

UNIVERSIDAD COMPLUTENSE DE MADRID  
**FACULTAD DE CIENCIAS MATEMÁTICAS**

DEPARTAMENTO DE SISTEMAS INFORMÁTICOS Y  
COMPUTACIÓN



**TRABAJO DE FIN DE GRADO**

**Métodos de Simulación de  
Monte-Carlo para la Optimización  
Heurística**

Presentado por: Sergio González Montero

Dirigido por: Rafael del Vado Vírveda

Grado en Matemáticas

Curso académico 2024-25

## Agradecimientos

Agradecimientos

## Resumen

Resumen

***Palabras clave:***

## Abstract

Resumen en inglés

***Keywords:***

# Índice general

<b>1. Introducción</b>	<b>6</b>
1.1. Motivación y objetivos . . . . .	6
1.2. Contexto y antecedentes del trabajo . . . . .	6
1.2.1. Problemas de optimización . . . . .	6
1.2.2. Algoritmos heurísticos . . . . .	6
1.2.3. Métodos de optimización de Monte-Carlo . . . . .	6
1.3. Estructura de la memoria . . . . .	6
<b>2. Fundamentos Teóricos</b>	<b>8</b>
2.1. Fundamentos Estadísticos . . . . .	8
2.1.1. Espacios de probabilidad y variables aleatorias . . . . .	8
2.1.2. Distribuciones de probabilidad relevantes . . . . .	9
2.1.2.1. Conceptos de estadística inferencial . . . . .	10
2.1.2.2. Muestreo y convergencia . . . . .	10
2.1.2.3. Métodos de simulación y aproximación . . . . .	11
2.2. Optimización Heurística . . . . .	12
2.2.1. Problemas de optimización . . . . .	12
2.3. Algoritmos Heurísticos . . . . .	15
2.3.1. Metaheurísticas de búsqueda . . . . .	16
2.3.2. Metaheurísticas evolutivas . . . . .	17
<b>3. Métodos de Simulación de Monte-Carlo</b>	<b>18</b>

3.1.	Integración de Monte-Carlo . . . . .	18
3.2.	Simulación estocástica . . . . .	19
3.3.	Técnicas de generación de muestras . . . . .	19
3.3.1.	Muestreo por rechazo . . . . .	19
3.3.2.	Muestreo enfatizado . . . . .	20
3.3.3.	Remuestreo por pesos . . . . .	21
3.4.	Métodos de simulación por cadenas de Markov . . . . .	23
3.4.1.	Cadena general para los métodos de Metrópolis-Hastings y Gibbs .	24
3.4.2.	Algoritmos de Metropolis-Hastings . . . . .	25
3.4.3.	Método de Gibbs . . . . .	25
3.5.	Métodos secuenciales de Monte-Carlo . . . . .	26
3.5.1.	Filtros de partículas . . . . .	27
3.6.	Métodos de simulación generales . . . . .	30
<b>4.</b>	<b>Técnicas de simulación estocástica en algoritmos de optimización heurística</b>	<b>32</b>
4.1.	Métodos de optimización de Monte-Carlo . . . . .	32
4.2.	Algoritmos genéticos . . . . .	36
4.2.0.1.	Tipos de aprendizaje . . . . .	36
4.3.	Caso práctico: problema de rutas de vehículos . . . . .	36
<b>5.</b>	<b>Conclusiones y Trabajo Futuro</b>	<b>37</b>
5.1.	Contribuciones . . . . .	37
5.2.	Conclusiones . . . . .	37
5.3.	Trabajos relacionados . . . . .	37
5.4.	Trabajo futuro . . . . .	37

# Capítulo 1

## Introducción

Este primer capítulo se centra en presentar el marco contextual en el que se desarrolla este trabajo. Por un lado,

### 1.1. Motivación y objetivos

### 1.2. Contexto y antecedentes del trabajo

[8903465]

#### 1.2.1. Problemas de optimización

#### 1.2.2. Algoritmos heurísticos

#### 1.2.3. Métodos de optimización de Monte-Carlo

### 1.3. Estructura de la memoria

La presente memoria se estructura del modo siguiente:

- En el presente **Capítulo 1** hemos presentado el tema de este trabajo, a través de
- El **Capítulo 2** se centra en
- En el **Capítulo 3** hablamos sobre
- El **Capítulo 4** se centra en
- Por último, en el **Capítulo 5** se presentan las conclusiones finales, el trabajo futuro y algunos trabajos publicados relacionados con la investigación realizada.

A lo largo de esta memoria se presentan diferentes cuadros de código que sirven para

apoyar las experimentaciones realizadas y las explicaciones textuales. Sin embargo, la totalidad del código implementado puede encontrarse en el siguiente repositorio de Github <https://github.com/mavice07/TFG-Mates.git>.

Ejemplo de cita para la bibliografía [**Sanchez2022**]

# Capítulo 2

## Fundamentos Teóricos

En este capítulo nos centraremos en

### 2.1. Fundamentos Estadísticos

#### 2.1.1. Espacios de probabilidad y variables aleatorias

Dado un experimento, recibirá el nombre de *espacio muestral*,  $\Omega$ , el conjunto de todos los posibles resultados de dicho experimento. Cada uno de esos posibles resultados se denomina *suceso elemental*, mientras que un subconjunto de más de un elemento y sus uniones se llaman *sucesos compuestos*. El espacio muestral puede ser de tipo *finito* (lanzamiento de un dado), de tipo *infinito numerable* (lanzamiento de una moneda hasta obtener la primera cruz) o de tipo *continuo* (posición de una partícula en desplazamiento en el plano) [1].

Sea  $\mathcal{A}$  una colección no vacía de subconjuntos del espacio muestral  $\Omega$ . Cada elemento  $A$  de  $\mathcal{A}$  se trata de un *suceso*, con su correspondiente probabilidad. Por ejemplo, “salir par en el lanzamiento de un dado”, con probabilidad  $\frac{1}{2}$ . Puede ser que sea un suceso *imposible*,  $\emptyset$ , como sacar un 7 en el lanzamiento de un dado, o un suceso *incompatible* (intersección nula entre los sucesos), como sacar par e impar a la vez. Denotando las *partes* de  $\Omega$  como  $\mathcal{P}(\Omega)$ , todos los subconjuntos posibles del espacio muestral, se tiene que  $\mathcal{A} \subseteq \mathcal{P}(\Omega)$ . Por lo general, se exigirá tener una estructura de  $\sigma$ -álgebra. Así,  $\mathcal{A}$  tiene dicha estructura si y sólo si:

1.  $\Omega \in \mathcal{A}$ .
2.  $\forall A \in \mathcal{A}$ , se tiene que  $A^c \in \mathcal{A}$ .
3. Para toda colección numerable  $\{A_n\}_{n \in \mathbb{N}} \subset \mathcal{A}$ , se tiene que  $\bigcup_{n=1}^{\infty} A_n \in \mathcal{A}$ .

Sea la terna  $(\Omega, \mathcal{A}, P)$ , de espacio muestral  $\Omega$ ,  $\sigma$ -álgebra  $\mathcal{A}$  sobre  $\Omega$ , y una medida de probabilidad  $P$  sobre  $\mathcal{A}$ , un *espacio probabilístico*. Sea la dupla  $(\mathbb{R}, \mathbb{B}(\mathbb{R}))$ , con  $\mathbb{R}$  la recta



real y  $\mathbb{B}(\mathbb{R})$  la  $\sigma$ -álgebra de Borel, generada por la clase de todos los intervalos de  $\mathbb{R}$  (si  $\mathcal{C} = \{(a, b)\}$ , se tiene que  $(a, b) = \bigcup_{n=1}^{\infty} \left(a, b - \frac{1}{n}\right]$ ), un espacio probabilizable o espacio medible. Sea la función:

$$\begin{aligned} X : \Omega &\rightarrow \mathbb{R} \\ w &\mapsto X(w) \end{aligned}$$

con  $w \in A$ . Dicha función será *variable aleatoria* si y sólo si  $X^{-1}(w) \in \mathcal{A}, \forall B \in \mathbb{B}(\mathbb{R})$ .

Una *medida de probabilidad*  $P$  se caracteriza mediante una función, llamada *función de distribución*, tal que  $F : \mathbb{R} \rightarrow [0, 1]$ , cumpliendo que sea monótona no decreciente, es decir,  $F(x_1) \leq F(x_2)$ , si  $x_1 < x_2$ , que sea continua por la derecha, esto es,  $\forall x \in \mathbb{R}, F(x) = \lim_{y \rightarrow x} F(y)$  y, además, se verifica que  $\lim_{x \rightarrow -\infty} F(x) = 0$  y  $\lim_{x \rightarrow \infty} F(x) = 1$ .

### 2.1.2. Distribuciones de probabilidad relevantes

Dentro de las numerosas *distribuciones de probabilidad* existentes, se definirán las siguientes:

**Distribución uniforme:** una variable aleatoria  $X$  se dice uniforme en  $(a, b)$  ( $X \sim U(a, b)$ ) si su función de densidad es de la forma

$$f(x) = \frac{1}{b-a} I_{(a,b)}(x)$$

Su uso es frecuente para la generación de números *pseudoaleatorios*, debido a su sencillez computacional. Se trata de una distribución de tipo continuo.

**Distribución normal** (también conocida como *gaussiana*): una variable aleatoria  $X$  sigue esta distribución, con media  $\mu$  y varianza  $\sigma^2$  ( $-\infty < \mu < \infty; \sigma > 0; X \sim \mathcal{N}(\mu, \sigma)$ ) si  $X$  es continua con función de densidad

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2} \quad \text{para } -\infty < x < \infty$$

Es una distribución común en inferencia estadística y en heurísticas que modelan procesos estocásticos, ya que, por el *Teorema Central del Límite* 2.1.2.2, con cierto número de muestras, se puede trabajar bajo hipótesis de normalidad. Se trata de una distribución de tipo continuo.

