

Iterated Greedy para la resolución del problema de la máxima intersección de k-conjuntos

XIV Congreso Español de Metaheurísticas, Algoritmos Evolutivos y Bioinspirados

Alejandra Casado-Ceballos
Sergio Pérez-Peló
Jesús Sánchez-Oro
Abraham Duarte



Contenido

1 Introducción

- Definición formal del problema
- Ejemplo
- Estado del arte

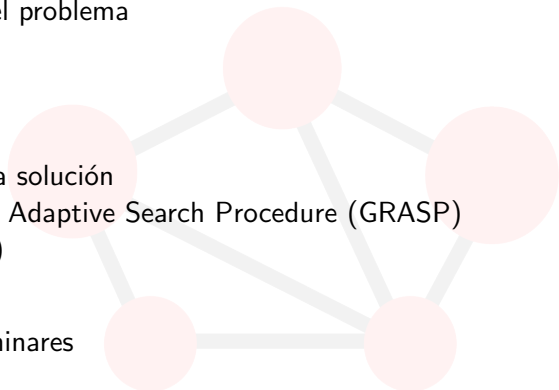
2 Desarrollo

- Representación de la solución
- Greedy Randomized Adaptive Search Procedure (GRASP)
- Iterated Greedy (IG)

3 Resultados

- Experimentos preliminares
- Experimento final

4 Conclusiones



Introducción

- El **problema** abordado pertenece a la familia de problemas de la **selección de individuos**.
- **Maximizar** el número de **características en común** de un subconjunto de elementos de un tamaño dado.
- Aplicaciones:
 - Asegurar la **privacidad**.
 - Seleccionar **músicos** para un festival.

Introducción

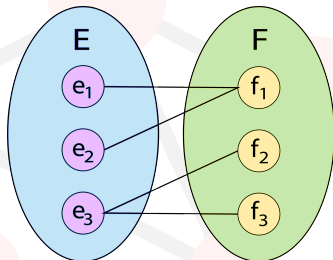
Maximización de la intersección entre k elementos (kMIS)

Los datos de **entrada** son:

- $E = \{e_1, e_2, \dots, e_n\}$
- $F = \{F_{e_1}, F_{e_2}, \dots, F_{e_n}\}$
- $k \in \mathbb{N}$, con $k < n$

Una **solución** S se modela como:

- $S \subset E$
- $|S| = k$



Introducción

Definición formal del problema

Una solución para el k MIS se **evalúa** como el **número de características** que tienen en **común** los elementos seleccionados:

$$k\text{MIS}(S) = \left| \bigcap_{e_i \in S} F_{e_i} \right|$$

El k MIS tiene como propósito encontrar la solución S^* que **maximiza** el valor de la **función objetivo**:

$$S^* \leftarrow \arg \max_{S \in \mathcal{S}} k\text{MIS}(S)$$

Introducción

Ejemplo

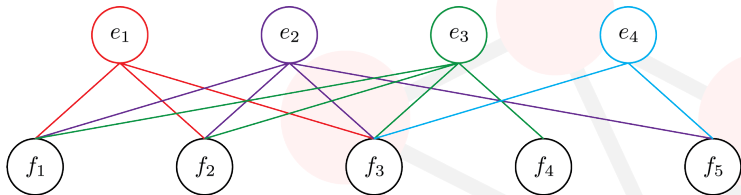


Figura 1: Ejemplo de instancia con 4 elementos y 5 características, donde se deben elegir 3 elementos.

