



**GRADO EN INGENIERÍA INFORMÁTICA DE GESTIÓN Y SISTEMAS DE INFORMACIÓN**

TRABAJO FIN DE GRADO

2015 / 2016

**EXTENSIÓN DE UN PAQUETE DE ALGORITMOS METAHEURÍSTICOS EN R PARA LA DOCENCIA**

**RESUMEN**

DATOS DE LA ALUMNA O DEL ALUMNO		DATOS CODIRECCIÓN	
NOMBRE ANDER	NOMBRE JOSU	NOMBRE BORJA	
APELLIDOS CARREÑO LÓPEZ	APELLIDOS CEBERIO URIBE	APELLIDOS CALVO MOLINOS	
	DEPARTAMENTO LSI	DEPARTAMENTO CCIA	
			
FDO.:	FDO.:	FDO.:	
FECHA: 17-06-2016	FECHA: 17-06-2016	FECHA: 17-06-2016	

## 1. Introducción

En los últimos años se han realizado numerosos avances en el campo de la optimización combinatoria. Los algoritmos evolutivos han demostrado ser una herramienta muy valiosa para resolver problemas de optimización costosos. En concreto, los Algoritmos de Estimación de Distribuciones han atraído la atención de los investigadores y se han hecho numerosos avances al respecto. Sin embargo, no existe software que acerque estos avances a los alumnos. El objetivo de este Trabajo de Fin de Grado es extender la sección de algoritmos metaheurísticos del paquete *metaheuR*, en concreto, los algoritmos de estimación de distribuciones. De esta manera, los estudiantes podrán comparar, combinar y aprender estos algoritmos resolviendo los problemas de optimización clásicos junto con el libro *Bilaketa Heuristikoak* aún en desarrollo. Bajo el lenguaje de programación R, este proyecto recoge 3 Algoritmos de Estimación de Distribuciones, el *Estimation of Bayesian Network Algorithm* [Etxeberria y Larrañaga, 1999] que aprende una red Bayesiana como modelo probabilístico; el *Edge Histogram Based Sampling Algorithm* [Tsutsui, 2002] que aprende probabilidades marginales de orden dos y el *Plackett-Luce Estimation of Distribution Algorithm* [Ceberio y col., 2013] que aprende un vector de pesos de la población. Además, se han realizado experimentos que verifican el correcto funcionamiento de los modelos implementados así como ejecuciones sobre problemas de optimización tales como el *Travelling Salesman Problem* o el *Maximum Independence Set*. Con estos resultados, también se han realizado comparativas y análisis de los resultados que dan una visión de la eficiencia de cada uno de los algoritmos.

## 2. Algoritmos de Estimación de Distribuciones

Los Algoritmos de Estimación de Distribuciones cuyo acrónimo en inglés es *Estimation of Distribution Algorithms* (EDA) [Larrañaga y Lozano, 2002; Lozano, 2006; Pelikan, Goldberg y col., 2000; Pelikan, Sastry y col., 2007] pertenecen a la familia de algoritmos evolutivos (EAs). La principal característica de los EAs es que utilizan técnicas inspiradas en la evolución natural de las especies. En la naturaleza, las especies cambian a lo largo del tiempo adaptándose a nuevas condiciones y entornos. Esta evolución hace que sobreviva el individuo que mejor se haya adaptado. Esta misma idea se traslada al campo de la optimización donde un individuo representa una solución para un problema; una población es un conjunto de soluciones (individuos), y operaciones como cruce, mutación y selección son utilizadas para que los individuos (soluciones) evolucionen (mejoren). El paradigma más conocido de los EAs son los Algoritmos Genéticos (GAs) [Goldberg, 2000].

La principal diferencia de los GAs y los EDAs es la creación de nuevas soluciones. Los GAs crean nuevas soluciones a partir del cruce y la mutación

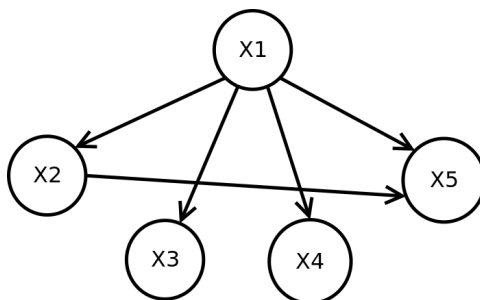


Figura 1: Ejemplo de red Bayesiana.

de individuos mientras que los EDAs aprenden un modelo probabilístico que se ajusta a los datos y después lo muestrean obteniendo así los nuevos individuos. Esta característica de los EDAs hace que estas etapas mencionadas sean las que determinen el funcionamiento, la complejidad y la eficiencia del algoritmo. A continuación se explican brevemente los 3 algoritmos implementados.

## 2.1. Estimation of Bayesian Network Algorithm

El algoritmo de *Estimation of Bayesian Network Algorithm* (EBNA) es un EDA que a cada iteración, a partir de la población de individuos aprende una red Bayesiana. Después, se generan las nuevas soluciones muestreando la red. Una red Bayesiana es un modelo gráfico probabilístico en el cual los nodos representan las variables y los arcos las dependencias entre sí (ver Figura 1).