**Distribución exponencial:** esta distribución está ligada a fenómenos de espera, como distribución de probabilidad del tiempo que tarda en ocurrir un suceso. Una variable aleatoria se distribuye de manera exponencial, con parámetro  $\theta > 0$  ( $X \sim \text{Exp}(\theta)$ ), si su función de densidad es

$$f(x) = \theta e^{-\theta x} I_{(x>0)}(x)$$

Se trata de una distribución también de tipo continuo.

**Distribución de Bernoulli:** un experimento Bernoulli consta de sólo dos posibles resultados en cada realización, denotando  $p$  como éxito y  $q = 1 - p$  como fracaso. Una variable aleatoria sigue una distribución de Bernoulli de parámetro  $p$  ( $X \sim BE(p)$ ) si su *función de masa* es de la forma

$$P(X = x) = p^x(1 - p)^{1-x}, \quad x \in \{0, 1\}$$

**Distribución Binomial:** repitiendo un experimento de tipo Bernoulli  $n$  veces, siendo cada repetición independiente del resto, la variable aleatoria  $X = x$  representa el número de éxitos totales. Si para cada repetición tenemos una variable aleatoria  $X_i, \forall i \in \{1, \dots, n\}$ , de forma que  $X_i = 1$ , si se ha tenido éxito en la  $i$ -ésima repetición,  $X_i = 0$  en caso contrario, y definimos  $X = \sum_{i=1}^n X_i$ , dicha variable aleatoria sigue una distribución binomial si

$$P(x; n, p) = \begin{cases} \binom{n}{x} p^x (1 - p)^{n-x} & x = 0, 1, 2, \dots, n \\ 0 & \text{en otro caso } (0 \leq p \leq 1, n \in \mathbb{Z}). \end{cases}$$

### 2.1.2.1. Conceptos de estadística inferencial

**Muestra:** en este contexto, una variable aleatoria  $X$  se conoce como *población*. Dada una población  $X$ , una *m.a.s.* de tamaño  $n$ , es la repetición de  $X_1, X_2, \dots, X_n$  de variables aleatorias independientes entre sí, con distribución igual a la de  $X$ .

**Esperanza:** se trata del promedio ponderado. Si la variable aleatoria es discreta, con  $p_k = P(X = x_k)$ , siendo  $k \geq 1, \dots$  como función de masa, y se verifica que  $\sum_{k=1}^{\infty} |x_k| \cdot p_k < \infty$ , la esperanza matemática de la variable aleatoria se define como:

$$\mu = E(X) = \sum_{k=1}^n x_k \cdot p_k$$

Si es una variable aleatoria continua, con función de densidad  $f(x)$  y sucede que  $\int_{-\infty}^{\infty} |x| \cdot f(x) dx < \infty$ , entonces la esperanza matemática se define como:

$$\mu = E(X) = \int_{-\infty}^{\infty} x \cdot f(x) dx$$

**Media, varianza y desviación estándar:** ...

### 2.1.2.2. Muestreo y convergencia

**Convergencia casi segura:** una sucesión de variables aleatorias  $\{X_n : n \geq 1\}$  converge *casi seguro* a una variable aleatoria  $X$ , denotándose como  $X_n \xrightarrow{c.s.} X$ , si y sólo si:

$$P\left(w \in \Omega : \lim_{n \rightarrow \infty} X_n(w) = X(w)\right) = 1$$

**Ley Fuerte de los Grandes Números:** una sucesión de variables aleatorias  $\{X_n : n \geq 1\}$  obedece dicha ley, respecto de una sucesión de constantes de normalización  $\{B_n : n \geq 1\}$ , con  $B_n > 0$  y  $B_n \rightarrow \infty$ , si existe una sucesión de constantes de centralización, reales,  $\{A_n : n \geq 1\}$ , tal que:

$$\frac{S_n - A_n}{B_n} \xrightarrow{c.s.} 0, \text{ cuando } n \rightarrow \infty.$$

De aquí se deduce que, al promediar varias simulaciones, se mejora la evaluación de los valores esperados.

**Teorema Central del Límite :** Sea  $X_1, X_2, \dots$  una sucesión de variables aleatorias independientes e idénticamente distribuidas (en adelante, v.a.i.i.d.), cuyas funciones generatrices de momentos existen centradas en 0. Sea  $E(X_i) = \mu$  y  $Var(X_i) = \sigma^2 > 0$ . Definimos la *media muestral* como  $\bar{X} = \frac{1}{n} \sum_{i=1}^n X_i$ . Además, sea  $G_n(x)$  la función de distribución de  $\frac{\sqrt{n}(\bar{X}_n - \mu)}{\sigma}$ . Así, para todo real  $x \in \mathbb{R}$ , se tiene:

$$\lim_{n \rightarrow \infty} G_n(x) = \int_{-\infty}^x \frac{1}{\sqrt{2\pi}} e^{-x^2/2} dx$$

Lo que resulta en la convergencia a la distribución  $\sqrt{n} \frac{\bar{X}_n - \mu}{\sigma} \xrightarrow{d} \mathcal{N}(0, 1)$ .

**Muestreo aleatorio:** dependiendo del método seguido para la extracción de la muestra, puede ser de los siguientes tipos:

- **Muestreo con reemplazamiento:** la elección de cada miembro de la población es equiprobable. Si el tamaño de la población es  $N$ , cada muestra tiene probabilidad  $\frac{1}{N}$ . Es de uso frecuente cuando la población no se compone de elementos tangibles, como la concentración de una sustancia en una disolución. Se repite la observación  $n$  veces, obteniendo  $X_n$  observaciones. También, si la población consta de elementos tangibles, se seleccionan  $n$ , extrayendo cada uno de ellos, observándolo y devolviéndolo a la población, acabando así en  $P(X_1 = x_1, X_2 = x_2, \dots, X_n = x_n) = \frac{1}{N^n}$ .
- **Muestreo sin reemplazamiento:** se extraen  $n$  elementos de la población  $N$ , pero, tras la observación de cada uno de ellos, no se devuelven.

Así,  $P(X_1 = x_1) = \frac{1}{N}$ ; y  $P(X_2 = x_2 / X_1 = x_1) = \frac{1}{N-1}$ , si  $x_2 \neq x_1$  y 0 en otro caso.

Tras estas definiciones, si queremos obtener una muestra aleatoria simple (m.a.s.) de tamaño  $n$ , sólo ocurrirá si se realiza un muestreo con reemplazamiento. Sin embargo, si  $N$  es lo bastante grande, ambos procedimientos, con reemplazo y sin reemplazo, acabarán dando probabilidades casi exactas: una,  $\frac{1}{N}$ , y otra,  $\frac{1}{N-1}$ .

### 2.1.2.3. Métodos de simulación y aproximación

**Simulación de Monte-Carlo:**

**Método de aceptación-rechazo:**

**Cadenas de Markov:** sea una colección de variables aleatorias  $X_1, X_2, \dots, X_n$ . Tomando  $X_i$  como el estado del sistema en el momento  $i$ , si los posibles valores de  $X_i$  (estados del sistema), van desde 1 hasta  $N$  y hay un conjunto  $p_{ij}$  con  $i, j \in \{1, \dots, N\}$  de números positivos menores que 1 tales que denotan la probabilidad de transitar del estado  $i$  al estado  $j$ , entonces la colección  $\{X_n, n \geq 1\}$  se conoce como *cadena de Markov* [2].

## OPTIMIZACIÓN A PARTIR DE AQUÍ

## 2.2. Optimización Heurística

La *optimización matemática* es una técnica de investigación operativa utilizada en la toma de decisiones. El campo de la investigación operativa se sirve, además, de otras herramientas, como la estadística, la teoría de la probabilidad, la simulación, y el análisis de decisiones.

En particular, la optimización matemática, en su búsqueda de la mejor solución posible a un problema, requiere de tres elementos:

- **Modelo de optimización:** constituido de variables de decisión, restricciones y funciones objetivo.
- **Datos:** para las demandas del modelo.
- **Algoritmo:** para resolver la instancia del modelo.

Con ello, se buscan los valores de las variables de decisión que satisfagan todas las restricciones y que optimicen las funciones objetivo.

### 2.2.1. Problemas de optimización

Para plantear un *problema de optimización*, hay que definir los elementos anteriormente citados.

Las **variables de decisión** en un modelo de optimización representan los parámetros variables del problema. Sus valores varían según marque el algoritmo, con el fin de obtener aquellos que mejor se ajusten a las funciones objetivo. Los aspectos más relevantes de estas variables son, el *dominio* al cual pertenecen, los *límites* entre los que se disponen (y que son delimitados por la frontera del dominio) y, finalmente, el *tipo* de variable, ya que puede ser real, entero, booleano, etc. Es importante hacer una elección coherente de estas características, pues ejerce un serio impacto en la redacción de las restricciones, la elección del algoritmo y, por tanto, en los resultados obtenidos.

Las **restricciones** representan los límites del conjunto de soluciones que se está dispuesto a admitir. Se elaboran mediante relaciones entre las variables de decisión, así como entre sus valores.

Las **funciones objetivo** en un modelo de optimización son representaciones matemáticas

del ámbito que se desea optimizar. Estas funciones pueden ser *maximizadas* o *minimizadas*, y su tratamiento, ya sea individual o combinado en problemas con múltiples objetivos, determina el tipo de algoritmo de optimización.

Una vez sentados estos elementos y finalizado el algoritmo, se llega a la **solución** del problema, la cual puede ser una combinación de solución *factible* (satisface las restricciones), *óptima* (factible que alcanza el mejor valor posible), *no factible* (fuera de la región factible, por lo que viola alguna restricción), y *no acotada* (tiende a más o a menos infinito en maximización y en minimización, respectivamente).