Introducción

Ejemplo

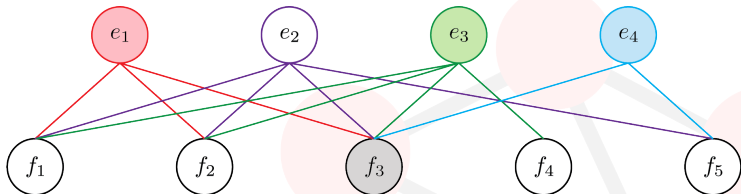


Figura 2: Solución $S_1 = \{e_1, e_3, e_4\}$

$$k\text{MIS}(S_1) = |F_{e_1} \cap F_{e_3} \cap F_{e_4}| = 1$$

Introducción

Ejemplo

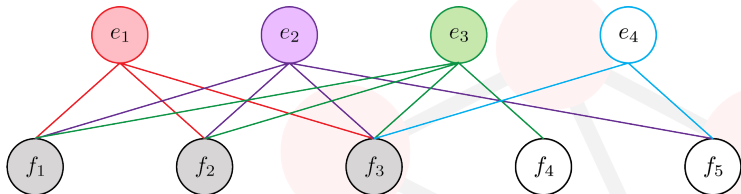
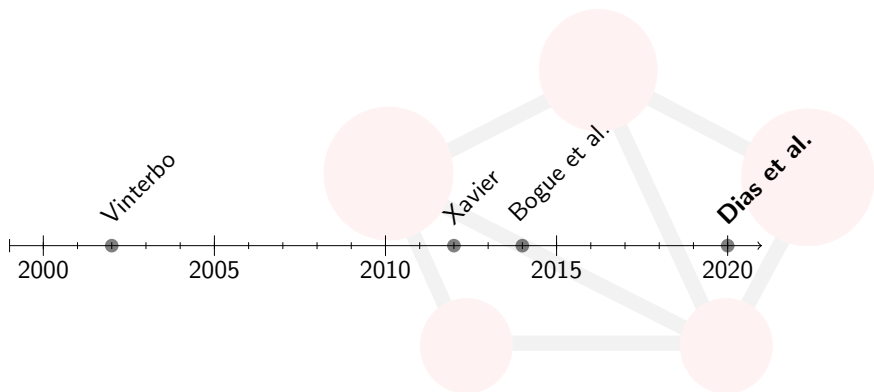


Figura 3: Solución $S_2 = \{e_1, e_2, e_3\}$

$$kMIS(S_2) = |F_{e_1} \cap F_{e_2} \cap F_{e_3}| = 3$$

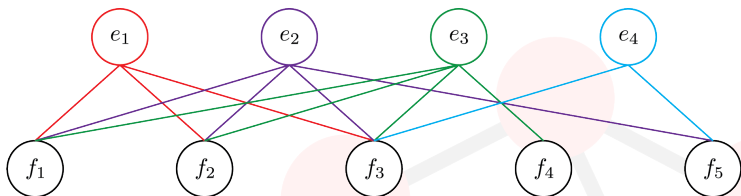
Introducción

Estado del arte



Desarrollo

Representación de la solución

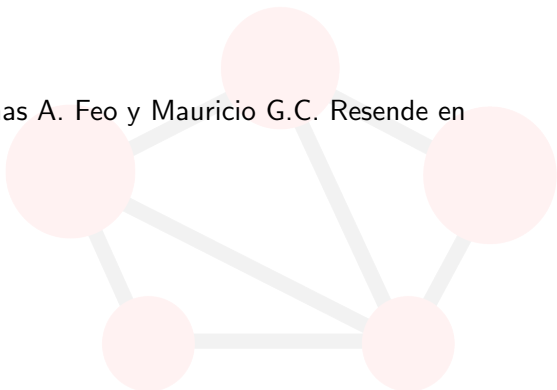


	S_1					S_2				
	f_1	f_2	f_3	f_4	f_5	f_1	f_2	f_3	f_4	f_5
e_1	1	1	1	0	0	1	1	1	0	0
e_2	1	1	1	0	1	1	1	1	0	0
e_3	1	1	1	1	0	1	1	1	0	0
e_4	0	0	1	0	1	0	0	1	0	1
AND	0	0	1	0	0	1	1	1	0	0

Desarrollo

Greedy Randomized Adaptive Search Procedure (GRASP)

- Propuesto por Thomas A. Feo y Mauricio G.C. Resende en 1989.
- Consta de dos fases:
 - 1 **Construcción**
 - 2 **Mejora**