La búsqueda de la estructura que mejor se ajusta a los datos es una tarea *np-hard* [Chickering y col., 1994], por tanto, se utilizan algoritmos de búsqueda local. El aprendizaje de la red Bayesiana se ha realizado utilizando, el paquete *bnLearn* [Scutari, 2009] mediante la función *hc*, esta función hace uso del algoritmo *Hill Climbing* y para valorar la estructura que se aprende a cada paso, implementa la medida *Bayesian Information Criterion* (BIC).

En cuanto al muestreo de la red, también se ha realizado utilizando el paquete *bnLearn* mediante la función *rbn* que muestrea los elementos siguiendo el orden ancestral de la red, es decir, los padres deben ser muestreados antes que los hijos. Para ello, se utiliza el algoritmo *Probabilistic Logical Sampling*.

## 2.2. Edge Histogram Based Sampling Algorithm

El *Edge Histogram Based Sampling Algorithm* (EHBSA) es un EDA que se caracteriza por aprender una matriz de probabilidades de orden dos (adyacencias) de la población [Tsutsui, 2002; Tsutsui y col., 2003]. La población está formada por individuos que pertenecen al dominio de permutaciones. Después, se muestrean nuevas soluciones teniendo en cuenta dicha matriz.

Cabe decir que este tipo de EDA se propuso inicialmente para problemas definidos sobre el dominio de permutaciones, sin embargo, es perfectamente aplicable a cualquier otro problema de optimización combinatoria.

El aprendizaje del modelo probabilístico se realiza representando el ratio de apariciones en una matriz. El primer elemento del nuevo individuo se obtiene de manera aleatoria. Los demás son obtenidos muestreando la matriz de adyacencias teniendo en cuenta el elemento seleccionado en el paso anterior. La elección se realiza mediante el muestreo de la fila correspondiente al elemento previo. Después, se anula la columna del elemento muestreado.

Para el muestreo de esta distribución de probabilidad se utiliza el método de la ruleta sesgada (*roulette-wheel*). Como se observa en el Algoritmo 1, se inicializa la primera posición del nuevo individuo con un valor aleatorio entre 0 y el número total de características. Las siguientes posiciones se eligen seleccionando el índice del individuo tal que  $\alpha \geq \sum_{i=0}^N E_{i,j}$  donde  $\alpha$  corresponde a un valor aleatorio entre 0 y el valor máximo de la fila de la matriz  $E_{i,j}$

---

**Algorithm 1:** Pseudocódigo general para muestrear el modelo EHB-SA.

---

```

1  $p \leftarrow 0$ .
2  $\pi[0] \leftarrow \text{random}(0, N - 1)$ ; // Obtener el primer elemento de forma
   aleatoria.
3 while  $p < N - 1$  do
4    $rw[j] \leftarrow e_{s[p],j}^t (j = 0, 1, \dots, N - 1)$ 
5    $rw[(rw[c[i]] \leftarrow 0 \text{ for } (i = 0, \dots, p)$ 
6    $\pi[p + 1] \leftarrow rw[x] / \sum_{j=0}^{N-1} rw[j]$ 
7    $p \leftarrow p + 1$ 
8 end
9 return  $s$ 

```

---

### 2.3. Plackett-Luce Estimation of Distribution Algorithm

El algoritmo *Plackett-Luce Estimation of Distribution Algorithm* (PLE-DA) fue propuesto por [Ceberio y col., 2013] y fue diseñado para resolver problemas en el dominio de las permutaciones. Este algoritmo incorpora el modelo *Plackett-Luce* que es una extensión del modelo *Bradley Terry*. En este modelo, los parámetros de cada elemento se especifican por un vector de parámetros  $\mathbf{w} = (w_1, w_2, \dots, w_n)$ .

Formalmente, para cualquier elemento  $i \in B$ , siendo  $B$  el rango de todos los posibles conjuntos no nulos de  $\{1, \dots, n\}$ ,  $P_B(i)$  es la probabilidad de que el elemento  $i$  sea elegido antes que los demás elementos posibles en  $B$  y se formaliza mediante la siguiente ecuación:

$$P_B(i) = \frac{w_i}{\sum_{j \in B} w_j} \quad (1)$$

El aprendizaje del modelo Plackett-Luce consiste en aprender los pesos  $w$  de la población. Para ello, se ha implementado el algoritmo *Minorization Maximization* siguiendo las indicaciones de [Ceberio y col., 2013; Hunter, 2004]. Como resultado, se obtiene un vector de pesos que representa las proporciones de las características de los individuos.

El muestreo del algoritmo se realiza siguiendo el método de la ruleta sesgada comentado en la Sección 2.2.