Más formalmente, sea  $\mathbf{x}$  el vector de  $N$  variables de decisión del sistema,  $\Omega$  el espacio sobre el cual están definidas las soluciones factibles,  $f_i(\mathbf{x})$  cada una de las  $O$  funciones objetivo del problema, y  $\text{optG}(\cdot)$  el método para conjuntar y optimizar los objetivos de manera sincrónica. A su vez, el espacio  $\Omega$  queda definido por el tipo  $\mathbf{T}$  de cada variable, y las relaciones  $g_j(\mathbf{x})$ ,  $h_k(\mathbf{x})$  que definen las restricciones del problema:

$$\begin{aligned} \text{optG}(f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_O(\mathbf{x})) \\ \mathbf{x} = [x_1, x_2, \dots, x_N] \\ \mathbf{x} \in \Omega \end{aligned} \quad \mathbf{x} \in \Omega = \begin{cases} x_i \in T_i \\ g_j(\mathbf{x}) \leq 0, j = 1..R \\ h_k(\mathbf{x}) = 0, k = 1..P \end{cases}$$

Existen, entre los métodos de combinación y optimización simultánea, dos grandes familias a destacar:

- Los métodos que convierten un problema con  $O > 1$  objetivos en sólo un problema mediante una *combinación* (a menudo, lineal) de todos los objetivos y que optimizan el valor combinado, como, por ejemplo,  $\min_{\mathbf{x}} \sum_{i=1}^O w_i f_i(\mathbf{x})$ .
- Los métodos que optimizan de forma *simultánea* todos los objetivos del problema utilizando una definición en la que se establece si una solución es mejor que otra, comparando cada uno de los objetivos por separado. Por ejemplo, con la definición de *Pareto optimalidad*, una solución  $\mathbf{A}$  es mejor que una solución  $\mathbf{B}$  si para todo  $i$ ,  $f_i(\mathbf{A})$  no es peor que  $f_i(\mathbf{B})$ , y existe un  $j$  tal que  $f_j(\mathbf{A})$  es mejor que  $f_j(\mathbf{B})$ . Este procedimiento se conoce como *frentes Pareto* de soluciones igualmente óptimas.

Hay que remarcar que los problemas *monoobjetivo* con un solo objetivo son un caso particular de los problemas *multiobjetivo* con objetivos múltiples, y en ellos no es estrictamente necesario el uso de una función  $G(\cdot)$ .

Más definiciones formales de los problemas de optimización son:

- **Vector de decisión factible:** vector  $\mathbf{x}$  que cumple las restricciones, es decir,  $\mathbf{x} \in \Omega$ .
- **Óptimo local:** un vector de decisión factible  $\mathbf{x}^*$  representa un *óptimo local* de la función  $G(f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_O(\mathbf{x}))$  si existe  $\varepsilon > 0$  tal que  $G(f_1(\mathbf{x}^*), f_2(\mathbf{x}^*), \dots, f_O(\mathbf{x}^*))$  no es peor que  $G(f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_O(\mathbf{x}))$ ,  $\forall \mathbf{x}$  en el entorno  $\|\mathbf{x} - \mathbf{x}^*\| < \varepsilon$ .
- **Óptimo global:** un vector de decisión factible  $\mathbf{x}^*$  representa un *óptimo global* de la función  $G(f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_O(\mathbf{x}))$  si  $G(f_1(\mathbf{x}^*), f_2(\mathbf{x}^*), \dots, f_O(\mathbf{x}^*))$  no es peor que  $G(f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_O(\mathbf{x}))$ ,  $\forall \mathbf{x} \in \Omega$ .

- **Punto de silla:** un vector de decisión factible  $\mathbf{x}^* = [\mathbf{x}_A^*, \mathbf{x}_B^*]$  representa un *punto de silla* de la función  $G(f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_O(\mathbf{x}))$  si existe  $\varepsilon > 0$  tal que  $G(f_1([\mathbf{x}_A, \mathbf{x}_B^*]), f_2([\mathbf{x}_A, \mathbf{x}_B^*]), \dots, f_O([\mathbf{x}_A, \mathbf{x}_B^*]))$  no es peor que  $G(f_1([\mathbf{x}_A^*, \mathbf{x}_B]), f_2([\mathbf{x}_A^*, \mathbf{x}_B]), \dots, f_O([\mathbf{x}_A^*, \mathbf{x}_B]))$ , y éste, a su vez, no es peor que  $G(f_1([\mathbf{x}_A^*, \mathbf{x}_B]), f_2([\mathbf{x}_A^*, \mathbf{x}_B]), \dots, f_O([\mathbf{x}_A^*, \mathbf{x}_B]))$ ,  $\forall \mathbf{x}$  en  $\|[\mathbf{x}_A, \mathbf{x}_B] - [\mathbf{x}_A^*, \mathbf{x}_B^*]\| < \varepsilon$ .

En cuanto a los problemas, atendiendo a distintos criterios, se pueden clasificar según:

**1. Existencia de restricciones:**

- a) *Sin restricciones:* en  $\Omega$  sólo se define el tipo de variable.
- b) *Con restricciones:* en  $\Omega$  se definen restricciones  $g_j(\mathbf{x}) \leq 0$ , o  $h_k(\mathbf{x}) = 0$ .

**2. Naturaleza de las variables de decisión:**

- a) *Estáticos:* la variable tiempo no es relevante.
- b) *Dinámicos:* el tiempo es la variable de la que dependen el resto de variables.

**3. Número de objetivos:**

- a) *Mono-objetivo:* se optimiza un sólo objetivo.
- b) *Multi-objetivo:* se optimizan varios objetivos.

**4. Naturaleza de las funciones involucradas:**

- a) *Problemas lineales:* todas las funciones objetivo de las restricciones son lineales.
- b) *Problemas no lineales:* alguna de las funciones objetivo o restricciones no es lineal.
- c) *Problemas de programación geométrica:* las funciones objetivo y las restricciones son expresables en forma de polinomios de las variables de decisión.
- d) *Problemas cuadráticos:* caso especial de problema no lineal en el que las funciones objetivo son cuadráticas y las restricciones son lineales.

**5. Valores permisibles de las variables de decisión:**

- a) *Problemas de programación reales:* las variables de decisión toman cualquier valor real.
- b) *Problemas de programación entera:* las variables de decisión toman sólo valores enteros (o discretos). Como caso particular, se tienen las de variable booleana (0 ó 1).
- c) *Problemas de programación mixta:* una mezcla de los anteriores.

**6. Naturaleza determinista de las variables de decisión:**

- a) *Problemas deterministas:* todos los parámetros del problema se conocen con certeza y, por tanto, se pueden evaluar las funciones.
- b) *Problemas estocásticos:* existe incertidumbre sobre alguno o todos los parámetros del problema, y se tienen que dar definiciones probabilísticas para algunas funciones.

**7. Separabilidad de las funciones:**

- a) *Problemas separables*: tanto las funciones objetivo como las restricciones pueden separarse en suma de funciones de diferentes variables de decisión. Si la suma se realiza de manera independiente sobre cada una de las variables de decisión, se dice que los problemas son *totalmente separables*.
- b) *Problemas no separables*: caso contrario al anterior.

#### 8. Complejidad computacional:

- a) *Problemas de clase P*: se conocen algoritmos capaces de obtener la solución óptima del problema en un tiempo polinomial, según el tamaño del problema. Se suele expresar como  $O(p(n))$ , con  $n$  el tamaño,  $p(\cdot)$  un polinomio, y  $O(\cdot)$  su orden de complejidad.
- b) *Problemas de clase NP*: un algoritmo no puede resolverlo en tiempo polinomial, o bien no se conoce. A su vez, pueden ser *NP-hard* o *NP-completos*. Existen dos vías para enfrentar estos problemas: o bien utilizar un algoritmo que asegure la optimalidad pero el tiempo de cómputo sea excesivo, o bien encontrar una solución no óptima pero aceptable, en un tiempo razonable.

En este último criterio, los algoritmos del primer grupo se conocen como *exactos*, y los del segundo grupo, *aproximados*. Algunos de estos algoritmos aproximados se los denominan *heurísticos*.

## 2.3. Algoritmos Heurísticos

En inteligencia artificial, el término *heurístico* hace referencia, en especial, a un procedimiento que intenta aportar soluciones, razonablemente buenas, teniendo en cuenta la calidad de las soluciones y los recursos empleados. En investigación operativa es similar, pero no se asegura la optimalidad ni la factibilidad de las soluciones, así como tampoco lo cerca de ellas que se está. También se usa el término *heurística* cuando se realizan modificaciones en el procedimiento de la solución en pos de mejorar su rendimiento.

En el contexto de las *metaheurísticas*, este término se utiliza para referirse a algoritmos heurísticos que integran un conjunto de mecanismos de control diseñados para optimizar el funcionamiento de la heurística, y mejorar así su rendimiento. Otra acepción sería un algoritmo basado en procedimientos heurísticos “inteligentes”.

Las propiedades que se desea que posean estos algoritmos son *sencillez*, *precisión* (sin ambigüedades), *eficiencia* (da soluciones de alta calidad y se aprovechan los recursos), *generalidad* (aplicable a muchos problemas), *robustez* (su comportamiento apenas varía con modificaciones o cambio de contexto), *multiplicidad de soluciones* (a elección del usuario), *flexibilidad* en los parámetros y restricciones y, finalmente, *maleabilidad* (para particularidades y generalidades de cada problema).

Atendiendo al modo en el que buscan y construyen sus soluciones, las cuatro grandes familias son:

1. **Metaheurísticas de relajación**: se resuelven versiones menos restrictivas y, por tanto, más fáciles de resolver que el problema original.

2. **Metaheurísticas constructivas:** se obtiene la solución del problema a partir del análisis y selección gradual de las componentes que la forman.
3. **Metaheurísticas de búsqueda:** se recorre el espacio de soluciones mediante transformaciones y movimientos que aprovechan la estructura del problema.
4. **Metaheurísticas evolutivas:** los valores de un conjunto de soluciones evolucionan de forma paralela, acercándose según aumenta el número de iteraciones, a la solución óptima.

### 2.3.1. Metaheurísticas de búsqueda

Las *metaheurísticas de búsqueda* establecen estrategias para recorrer el espacio de soluciones del problema, transformando de forma iterativa, y a través de reglas astutas, una solución inicial en otra.

