de feromonas “evaporadas”, las feromonas “ F ” se actualizan según $F = F * (1 - e)$ para cada nodo en cada iteración.

También se experimentó con una forma alternativa de depositar las feromonas en la que se deposita la totalidad del valor d solo cuando se encuentra una mejora global. Esto se representa con la variable Deposit Type que puede tomar valor *fixed* o *proportional*. Siendo el segundo el que finalmente se utilizó.

Configuración de parámetros en IRace:

Param name	Prefix	Value type	Values/Rage value
evap_rate	-e	r	(0, 1)
deposit_amount	-d	o	(10, 20, 50, 100, 500, 1000, 5000, 10000, 100000)
deposit_type	-p	c	(proportional, fixed)

La ejecución del IRace con los parámetros anteriores entregó 3 configuraciones para ejecutar:

IP test	Evap rate	Deposit amount	Deposit type
61	0.226	100	proportional
123	0.199	100	proportional
114	0.300	500	proportional

La configuración utilizada en los experimentos a continuación será:

$e = 0.226$ $d = 100$ Con el modo de depósito proporcional.

$e = 0.226$ es una fracción significativa de las feromonas a eliminar, aunque $d = 100$ es un valor moderado, esta configuración explota de manera relativamente breve las buenas soluciones priorizando la exploración.

Resultados:

Ant Colony (60 segundos)

Resultados de ant colony promediados entre todos los archivos del dataset (30 por densidad)

Tamaño: 1000 nodos

Density	Avg_MISP_Size	Avg_Iterations
0.1	65.73	18035764
0.2	37.13	66369499
0.3	25.43	145294095
0.4	19	256618577
0.5	14.97	396534677
0.6	12.03	562712370
0.7	9.4	756366481
0.8	7.47	985674591
0.9	5.77	1260716099

Tamaño: 2000 nodos

Density	Avg_MISP_Size	Avg_Iterations
0.1	75.93	9119627
0.2	39.97	36497622
0.3	26.17	80467555
0.4	19.3	138076228
0.5	15.53	208495267
0.6	12.87	291326994
0.7	10.17	387487465
0.8	8.07	501150657
0.9	6	637091864

Tamaño: 3000 nodos

Density	Avg_MISP_Size	Avg_Iterations
0.1	80.7	5194226
0.2	38.63	20138979
0.3	26.17	43510990
0.4	19.9	74216376
0.5	16	112049640
0.6	13	157068674
0.7	10.77	209740735
0.8	8.8	272330384
0.9	6.27	347957632

Comparaciones (10 segundos)

Tablas de comparación entre Greedy, Random Greedy, ILS y Ant Colony. Los valores de estas tablas son promedios entre todos los archivos del dataset (30 por densidad)

Tamaño: 1000 nodos

Density	Greedy Time(μ s)	Greedy MISP	Rand Greedy Time(μ s)	Rand Greedy MISP	ILS MISP	Iterations ILS	AntCol MISP	Iterations AntCol
0.1	8.333804	44	0.190858	10	77.5	115487	64.1	823643
0.2	8.672743	24	0.187942	5	42.27	138495	36.37	3051368
0.3	9.06669	17	0.17537	4	28.7	107147	24.7	6635357
0.4	9.048262	12	0.170476	3	21.13	108303	18.23	11633449
0.5	8.620524	9	0.169147	2	16.63	116190	14.27	17926904
0.6	7.694334	7	0.165362	2	12.8	141169	11.7	25329589
0.7	7.628834	6	0.171762	2	9.87	138755	9.27	33936899
0.8	8.48613	4	0.200094	2	6.93	150652	7.47	43650263
0.9	7.869812	3	0.221967	2	5.03	158833	5.77	55669883

Tamaño: 2000 nodos

Density	Greedy Time(μ s)	Greedy MISP	Rand Greedy Time(μ s)	Rand Greedy MISP	ILS MISP	Iterations ILS	AntCol MISP	Iterations AntCol
0.1	19.163621	50	0.354984	10	73.8	126158	73.83	426802
0.2	19.083393	27	0.331098	5	40.33	142794	38.63	1650535
0.3	19.726066	18	0.392128	4	27.63	154869	25.03	3610627
0.4	19.996258	13	0.323833	3	20.57	167998	18.8	6147093
0.5	18.701052	10	0.312964	2	16	176165	14.97	9260288
0.6	17.489468	8	0.303793	2	11.7	217683	12.03	12940043
0.7	18.519467	6	0.377138	2	9.17	215124	10	17084542
0.8	19.596097	5	0.428548	2	6.7	262662	8	21797926
0.9	14.964753	3	0.386123	2	5	355561	6	27829421