### 3. Experimentación

Los experimentos realizados a lo largo del proyecto se pueden dividir en dos bloques, siendo el primero de ellos una verificación del correcto funcionamiento de las funciones de muestreo y aprendizaje de los modelos y el cálculo de probabilidad de una población de individuos dado el modelo. En segundo lugar, se han realizado ejecuciones sobre problemas de optimización como el *Travelling Salesman Problem* (TSP) y el *Maximum Independence Set* (MIS). Estos dos bloques de experimentos son los correspondientes a los casos de uso que un usuario cualquiera tiene ante el desarrollo realizado en este TFG.

Las ejecuciones sobre problemas de optimización, en el caso del MIS se ha realizado sobre 5 grafos diferentes creados de forma aleatoria. Cada grafo, se ha resuelto 5 veces para obtener después una mediana de los resultados. La cantidad de variables de la solución ha sido proporcional al tamaño de la población, y el número de nodos ha ido incrementando de 20 a 100 en intervalos de 20. En el caso del TSP, las instancias se han obtenido de *TSPLib*<sup>1</sup>, las instancias también se han resuelto 5 veces cada uno y han seguido una relación lineal entre el número de ciudades y el tamaño de la población, además, como en el apartado anterior, los resultados se han obtenido haciendo una mediana de 5 ejecuciones.

Los resultados obtenidos se miden teniendo en cuenta el error normalizado frente al valor óptimo encontrado. Por tanto, según las pruebas realizadas, se puede observar que el algoritmo que mejores resultados obtiene es el UMDA [Pelikan y Mühlenbein, 1998] en el caso de instancias del MIS y el EHBSA en el caso del TSP. Estos resultados se pueden ver en la Tabla 1.

---

<sup>1</sup>TSPLib es una repositorio de problemas TSP de la cual se pueden obtener tanto las matrices de coste que definen el problema como la mejor solución encontrada hasta la fecha. Enlace: [comopt.ifi.uni-heidelberg.de/software/TSPLIB95/](http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/)

Instancia	Mejor	UMDA	EBNA	Instancia	Mejor	EHBSA	PLEDA
1	14,300	<b>0,85</b>	7,3	burma14	3323	<b>0,000</b>	0,016
2	13,725	<b>0,962</b>	2,9	ulysses22	7013	<b>0,045</b>	0,176
3	16,717	<b>0,985</b>	8,217	ftv33	1335	<b>0,551</b>	1,021
4	15,813	<b>0,988</b>	2,438	dantzig42	699	1,262	<b>0,980</b>
5	17,870	<b>0,49</b>	5,93	ft53	15186	<b>0,042</b>	0,299

(a) Resultados UMDA y EBNA sobre instancias del MIS.

(b) Resultados del EHBSA y PLEDA sobre instancias del TSP.

Tabla 1: Resultados de las Ejecuciones

## Referencias

Ceberio, Josu., Mendiburu, Alexander y Jose A Lozano

2013 “The Plackett-Luce ranking model on permutation-based optimization problems”, en *Evolutionary Computation (CEC), 2013 IEEE Congress on*, IEEE, págs. 494-501.

Chickering, David M, Dan Geiger, David Heckerman y col.

1994 *Learning Bayesian networks is NP-hard*, inf. téc., Citeseer.

Ettxeberria, Ramon y Pedro Larrañaga

1999 “Global optimization using Bayesian networks”, en *Second Symposium on Artificial Intelligence (CIMAF-99)*, Habana, Cuba, págs. 332-339.

Goldberg, David E

2000 *Genetic Algorithms—In Search, Optimization & Machine Learning, Revised ed.*

Hunter, David R

2004 “MM algorithms for generalized Bradley-Terry models”, *Annals of Statistics*, págs. 384-406.

Larrañaga, Pedro y Jose A Lozano

2002 *Estimation of distribution algorithms: A new tool for evolutionary computation*, Springer Science & Business Media, vol. 2.

Lozano, Jose A

2006 *Towards a new evolutionary computation: advances on estimation of distribution algorithms*, Springer Science & Business Media, vol. 192.

Pelikan, Martin, David E Goldberg y E Cantú-Paz

2000 “Hierarchical Problem Solving and the Bayesian Optimization Algorithm.”, en *GECCO*, págs. 267-274.

- Pelikan, Martin y Heinz Mühlenbein  
1998 “Marginal distributions in evolutionary algorithms”, en *Proceedings of the International Conference on Genetic Algorithms Mendel*, Citeseer, vol. 98, págs. 90-95.
- Pelikan, Martin, Kumara Sastry y Erick Cantú-Paz  
2007 *Scalable optimization via probabilistic modeling: From algorithms to applications*, Springer, vol. 33.
- Scutari, Marco  
2009 “Learning Bayesian networks with the bnlearn R package”, *arXiv preprint arXiv:0908.3817*.
- Tsutsui, Shigeyoshi  
2002 “Probabilistic model-building genetic algorithms in permutation representation domain using edge histogram”, en *Parallel Problem Solving from Nature—PPSN VII*, Springer, págs. 224-233.
- Tsutsui, Shigeyoshi, Martin Pelikan y David E Goldberg  
2003 “Using edge histogram models to solve permutation problems with probabilistic model-building genetic algorithms”, *IlliGAL Report*, 2003022.