A su vez, hay dos grandes familias: las *técnicas de búsqueda local* y las *técnicas de búsqueda global*.

Por un lado, las *técnicas de búsqueda local* se basan en alguna regla inteligente que modifica una solución del problema. Ésta se aplica de forma iterativa hasta que no se pueda mejorar la solución previa. Las modificaciones sólo permiten trasladar una solución actual a una de su vecindad. Se dejarán de realizar estas modificaciones si no se encuentra ninguna solución mejor que la de partida. Si la hubiera, se utilizará la mejor solución hallada para la solución de partida de la siguiente iteración. Este tipo de técnicas están relacionadas con *algoritmos de búsqueda monótona* o con *técnicas de escalado* (*hill-climbing*). Además, al elegir la mejor solución posible determinada por la regla elegida, convierte estas metaheurísticas en *algoritmos voraces* (*greedy*). La desventaja primordial de las búsquedas locales es que pueden quedar fácilmente “atrapadas” en un *óptimo local*. Esto es, no puede mejorarse una solución mediante las transformaciones de la heurística.

Por otro lado, las *metaheurísticas de búsqueda global* pretenden eliminar la localidad extendiendo la búsqueda de los algoritmos a otras regiones del espacio. Esto se consigue mediante varias mecánicas, como son las *búsquedas de arranque múltiple* (*Multi Start Search*), en las que se reinicia la búsqueda a partir de otra solución inicial distinta; las *búsquedas por entorno variable* (*Variable Neighbourhood Search*, *VNS*), en las que se modifica el tipo de movimiento en el entorno definido por la regla para evitar quedar atrapada en una región rígida; *búsquedas no monótonas*, en las que se admite que el movimiento empeore durante la búsqueda, ya sea utilizando memoria para que con la información adquirida no se caiga en una zona concreta del espacio, evitando momentáneamente soluciones muy parecidas entre sí, como en la *búsqueda tabú* (*Taboo Search*, *TS*), o bien se establezcan normas para controlar la probabilidad de aceptar una solución igual o peor, como en el *enfriamiento simulado* (*Simulated Annealing*), en el que la probabilidad de aceptación viene dada por una función exponencial del empeoramiento producido.



### 2.3.2. Metaheurísticas evolutivas

Las *metaheurísticas evolutivas* establecen estrategias para guiar la evolución de un conjunto de posibles soluciones, al que llamaremos *población*, hacia la solución óptima.

La exploración del espacio de soluciones se hace en cada iteración con una población, modificando cada individuo teniendo en cuenta los valores del resto y las funciones de ajuste asociadas. Esto último no es estrictamente necesario, pues habrá regiones no explorables con la información exclusiva de la población, por lo que se pueden añadir mecanismos adicionales. En cada iteración, se sustituye la población con nuevas soluciones y, en ocasiones, parte de las ya existentes. En algunos algoritmos se guardarían los mejores individuos para tenerlos en cuenta posteriormente. La distinción entre metaheurísticas evolutivas se hace respecto a cómo combinan la información obtenida de la población.

Entre otros, existen tres grandes grupos:

**Algoritmos genéticos** (*Genetic Algorithms, GA*), en los que, en cada iteración (generación), mantiene una población de posibles soluciones que evolucionan hacia las soluciones óptimas del problema mediante una selección de individuos (selección natural) y operadores genéticos (cruce y mutación). Se inspira en la teoría de la evolución de Darwin.

**Nubes de partículas** (*Particle Swarm Optimization, PSO*), en las que se mantiene un conjunto de posibles soluciones (partículas) que son desplazadas por el espacio de búsqueda teniendo en cuenta el desplazamiento anterior, el mejor valor hallado por la partícula (líder o mejor local), y el mejor valor hallado por todas las partículas (líder o mejor global). Está inspirado en el comportamiento de bandadas de aves o bancos de peces.

**Evolución diferencial** (*Differential Evolution, DE*), en la que, a lo largo de las iteraciones, el conjunto de posibles soluciones son desplazadas incrementalmente (diferencial) teniendo en cuenta las diferencias entre pares de soluciones del conjunto, adaptándose a la diversidad de soluciones existentes. Los métodos de escalada o gradiente son de este estilo.

# Capítulo 3

## Métodos de Simulación de Monte-Carlo

Este capítulo se centra en la introducción de varios métodos de integración sobre una función de probabilidad combinando la integración de Monte Carlo con alguna técnica de simulación estocástica. Además, se presentarán otros dos grupos de métodos de este tipo para la generación de muestras que, junto con lo anterior, permiten calcular el valor de integrales sobre funciones de probabilidad.[3]

### 3.1. Integración de Monte-Carlo

Para estudiar este método en profundidad, utilizaremos la siguiente notación:  $U$  representa una variable aleatoria y  $\mathbf{U}$  un conjunto de ellas, y en minúscula,  $u$ ,  $\mathbf{u}$ , para valores específicos, respectivamente.  $\mathbf{V}$  será el conjunto de todas las variables del problema, siendo  $X$ ,  $x$ ,  $\mathbf{X}$ ,  $\mathbf{x}$ , las variables ocultas (desconocidas) e  $Y$ ,  $y$ ,  $\mathbf{Y}$ ,  $\mathbf{y}$ , las observadas (medidas). Para las variables de un conjunto, se usará un subíndice ( $U_g \in \mathbf{U}$ ), y para un subconjunto de variables indexadas desde  $k$  hasta  $s$  usaremos un subíndice  $k:s$ . Para el  $i$ -ésimo valor de una muestra se indicará con el superíndice  $(i)$  y para su peso,  $w(\cdot)$ .

En cuanto a funciones, la densidad de probabilidad original sobre la que se desea integrar será  $p(\cdot)$ , mientras que  $f(\cdot)$  será una función genérica. Además, una densidad de probabilidad genérica será  $Pr(\cdot)$ , la densidad auxiliar para obtener las muestras se denotará  $q(\cdot)$ , la operación de muestreo respecto a una densidad  $s(\mathbf{U})$  como  $\mathbf{u}^{(i)} \sim s(\mathbf{U})$  y, finalmente, la función delta de Dirac,  $\delta(\cdot)$ .

Esta técnica consigue aproximar el resultado de la integral 3.1, cuyo valor corresponde con el esperado de una serie de funciones, útil para la resolución de problemas de aprendizaje, cuyo cálculo exacto se reduce a un conjunto muy específico de funciones.

$$I = E_{p(\mathbf{U})} [f(\mathbf{U})] = \int f(\mathbf{U}) p(\mathbf{U}) d\mathbf{U} \quad (3.1)$$

La técnica de integración de Monte-Carlo evalúa  $f(\mathbf{U})$  sobre un conjunto  $\{\mathbf{u}^{(1)}, \mathbf{u}^{(2)}, \dots, \mathbf{u}^{(N)}\}$  distribuidas según  $p(\mathbf{U})$ .

El método de integración, pues, aproxima 3.1 por medio de la siguiente expresión:

$$I_N = \frac{1}{N} \sum_{i=1}^N f(\mathbf{u}^{(i)}) \quad \text{con} \quad \mathbf{u}^{(i)} \sim p(\mathbf{U}) \quad (3.2)$$

lo que permite evaluar  $f(\mathbf{U})$  en puntos de  $\mathbf{U}$  donde la probabilidad no sea nula. Esto se consigue mediante la aproximación de la función original  $p(\mathbf{U})$  como sigue:

$$p_N(\mathbf{U}) = \frac{1}{N} \sum_{i=1}^N \delta(\mathbf{U} - \mathbf{u}^{(i)}) \quad \text{con} \quad \mathbf{u}^{(i)} \sim p(\mathbf{U}) \quad (3.3)$$

Sustituyendo en 3.1  $p(\mathbf{U})$  por  $p_N(\mathbf{U})$  para comprobar la veracidad de la aproximación:

$$\begin{aligned} I &= \int f(\mathbf{U}) p(\mathbf{U}) d\mathbf{U} \simeq \int f(\mathbf{U}) p_N(\mathbf{U}) d\mathbf{U} = \\ &= \frac{1}{N} \sum_{i=1}^N \int f(\mathbf{U}) \delta(\mathbf{U} - \mathbf{u}^{(i)}) d\mathbf{U} = \frac{1}{N} \sum_{i=1}^N f(\mathbf{u}^{(i)}) = I_N \end{aligned}$$

A través de la ley fuerte de los grandes números se puede comprobar que, si  $N \rightarrow \infty$ , entonces converge casi seguro  $I_N \xrightarrow{c.s.} I$ . También que, para ciertos pares de funciones  $f(\mathbf{U})$ ,  $p(\mathbf{U})$ , la varianza de  $I_N$  decrece según aumenta  $N$ .

## 3.2. Simulación estocástica

La simulación estocástica es el mecanismo responsable de generar el conjunto de muestras y, según sea su naturaleza, distinguir unos métodos de simulación de Monte-Carlo de otros.

## 3.3. Técnicas de generación de muestras

Las siguientes técnicas forman parte de un amplio conjunto de métodos de muestreo auxiliar. Será clave la noción de *soporte* de una función  $p(\mathbf{U})$ , que se define como una región o un conjunto de ellas donde sucede  $p(\mathbf{U}) > 0$ . Generalmente, es necesario poder evaluar dicha función o, al menos, un múltiplo de ella, pues en muchas ocasiones se trabajará con la proporción de dos evaluaciones de la función de densidad, por lo que la constante se cancela.

### 3.3.1. Muestreo por rechazo

El muestreo por rechazo (*Rejection Sampling*, *RS*) genera un conjunto de muestras distribuidas según la función de densidad de probabilidad original  $p(\mathbf{U})$  a través de:

1. Una función de densidad auxiliar  $q(\mathbf{U})$  para generar los candidatos  $\mathbf{u}^{(i)}$ .
2. Una función de probabilidad uniforme  $\mathcal{U}_{(0,1)}(A)$  para decidir si los candidatos  $\mathbf{u}^{(i)}$  son aceptados con una probabilidad proporcional a la diferencia de  $p(\mathbf{u}^{(i)})$  y  $q(\mathbf{u}^{(i)})$ .