Tamaño: 3000 nodos

Density	Greedy Time(μ s)	Greedy MISP	Rand Greedy Time(μ s)	Rand Greedy MISP	ILS MISP	Iterations ILS	AntCol MISP	Iterations AntCol
0.1	37.216061	54	0.384533	10	67.1	247166	78.07	249405
0.2	57.096374	29	0.35966	5	37.33	289771	37.33	921320
0.3	34.193847	19	0.367701	4	25.43	312097	25.27	1956868
0.4	24.310741	14	0.375784	3	19	344132	19.2	3313223
0.5	33.781044	11	0.371442	2	14.9	359601	15.2	4934405
0.6	25.986004	8	0.369517	2	10.87	432237	12.27	6875701
0.7	69.076707	7	0.458698	2	8.2	427339	10.07	9129420
0.8	21.097321	5	0.516198	2	6.17	514455	8.13	11759180
0.9	21.838387	4	0.611401	2	4.7	697336	6.13	15397822

Gráfico 1000 nodos

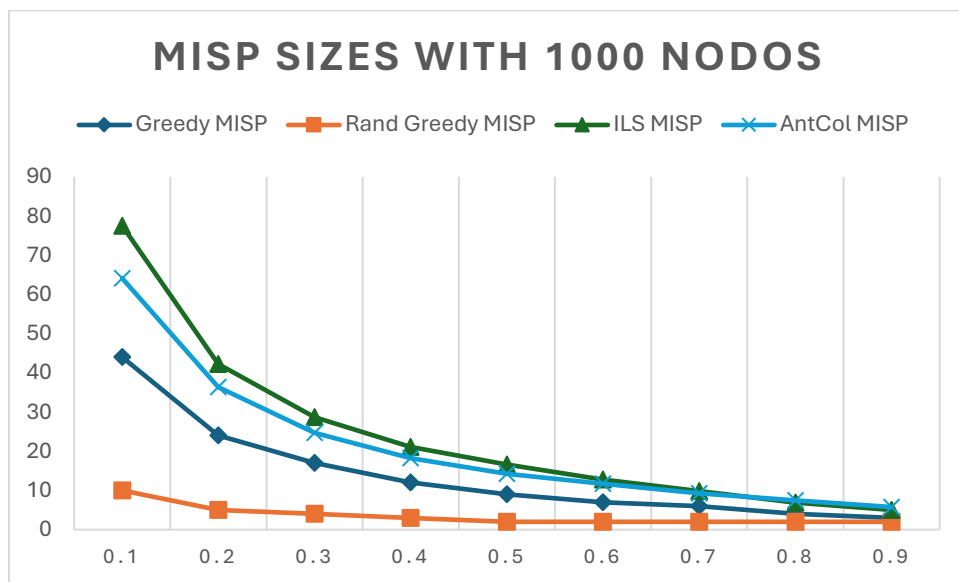


Gráfico 2000 nodos

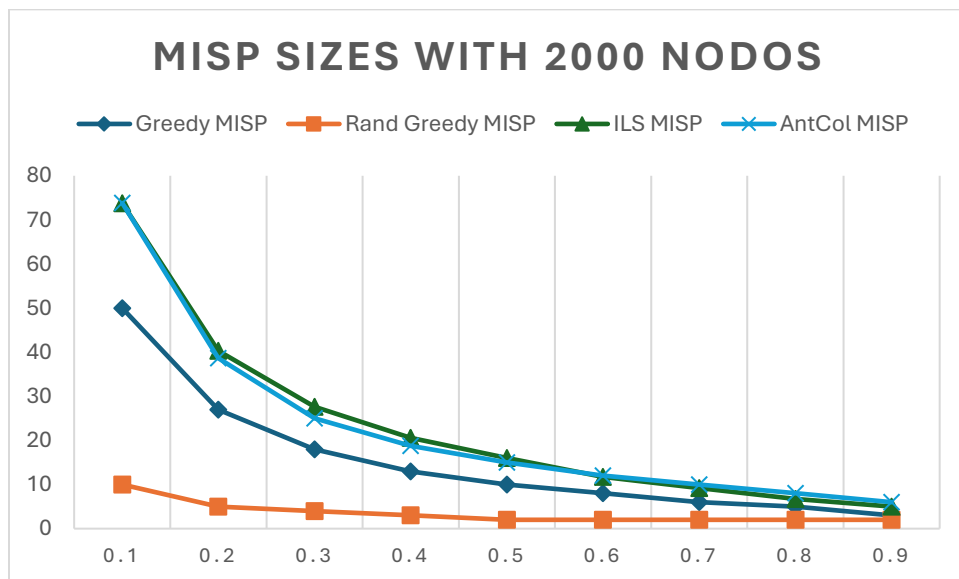


Gráfico 3000 nodos