Desarrollo

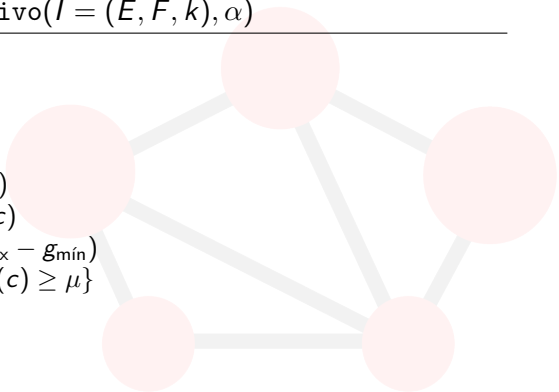
GRASP-Construcción

Algoritmo 1 Constructivo($I = (E, F, k), \alpha$)

```

1:  $e \leftarrow \text{Random}(E)$ 
2:  $S \leftarrow \{e\}$ 
3:  $CL \leftarrow E \setminus \{e\}$ 
4: while  $|S| < k$  do
5:    $g_{\min} \leftarrow \min_{c \in CL} g(c)$ 
6:    $g_{\max} \leftarrow \max_{c \in CL} g(c)$ 
7:    $\mu \leftarrow g_{\max} - \alpha \cdot (g_{\max} - g_{\min})$ 
8:    $RCL \leftarrow \{c \in CL : g(c) \geq \mu\}$ 
9:    $e \leftarrow \text{Random}(RCL)$ 
10:   $S \leftarrow S \cup \{e\}$ 
11:   $CL \leftarrow CL \setminus \{e\}$ 
12: end while
13: return  $S$ 

```



Desarrollo

GRASP-Mejora

- **Movimiento:** intercambio.

$$\text{Intercambio}(S, e_i, e_j) = (S \setminus e_i) \cup e_j$$

- **Vecindad:** soluciones a las que se puede llegar con un único intercambio.

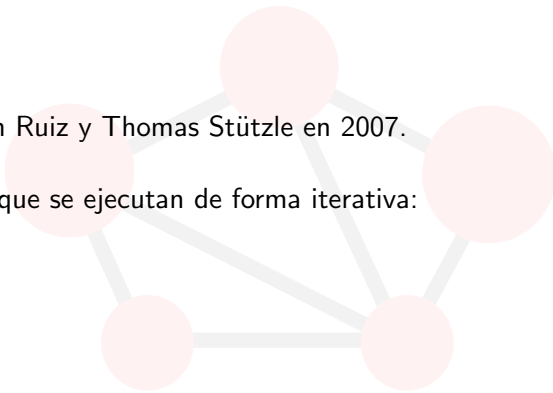
$$N(S) = \{S' \leftarrow \text{Intercambio}(S, e_i, e_j), \forall e_i \in S \wedge \forall e_j \in (E \setminus S)\}$$

- **Recorrido:** *first improvement*.
- Búsqueda local **eficiente:** no se realiza el movimiento si no mejora.

Desarrollo

Iterated Greedy

- Propuesto por Rubén Ruiz y Thomas Stützle en 2007.
- Consta de dos fases que se ejecutan de forma iterativa:
 - 1 **Destrucción**
 - 2 **Reconstrucción**