La función auxiliar debe satisfacer  $Mq(\mathbf{U}) - p(\mathbf{U}) > 0$ , con  $0 < M < \infty$ , lo que fuerza a que tenga el mismo soporte que la original, envolviéndola y, además, haciendo que  $\frac{p(\mathbf{U})}{Mq(\mathbf{U})} \leq 1$ . Así,  $\mathbf{u}^{(i)} \sim q(\mathbf{U})$  es aceptado si  $a^{(i)}Mq(\mathbf{u}^{(i)}) < p(\mathbf{u}^{(i)})$ . En otro caso se rechaza y los valores de  $\mathbf{u}^{(i)}$  y de  $a^{(i)}$  se reemplazan por otro par.

```

1.   $i = 1$ 
2.  while ( $i \leq N$ )
3.       $\mathbf{u}^{(i)} \sim q(\mathbf{U}), \quad a^{(i)} \sim \mathcal{U}_{(0,1)}(A)$ 
4.      if  $a^{(i)}Mq(\mathbf{u}^{(i)}) < p(\mathbf{u}^{(i)})$  then  $i = i + 1$ 

```

Figura 3.1: Método de muestreo por rechazo.

Esta aceptación y rechazo sucede con una probabilidad  $Pr(A|\mathbf{u}^{(i)}) = p(\mathbf{u}^{(i)})/Mq(\mathbf{u}^{(i)})$  y de  $1 - Pr(A|\mathbf{u}^{(i)})$ , respectivamente. Por tanto, cuanto más cercanos sean  $p(\mathbf{U})$  y  $Mq(\mathbf{U})$  en el soporte de  $p(\mathbf{U})$  más eficiente será el método. Es conveniente elegir  $q(\mathbf{U})$  de modo que permita fijar valores pequeños de  $M$ , pues  $Pr(A) = \int Pr(A|\mathbf{U})q(\mathbf{U})d\mathbf{U} = \frac{1}{M} \int p(\mathbf{U})d(\mathbf{U}) = \frac{1}{M}$ .

También se da que  $Pr(\mathbf{U}|A) = \frac{Pr(A|\mathbf{U})q(\mathbf{U})}{Pr(A)} = \frac{p(\mathbf{U})q(\mathbf{U})M}{Mq(\mathbf{U})} = p(\mathbf{U})$ , luego  $\mathbf{u}^{(i)}$  se distribuyen según la función  $p(\mathbf{U})$  original.

Cabe destacar que el método de muestreo por rechazo es igualmente válido si se conoce una expresión de la forma  $Cte \cdot p(\mathbf{U})$ . Sin embargo, no es sencillo de implementar en problemas multidimensionales complejos, pues es difícil encontrar  $q(\mathbf{U})$  y  $M$  adecuados para generar pocos rechazos, problema solventable haciendo uso de la versión adaptativa de este mismo método.

### 3.3.2. Muestreo enfatizado

En el muestreo enfatizado (*Importance Sampling, IS*), las muestras  $\mathbf{u}^{(i)}$  se toman de una función de densidad auxiliar  $q(\mathbf{U})$  con soporte igual o mayor que el de  $p(\mathbf{U})$ . Los pesos  $w(\mathbf{u}^{(i)})$  representan la proporción de la muestra dentro de  $p(\mathbf{U})$ , midiendo la disonancia entre la densidad original y la auxiliar, es decir, lo lejos que están de ser iguales. Por tanto, al tener que  $w(\mathbf{u}^{(i)}) \propto \frac{p(\mathbf{u}^{(i)})}{q(\mathbf{u}^{(i)})}$ , es suficiente con conocer una expresión proporcional a la densidad original  $Cte \cdot p(\mathbf{U})$ . En este caso,  $p(\mathbf{U})$  e  $I$  de la ecuación 3.1 se aproximan, respectivamente, como:

$$\begin{aligned}
 p_{IS} &= \sum_{i=1}^N w(\mathbf{u}^{(i)}) \delta(\mathbf{U} - \mathbf{u}^{(i)}); \quad I_{IS} = \sum_{i=1}^N w(\mathbf{u}^{(i)}) f(\mathbf{u}^{(i)}) \\
 \text{con } \mathbf{u}^{(i)} &\sim q(\mathbf{U}), \quad w(\mathbf{u}^{(i)}) \propto \frac{p(\mathbf{u}^{(i)})}{q(\mathbf{u}^{(i)})}, \quad \sum_{i=1}^N w(\mathbf{u}^{(i)}) = 1
 \end{aligned} \tag{3.4}$$

El pseudocódigo de la Figura 3.2 recoge el algoritmo para la generación de  $N$  muestras  $\mathbf{u}^{(i)}$  con sus correspondientes pesos  $w(\mathbf{u}^{(i)})$ .

```

1.   $T = 0$ 
2.  for  $(i = 1 : N)$ 
3.       $\mathbf{u}^{(i)} \sim q(\mathbf{U}), \quad w(\mathbf{u}^{(i)}) = \frac{p(\mathbf{u}^{(i)})}{q(\mathbf{u}^{(i)})}, \quad T = T + w(\mathbf{u}^{(i)})$ 
4.  for  $(i = 1 : N)$ 
5.       $w(\mathbf{u}^{(i)}) = \frac{w(\mathbf{u}^{(i)})}{T}$ 

```

Figura 3.2: Método genérico de muestreo enfatizado.

Las ecuaciones de 3.4 se derivan de la expresión 3.1 introduciendo la función de muestreo auxiliar  $q(\mathbf{U})$ . Además, se considera que la función de muestreo  $w(\mathbf{U})q(\mathbf{U})$  puede aproximarse mediante  $p_{IS}(\mathbf{U})$  combinando las probabilidades de densidad puntual  $\delta(\mathbf{U} - \mathbf{u}^{(i)})$  centradas en las  $N$  muestras  $\mathbf{u}^{(i)}$  obtenidas de manera independiente con la función auxiliar  $q(\mathbf{U})$  y pesadas según  $w(\mathbf{u}^{(i)})$ :

$$\begin{aligned}
 I &= \int f(\mathbf{U}) p(\mathbf{U}) d\mathbf{U} = \int f(\mathbf{U}) \frac{p(\mathbf{U})}{q(\mathbf{U})} q(\mathbf{U}) d\mathbf{U} = \int f(\mathbf{U}) w(\mathbf{U}) q(\mathbf{U}) d\mathbf{U} \simeq \\
 &\simeq \int f(\mathbf{U}) w(\mathbf{U}) \frac{1}{N} \sum_{i=1}^N \delta(\mathbf{U} - \mathbf{u}^{(i)}) d\mathbf{U} = \int f(\mathbf{U}) \frac{1}{N} \sum_{i=1}^N w(\mathbf{u}^{(i)}) \delta(\mathbf{U} - \mathbf{u}^{(i)}) d\mathbf{U}
 \end{aligned}$$

La elección de  $q(\mathbf{U})$  hace controlar la región de donde se obtienen las muestras, y así introducir cualquier tipo de conocimiento previo que se tenga de la original  $p(\mathbf{U})$ . Además, como se quiere aproximar el valor integral, es conveniente minimizar la varianza. Para ello, la función auxiliar óptima es  $q^*(\mathbf{U}) = \frac{|f(\mathbf{U})|p(\mathbf{U})}{\int |f(\mathbf{U})|p(\mathbf{U})}$ , cuyo valor minimiza la varianza  $E_{q(\mathbf{U})}[w^2(\mathbf{U}) f^2(\mathbf{U})]$  (si  $E_{q(\mathbf{U})}[x]$  denota la esperanza,  $E_{q(\mathbf{U})}[x^2]$  denota la varianza) [3]. De ella no se puede muestrear directamente, pero nos señala que se pierde eficiencia si el numerador se anula. Si se quiere obtener los valores integrales para varias funciones generales  $f(\mathbf{U})$ , se puede emplear varias o una sola  $q(\mathbf{U})$ , que ignore las funciones generales e intente aproximarse a  $p(\mathbf{U})$ .

Sin embargo, la elección de una función auxiliar eficiente es más complicado en el caso multidimensional. Existen técnicas para sortear este inconveniente, como el muestreo por importancia adaptativo o el muestreo enfatizado secuencial (*Sequential Importance Sampling, SIS*), implementado en los filtros de partículas.

### 3.3.3. Remuestreo por pesos

En el método de remuestreo por pesos (*Weighted Resampling, WR*) se aproxima una función de densidad  $p_A(\mathbf{U})$ , que es una combinación de  $N_A$  funciones de densidad puntuales

$\delta(\mathbf{U} - \mathbf{u}_A^{(i)})$ , centradas en  $\mathbf{u}_A^{(i)}$  y ponderadas según  $w(\mathbf{u}_A^{(i)})$ , por otra función  $p_B(\mathbf{U})$  de la misma clase.

Los centros de  $p_B(\mathbf{U})$  son muestreados entre los valores de  $\mathbf{u}_A^{(i)}$  mediante  $r(\mathbf{U})$ , la cual indica en qué regiones del espacio generar una determinada cantidad de muestras, y la función de probabilidad  $q(I|\mathbf{u}_A^{(i)})$ , que elige las muestras de forma proporcional a su correspondiente valor  $r(\mathbf{u}_A^{(i)})$ . Los pesos  $w(\mathbf{u}_B^{(j)})$  son el cociente de los pesos en la densidad original  $w(\mathbf{u}_A^{(i)})$  entre el valor  $r(\mathbf{u}_A^{(i)})$ , el cual toman de la función de remuestreo ( $\mathbf{u}_B^{(j)} = \mathbf{u}_A^{(i)}$ ):

$$\begin{aligned} p_A(\mathbf{U}) &= \sum_{i=1}^{N_A} w(\mathbf{u}_A^{(i)}) \delta(\mathbf{U} - \mathbf{u}_A^{(i)}); \quad p_B(\mathbf{U}) = \sum_{i=1}^{N_B} w(\mathbf{u}_B^{(j)}) \delta(\mathbf{U} - \mathbf{u}_B^{(j)}) \\ \text{con } \mathbf{u}_A^{(i)} &= \mathbf{u}_B^{(j)}, \quad i \sim q(I|\mathbf{u}_A^{(i)}), \quad q(I|\mathbf{u}_A^{(i)}) \propto r(\mathbf{u}_A^{(i)}) \\ w(\mathbf{u}_B^{(j)}) &\propto \frac{w(\mathbf{u}_A^{(i)})}{r(\mathbf{u}_A^{(i)})}, \quad \sum_{i=1}^{N_B} w(\mathbf{u}_B^{(j)}) = 1 \end{aligned} \quad (3.5)$$

El pseudocódigo de la Figura 3.3 ilustra la generación de  $N$  muestras  $w(\mathbf{u}_B^{(j)})$  con sus correspondientes pesos  $w(\mathbf{u}_B^{(j)})$ .

```

1.   $T = 0$ 
2.  for ( $j = 1 : N$ )
3.       $I \sim q(I|\mathbf{u}_A^{(i)})$ ,
4.       $\mathbf{u}_B^{(j)} = \mathbf{u}_A^{(i)}, \quad w(\mathbf{u}_B^{(j)}) = \frac{w(\mathbf{u}_A^{(i)})}{r(\mathbf{u}_A^{(i)})}, \quad T = T + w(\mathbf{u}_B^{(j)})$ 
5.  for ( $j = 1 : N$ )  $w(\mathbf{u}_B^{(j)}) = \frac{w(\mathbf{u}_B^{(j)})}{T}$ 

```

Figura 3.3: Método genérico de remuestreo por pesos.

Estas aproximaciones garantizan que, cuando  $N \rightarrow \infty$ , ambas distribuciones converjan, a condición de que el soporte de  $r(\mathbf{U})$  contenga al de  $p_A(\mathbf{U})$ .

Además de elegir la función  $r(\mathbf{U})$ , cuando se utiliza un método de remuestreo por pesos, es necesario seleccionar el método de muestreo proporcional  $q(I|\mathbf{u}_A^{(i)})$ , usualmente el muestreo multinomial, el sistemático y el residual, los cuales no trataremos en esta sección [3].

La base del método de remuestreo por pesos se asemeja a la empleada por el método de muestreo enfatizado, dado que al incorporar una nueva función de muestreo proporcional a

$r(\mathbf{U})$ , esta se encuentra en el denominador de la función que determina los pesos  $w(\mathbf{u}_B^{(j)})$ . La libertad de elección de  $r(\mathbf{U})$  nos permite generar muestras en el soporte que merecen ser estudiadas en profundidad. Es habitual elegir el valor de  $r(\mathbf{u}_A^{(i)})$  igual al de los pesos  $w(\mathbf{u}_A^{(i)})$ , con el fin de que las nuevas muestras  $\mathbf{u}_B^{(j)}$  se concentren en las regiones más probables de la función de densidad original, y que  $w(\mathbf{u}_B^{(j)}) = \frac{1}{N}$ , lo cual cobra sentido en la creación de algunos filtros de partículas, sobre los que hablaremos en la sección ??.

### 3.4. Métodos de simulación por cadenas de Markov

El fundamento de método de muestreo mediante cadenas de Markov (*Markov Chains, MC*) reside en que la siguiente muestra,  $\mathbf{u}^{(i)}$ , es generada mediante un proceso estocástico definido por una función de probabilidad, *kernel de transición*,  $K(\mathbf{U}|\mathbf{u}^{(i-1)})$ , la cual sólo depende de la muestra inmediatamente anterior (*propiedad de Markov*). Para la aproximación de la función de densidad original, se usarán las  $N$  últimas muestras como sigue:

$$p_{MC} = \frac{1}{N} \sum_{j=B}^{N+B} \delta(\mathbf{U} - \mathbf{u}^{(j)}) \quad \text{con} \quad \mathbf{u}^{(j)} \sim K(\mathbf{u}^{(j)}|\mathbf{u}^{(j-1)}) \quad (3.6)$$

El pseudocódigo de la Figura 3.4 muestra cómo se generan las  $B + N$  muestras, de las cuales se desechan las  $B$  primeras.

<ol style="list-style-type: none"> <li>1. <math>i = 1, \quad j = 1, \quad \text{fijar } \mathbf{u}^{(0)}</math></li> <li>2. <math>\text{while } (i \leq N)</math></li> <li>3. <math>\mathbf{u}^{(i)} \sim K(\mathbf{U}   \mathbf{u}^{(i-1)})</math></li> <li>4. <math>\text{if } j \geq B \text{ then } i = i + 1,</math></li> <li>5. <math>\text{else } \mathbf{u}^{(i-1)} = \mathbf{u}^{(i)}, \quad j = j + 1</math></li> </ol>
---

Figura 3.4: Método genérico de muestreo por cadena de Markov.

Para que el proceso definido con el kernel de transición, conocido como cadena de Markov, genere las muestras acorde a la función de probabilidad original  $p(\mathbf{U})$ , se requiere de:

1. La convergencia de la cadena de Markov tras la toma de unas muestras iniciales posteriormente desechadas.
2. La función de probabilidad estacionaria a la que converja ha de ser  $p(\mathbf{U})$ .

Esto es, que  $K(\mathbf{U}|\mathbf{u}^{(i-1)})$  cumpla las condiciones de ser *irreducible* (visita todo el soporte en tiempo limitado), *aperiódica* (no fluctuación entre distintas regiones del soporte de forma periódica) y *recurrente de forma positiva* (para la existencia de función estacionaria).

Una condición suficiente para 2 es que la cadena sea reversible y, por tanto, en el momento estacionario la velocidad de la transición sea igual en ambos sentidos:

$$p(\mathbf{u}^{(i-1)}) K(\mathbf{u}^{(i)}|\mathbf{u}^{(i-1)}) = p(\mathbf{u}^{(i)}) K(\mathbf{u}^{(i-1)}|\mathbf{u}^{(i)}) \quad (3.7)$$

Una función de probabilidad  $p(\mathbf{U})$  puede dar lugar a distintas cadenas de Markov, cada una con una velocidad a priori distinta, lo que hace más beneficioso el uso de unas frente a otras. Al tomar las últimas  $N$  muestras, antes de la convergencia, las  $B$  primeras se suelen desechar (*burn-in*), luego la aproximación de  $p(\mathbf{U})$  se encuentra en el rango entre  $B$  y  $B + N$ . La elección del valor inicial de  $\mathbf{u}^{(0)}$  influye en el valor del *burn-in*, lo que a su vez repercute, además del número necesario de muestras  $N$ , en el error de la aproximación mediante  $p_{MC}$ . Por ello, se suelen emplear otras técnicas avanzadas para su obtención. También es posible generar los datos utilizando cadenas independientes y/o utilizar los subconjuntos de muestras obtenidas una vez dada la convergencia. Finalmente, pese a la dificultad de construcción de una cadena  $K(\mathbf{U}|\mathbf{u}^{(i-1)})$ , existen algunas cadenas generales ya definidas dentro de algunos métodos con un uso amplio plausible.

### 3.4.1. Cadena general para los métodos de Metrópolis-Hastings y Gibbs

El tipo de cadena elegida para estos procedimientos permite atacar problemas en los que se conoce  $Cte \cdot p(\mathbf{U})$  en vez de  $p(\mathbf{U})$ . La generalidad de  $K(\mathbf{U}|\mathbf{u}^{(i-1)})$  recae en la posibilidad de modelar su comportamiento mediante la densidad condicionada  $q(\mathbf{U}|\mathbf{u}^{(i-1)})$ .

La cadena genera valores para  $\mathbf{u}^{(i)}$  a partir de la densidad condicionada anterior, con probabilidad de aceptación  $\Pr(A|\mathbf{U}, \mathbf{u}^{(i-1)})$ , e iguales a  $\mathbf{u}^{(i-1)}$  con probabilidad de rechazo de  $\Pr(\neg A|\mathbf{u}^{(i-1)})$ . La probabilidad de aceptación de  $\Pr(A|\mathbf{U}, \mathbf{u}^{(i-1)})$  está supeditada a los valores de  $p(\cdot)$ ,  $q(\cdot)$ , evaluadas de manera directa e inversa sobre los valores de  $\mathbf{U} = \mathbf{u}^{(i)}$  y  $\mathbf{u}^{(i-1)}$ .

Así, por construcción, se asegura que la cadena  $K(\mathbf{U}|\mathbf{u}^{(i-1)})$  sea reversible (3.7) y, por tanto, que la función de probabilidad estacionaria sea  $p(\mathbf{U})$ . Además, forzando que el soporte de  $q(\mathbf{U}|\mathbf{u}^{(i-1)})$  incluya el soporte de  $p(\mathbf{U})$  se logra la irreducibilidad y la aperioidicidad por la posibilidad de no aceptar la muestra candidata mediante  $q(\mathbf{U}|\mathbf{u}^{(i-1)})$ . Por tanto, con estas elecciones la cadena  $K(\mathbf{U}|\mathbf{u}^{(i-1)})$  consigue generar muestras según  $p(\mathbf{U})$ .