Desarrollo

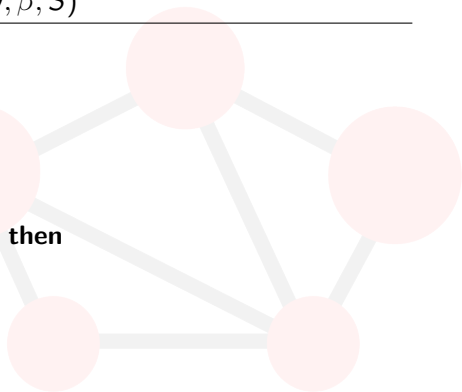
Iterated Greedy

Algoritmo 2 IteratedGreedy(θ, β, S)

```

1:  $S_{mejor} \leftarrow S$ 
2:  $i = 0$ 
3: while  $i < \theta$  do
4:    $S' \leftarrow \text{Destrucción}(S_{mejor}, \beta)$ 
5:    $S'' \leftarrow \text{Reconstrucción}(S')$ 
6:    $S''' \leftarrow \text{BúsquedaLocal}(S'')$ 
7:   if  $kMIS(S''') > kMIS(S_{mejor})$  then
8:      $S_{mejor} \leftarrow S'''$ 
9:      $i = 0$ 
10:  else
11:     $i = i + 1$ 
12:  end if
13: end while
14: return  $S_{mejor}$ 

```



Resultados

- Lenguaje de programación: **Java 11**.
- Características máquina de experimentación: AMD EPYC 7282 (2.8 GHz), 8 GB RAM.
- Total de instancias: 238.
- Conjunto preliminar de instancias: 27.
- Métricas utilizadas:
 - **Promedio**: valor promedio de la función objetivo.
 - **Tiempo (s)**: tiempo de ejecución medido en segundos.
 - **Desv. (%)**: desviación porcentual media respecto a al mejor solución del experimento.
 - **# Mejores**: número de mejores soluciones encontradas en el experimento.

Resultados

Experimentos preliminares

Algoritmo	Promedio	Tiempo (s)	Desv. %	# Mejores
GRASP(0.25)	24.26	14.66	5.31	18
GRASP(0.50)	22.37	17.26	25.97	11
GRASP(0.75)	21.63	17.80	28.94	10
GRASP(RND)	25.07	15.99	0.66	25

Tabla 1: Comparativa de los diferentes valores de α considerados construyendo y mejorando 1000 soluciones independientes.

Resultados

Experimentos preliminares - Constructivo y búsqueda local

Algoritmo	Promedio	Tiempo (s)	Desv. (%)	#Mejores
GRASP(<i>RND</i>)	25.07	15.99	0.44	24
GRASP(<i>RND</i>) ++¹	25.11	1.20	0.51	25

Tabla 2: Comparativa de la búsqueda local frente a la eficiente.

¹++ se usa para diferenciar GRASP cuya búsqueda local se trata de la aproximación y GRASP que tiene como búsqueda local la eficiente.

Resultados

Experimentos preliminares - Parámetros Iterated Greedy

- $\beta = 0.2$ (Porcentaje de la solución que es destruida).
- Destrucción: aleatoria.
- Reconstrucción: voraz.

θ	Promedio	Tiempo (s)	Desv. (%)	#Mejores
5	25.15	1.24	0.13	26
10	25.19	1.20	0.00	27
15	25.19	1.21	0.00	27
20	25.19	1.22	0.00	27

Tabla 3: Comparativa del efecto de los distintos valores de θ^2 en IG_{RG} cuando $\beta = 0.2$.

² θ : número de iteraciones sin mejora que se permiten en Iterated Greedy.

Resultados

Experimento final

Algoritmo	Promedio	Tiempo (s)	Desv. (%)	# Mejores
<i>Reactive VNS</i>	32.59	21.01	2.04	186
GRASP+IG	33.00	0.85	0.57	225

Tabla 4: Comparativa entre el algoritmo propuesto (*GRASP+IG*) y el estado del arte (*Reactive VNS*) sobre el total de 238 instancias.

Conclusiones



El algoritmo propuesto obtiene resultados competitivos con el **estado del arte**.



Se ha enviado un **artículo** a la revista Journal of Heuristics (actualmente en primera revisión).



Trabajo futuro: probar la representación de la solución propuesta para otros problemas de selección de características.

Iterated Greedy para la resolución del problema de la máxima intersección de k-conjuntos

XIV Congreso Español de Metaheurísticas, Algoritmos Evolutivos y Bioinspirados

Alejandra Casado-Ceballos
Sergio Pérez-Peló
Jesús Sánchez-Oro
Abraham Duarte