Dicha cadena se define de la siguiente manera:

$$\begin{aligned} K(\mathbf{U}|\mathbf{u}^{(i-1)}) &= q(\mathbf{U}|\mathbf{u}^{(i-1)}) \Pr(A|\mathbf{U}, \mathbf{u}^{(i-1)}) + \\ &\quad + \delta(\mathbf{U} - \mathbf{u}^{(i-1)}) \Pr(\neg A|\mathbf{u}^{(i-1)}) \\ \text{con } \Pr(A|\mathbf{U}, \mathbf{u}^{(i-1)}) &= \min \left\{ 1, \frac{p(\mathbf{U}) q(\mathbf{u}^{(i-1)}|\mathbf{U})}{p(\mathbf{u}^{(i-1)}) q(\mathbf{U}|\mathbf{u}^{(i-1)})} \right\} \\ \Pr(\neg A|\mathbf{u}^{(i-1)}) &= \int \Pr(\neg A|\mathbf{U}, \mathbf{u}^{(i-1)}) q(\mathbf{U}|\mathbf{u}^{(i-1)}) d\mathbf{U} \\ \Pr(\neg A|\mathbf{U}, \mathbf{u}^{(i-1)}) &= 1 - \Pr(A|\mathbf{U}, \mathbf{u}^{(i-1)}) \end{aligned} \quad (3.8)$$



Al aparecer el cociente  $p(\mathbf{U})/p(\mathbf{u}^{(i-1)})$ , no es necesario conocer la expresión de  $p(\mathbf{U})$ , por lo que bastaría considerar un múltiplo.

### 3.4.2. Algoritmos de Metropolis-Hastings

Los algoritmos de esta familia se desarrollan utilizando directamente la cadena de Markov descrita en la sección anterior, implementada en un programa que la emula. El pseudocódigo mostrado en la Figura 3.5 da una posible implementación.

El soporte de la densidad  $q(\mathbf{U}|\mathbf{u}^{(i-1)})$  debe incluir al de la original  $p(\mathbf{U})$  y la elección de ésta influirá en la velocidad de convergencia del método.

Por un lado, una variante de esta familia de algoritmos es la de Metropolis, en la que la densidad auxiliar se toma simétrica,  $q(\mathbf{U}|\mathbf{u}^{(i-1)}) = q(\mathbf{u}^{(i-1)}|\mathbf{U})$ , quedando entonces

$$\Pr(A|\mathbf{U}, \mathbf{u}^{(i-1)}) = \min \left\{ 1, \frac{p(\mathbf{U})}{p(\mathbf{u}^{(i-1)})} \right\}.$$

```

1.   $i = 1, \quad j = 1, \quad \text{fijar } \mathbf{u}^{(0)}$ 
2.  while ( $i \leq N$ )
3.       $\mathbf{u}^{(i)} \sim q(\mathbf{U} | \mathbf{u}^{(i-1)}), \quad a^{(i)} \sim \mathcal{U}_{(0,1)}(A)$ 
4.       $\Pr(A|\mathbf{u}^{(i)}, \mathbf{u}^{(i-1)}) = \min \left\{ 1, \frac{p(\mathbf{u}^{(i)}) q(\mathbf{u}^{(i-1)}|\mathbf{u}^{(i)})}{p(\mathbf{u}^{(i-1)}) q(\mathbf{u}^{(i)}|\mathbf{u}^{(i-1)})} \right\}$ 
5.      if  $a^{(i)} > \Pr(A|\mathbf{u}^{(i)}, \mathbf{u}^{(i-1)})$  then  $\mathbf{u}^{(i)} = \mathbf{u}^{(i-1)}$ 
6.      if  $j \geq B$  then  $i = i + 1$ 
7.      else  $\mathbf{u}^{(i-1)} = \mathbf{u}^{(i)}, \quad j = j + 1$ 

```

Figura 3.5: Algoritmo de Metropolis-Hastings.

Por otro lado, la variante conocida como *muestreador independiente* asume que  $q(\mathbf{U}|\mathbf{u}^{(i-1)}) = q(\mathbf{U})$ , de ahí el nombre, lo que resulta en  $\Pr(A|\mathbf{U}, \mathbf{u}^{(i-1)}) = \min \left\{ 1, \frac{w(\mathbf{U})}{w(\mathbf{u}^{(i-1)})} \right\}$ , siendo  $w(\mathbf{U}) = \frac{p(\mathbf{U})}{q(\mathbf{U})}$ , función de peso ya presentada en el método IS (3.3.2). Este método crea una dependencia entre las muestras, de modo que si el peso de una muestra  $\mathbf{u}^{(i)}$  es mayor que la anterior, se acepta y en caso contrario, se rechaza o se acepta según lo fuerte que sea dicha discrepancia.

### 3.4.3. Método de Gibbs

El método de simulación de Gibbs, frecuente en problemas multivariados, consiste en una implementación en serie de la cadena de Markov de 3.4.1 y unas funciones de muestreo auxiliar  $q(\cdot)$  determinadas.

Durante el proceso se actualiza cada una de las  $U_g$  de  $\mathbf{U} = \{U_1, U_2, \dots, U_G\}$  mediante una cadena distinta  $K\left(\mathbf{U}_g | \mathbf{u}_{1:g-1}^{(i)}, \mathbf{u}_{g+1:G}^{(i-1)}\right)$ , con una probabilidad de aceptación que se rige por la función  $p\left(\mathbf{U}_g | \mathbf{u}_{1:g-1}^{(i)}, \mathbf{u}_{g+1:G}^{(i-1)}\right)$  y funciones auxiliares  $q\left(\mathbf{U}_g | \mathbf{u}_{1:g-1}^{(i)}, \mathbf{u}_{g+1:G}^{(i-1)}\right)$ . Para cada subíndice  $g$ ,  $p(\cdot)$  y  $q(\cdot)$  coinciden, luego  $Pr\left(A | \mathbf{U}_k, \mathbf{u}_{1:g-1}^{(i)}, \mathbf{u}_{g+1:G}^{(i-1)}\right) = 1$  y, por tanto, el proceso de aceptación del método de Metropolis-Hastings no es necesario, pues se aceptan todas las muestras.

El pseudocódigo que aparece en la Figura 3.6 ilustra las diferentes etapas del método de Gibbs. Cada elemento del bucle interno se relaciona con una de las cadenas muestradas en serie. El muestreo de  $u_k^{(i)}$  de la función  $p\left(\mathbf{U}_g | \mathbf{u}_{1:g-1}^{(i)}, \mathbf{u}_{g+1:G}^{(i-1)}\right)$  hace que, como se ha comentado antes, desaparezca el proceso de aceptación típico de Metropolis-Hastings.

<pre> 1.   i   =   1,   j   =   1,   fijar <math>\mathbf{u}^{(0)} = \{u_1^{(0)}, u_2^{(0)}, \dots, u_G^{(0)}\}</math> 2.   while (<math>i \leq N</math>) 3.       g = 1 4.       while (<math>g \leq G</math>) 5.           <math>\mathbf{u}_g^{(i)} \sim p\left(\mathbf{U}_g   \mathbf{u}_{1:g-1}^{(i)}, \mathbf{u}_{g+1:G}^{(i-1)}\right)</math>,   g = g + 1 6.           if <math>j \geq B</math> then   i = i + 1 7.           else <math>\mathbf{u}^{(i-1)} = \mathbf{u}^{(i)}</math>,   j = j + 1 </pre>
---

Figura 3.6: Método de Gibbs.

A la hora de aplicar el método de Gibbs, la dificultad radica principalmente en determinar las funciones de distribución condicional  $p\left(\mathbf{U}_g | \mathbf{u}_{1:g-1}^{(i)}, \mathbf{u}_{g+1:G}^{(i-1)}\right)$  a partir de la función de probabilidad original  $p(\mathbf{U})$ . Sin embargo, esto se puede sortear parcialmente si se modela el problema con una **red bayesiana. (la explico, en anexo...?)**

### 3.5. Métodos secuenciales de Monte-Carlo

Los métodos secuenciales de Monte-Carlo, aplicables a problemas multivariantes, estos métodos hacen uso de una implementación secuencial del muestreo enfatizado combinado, a elección, con una implementación secuencial del remuestreo por pesos. Por medio de las independencias condicionales de las  $U_g \in \mathbf{U}$ , generan los  $N$  valores  $\mathbf{u}_g^{(i)}$  de  $U_g$ , teniendo en cuenta los valores de  $\mathbf{u}_k^{(i)}$  de las variables  $\{U_k | k \subset \{1 : G\}\}$  ya muestradas.

En el diseño del algoritmo interfieren las independencias condicionales propias del problema a resolver y, por tanto, su implementación se adapta a las características específicas de dicho problema. Dado que el método depende del problema, resulta útil escoger un tipo particular de problema para ejemplificar el funcionamiento de esta familia de técnicas.

Por ejemplo, para problemas modelables mediante redes bayesianas dinámicas genéricas, un conjunto de métodos secuenciales son los filtros de partículas (*Particle Filters, PF*).

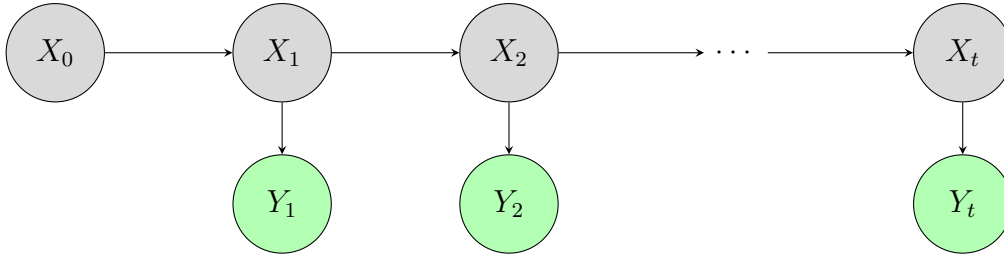


Figura 3.7: Red bayesiana para el desarrollo de los PF

En la red bayesiana de la Figura 3.7 se declara que el valor de la variable oculta  $X_t$  depende únicamente del valor de  $X_{t-1}$ , y que el valor de la variable observada  $Y_t$  solamente depende de  $X_t$ . El comportamiento de cada problema se modela según la elección de las funciones  $p(X_t|X_{t-1})$  y  $p(Y_t|X_t)$  para las dependencias anteriores. Sobre esta misma red se pueden definir varios problemas de estimación. En particular, para ejemplificar el funcionamiento

se escogerá estimar  $p(X_{0:t}|y_{1:t})$  por medio de  $p_N(X_{0:t}|y_{1:t}) = \sum_{i=1}^N w(x_{0:t}^{(i)}) \delta(X_{0:t} - x_{0:t}^{(i)})$ .

Este tipo de problemas se dan en diversas áreas del conocimiento: seguimiento de objetos, reconocimiento de habla, economía, etc [3]. Un ejemplo del primer tipo es la determinación de la trayectoria  $X_{0:t}$ , seguida por una partícula móvil, a partir de las medidas  $y_{1:t}$  captadas por uno o varios sensores.

### 3.5.1. Filtros de partículas

El elemento principal de los PF es una implementación secuencial del método de muestreo enfatizado (*Importance Sampling, IS*, 3.3.2). Para aproximar  $p(X_{0:t}|y_{1:t})$  se usa una función de muestreo auxiliar  $q(X_{0:t}|y_{1:t})$ , y se calcula el peso de  $x_{0:t}^{(i)} \sim q(X_{0:t}|y_{1:t})$  según

la expresión  $w(x_{0:t}^{(i)}) = \frac{p(x_{0:t}^{(i)}|y_{1:t})}{q(x_{0:t}^{(i)}|y_{1:t})}$ . Utilizando el *teorema de Bayes*, la regla de la cadena

y las relaciones de independencia condicional del problema, se manipulan las funciones de densidad para lograr la secuencialidad:

$$\begin{aligned}
 p(X_{0:t}|y_{1:t}) &= p(X_{0:t}, y_{1:t}) p(y_{1:t}) \propto \\
 &\propto p(y_t|X_{0:t}, y_{1:t-1}) p(X_t|X_{0:t-1}, y_{1:t-1}) p(X_{0:t-1}, y_{1:t-1}) \\
 &= p(y_t|X_t) p(X_t|X_{t-1}) p(X_{0:t-1}, y_{1:t-1}) p(y_{1:t-1}) \propto \\
 &\propto p(y_t|X_t) p(X_t|X_{t-1}) p(X_{0:t-1}, y_{1:t-1})
 \end{aligned} \tag{3.9}$$

$$\begin{aligned}
 q(X_{0:t}|y_{1:t}) &= q(X_t|X_{0:t-1}, y_{1:t}) q(X_{0:t-1}|y_{1:t}) \simeq \\
 &\simeq q(X_t|X_{0:t-1}, y_{1:t}) q(X_{0:t-1}|y_{1:t-1})
 \end{aligned} \tag{3.10}$$

Sustituyendo en la expresión de los pesos, se ve qué pesos en tiempos consecutivos están relacionados, se obtiene una función auxiliar  $q(\cdot)$  para el muestreo de  $x_t^{(i)}$  y las funciones

de densidad condicionales  $p(\cdot)$  que definen el comportamiento dinámico del problema en su totalidad:

$$\begin{aligned} w(x_{0:t}^{(i)}) &= \frac{p(x_{0:t}^{(i)}|y_{1:t})}{q(x_{0:t}^{(i)}|y_{1:t})} \simeq \frac{p(y_t|x_t^{(i)}) p(x_t^{(i)}|x_{t-1}^{(i)}) p(x_{0:t-1}^{(i)}|y_{1:t-1})}{q(x_t^{(i)}|x_{0:t-1}^{(i)}, y_{1:t}) q(x_{0:t-1}^{(i)}|y_{1:t-1})} = \\ &= \frac{p(y_t|x_t^{(i)}) p(x_t^{(i)}|x_{t-1}^{(i)})}{q(x_t^{(i)}|x_{0:t-1}^{(i)}, y_{1:t})} w(x_{0:t-1}^{(i)}) \end{aligned} \quad (3.11)$$

Lo anteriormente expuesto se recoge en el pseudocódigo, mostrando el PF normalmente denominado *Sequential Importance Sampling, SIS*. La generación secuencial de las muestras, es decir, la obtención de  $x_t^{(i)}$  correspondiente a la muestra  $x_{0:t}^{(i)}$ , es controlada por el bucle exterior de la Figura 3.8, línea 1, mientras que los valores asociados a las  $N$  muestras son controlados por el bucle interior. Las etapas opcionales muestran otras variantes, pero estos procesos no se realizan en el SIS.

Diferentes filtros de partículas tipo SIS surgen de la elección de distintas  $q(X_t|X_{0:t-1}, y_{1:t})$ . La elección óptima, sin tener en cuenta la función general  $f(X_{0:t})$ , es  $q^*(X_t|X_{0:t-1}, y_{1:t}) = p(X_t|X_{t-1}, y_t)$  [3]. Por lo general, no es posible generar muestras a partir de la misma, por lo que lo más sencillo es elegir  $q(X_t|X_{0:t-1}, y_{1:t}) = p(X_t|x_{t-1}^{(i)})$ , que figura en las primeras versiones de los PF tipo SIS, útil porque facilita la generación de muestras y los cálculos de los pesos  $w(x_{0:t}^{(i)}) = p(y_t|x_t^{(i)}) w(x_{0:t-1}^{(i)})$ .

Sin embargo, una elección distinta a la óptima hace que con el tiempo el número de muestras que toman un valor de peso nulo se incremente, provocando una degeneración de las muestras. Al no contribuir al cálculo integral, hace menos eficiente el algoritmo, por lo que se requiere de otras técnicas para compensarlo: se combina el SIS con etapas de remuestreo por pesos (*Weighted Resampling, WR*, 3.3.3).

Sobre el pseudocódigo de la Figura 3.8, en la etapa asociada a la parte *optionally, WR for SIR*, utiliza  $r(x_{0:t}^{(i)}) = w(x_{0:t}^{(i)})$ . En esta modalidad del filtro de partículas *Sampling Importance Resampling, SIR*, los pesos de las muestras tras el remuestreo son  $w(x_{0:t}^{(i)}) = \frac{1}{N}$ , y las muestras  $x_{0:t}^{(i)}$  se acumulan en regiones donde previamente los pesos toman valores no nulos. El problema que surge de aplicar esto, indiscriminadamente, es el contrario al anterior, el empobrecimiento. Ahora, la variable  $X_k$  tiende a tomar el mismo valor en todas las muestras  $x_{0:t}^{(i)}$ , para algún  $k < t$ . Para evitarlo, puede aplicarse solamente a las iteraciones  $t$  en las que se detecte la degeneración mediante algún proceso auxiliar como el cálculo del número efectivo de partículas.

Además, en la parte del pseudocódigo *optionally, WR for Auxiliary PF*, se utiliza  $r(x_{0:t-1}^{(i)}) = p(y_t|x_{0:t-1}^{(i)})$ , haciendo que el remuestreo considere las dependencias entre la nueva medida y los valores disponibles de la muestra. Calcular el valor de la función no es, en la mayoría de los casos, sencillo, de modo que se usa la aproximación 3.12.

```

1.   for ( $i = 1 : N$ )
2.       set  $x_0^{(i)}$ ,   set  $w(x_0^{(i)})$ 
3.    $t = 1$ 
4.   while (true)
5.        $F = 0$ 
6.       optionally, WR for Auxiliary PF
7.       for ( $i = 1 : N$ )
8.            $x_t^{(i)} \sim q(X_t | X_{0:t-1}, y_{1:t}), \quad \frac{p(y_t | x_t^{(i)}) p(x_t^{(i)} | x_{t-1}^{(i)})}{q(x_t^{(i)} | x_{0:t-1}^{(i)}, y_{1:t})} w(x_{0:t-1}^{(i)})$ 
9.            $F = F + w(x_{0:t}^{(i)})$ 
10.      for ( $i = 1 : N$ )
11.           $w(x_{0:t}^{(i)}) = \frac{w(x_{0:t-1}^{(i)})}{F}$ 
12.      optionally, WR por SIR
13.       $t = t + 1$ 

```

Figura 3.8: Filtro de partículas Sequential Importance Sampling (SIS) con las etapas opcionales que lo convierten en Sequential Importance Resampling (SIR) y Auxiliary Particle Filter (APF).

Esta etapa, WR, junto a la función de muestreo  $q(X_t | X_{0:t-1}, y_{1:t}) = p(X_t | x_{t-1}^{(i)})$ , se utilizan de manera auxiliar en el conocido como *Auxiliary PF (APF)*, por lo que los pesos de las partículas al finalizar cada iteración son  $w(x_{0:t}^{(i)}) \propto p(y_t | x_t^{(i)}) / p(y_t | m_t^{(i)})$ :

$$\begin{aligned}
 p(y_t | x_{0:t-1}^{(i)}) &= \int p(y_t | X_t) p(X_t | x_{t-1}^{(i)}) dX_t \simeq \\
 &\simeq p(y_t | m_t^{(i)}) p(X_t | x_{t-1}^{(i)}) dX_t = p(y_t | m_t^{(i)}); \quad \text{con } m_t^{(i)} = E_{p(X_t | x_{t-1}^{(i)})}[X_t]
 \end{aligned} \tag{3.12}$$

Todas las combinaciones vistas de  $q(X_t | X_{0:t-1}, y_{1:t})$  y etapas de WR generan distintos algoritmos secuenciales de simulación, con el objetivo final de generar las muestras en las zonas importantes del espacio, por lo que cualquier conocimiento previo que pueda facilitar la generación de las muestras en las regiones de interés puede ser una guía para la elección de  $q(X_t | X_{0:t-1}, y_{1:t})$  y de las funciones  $r(X_{0:t})$  para las etapas de WR.

### 3.6. Métodos de simulación generales

Los métodos introducidos hasta ahora, (RS, IS, WR, MC), pueden ser caracterizados por aproximar una función de densidad original  $p(\mathbf{U})$  mediante una auxiliar  $p_N(\mathbf{U})$  comprendida por múltiples  $p(\mathbf{U} - \mathbf{u}^{(i)})$ , obtenidas a partir de un muestreo realizado con las funciones de muestreo auxiliar  $q(\cdot)$ , y ponderadas por medio de  $w(\mathbf{u}^{(i)})$ . Sin embargo, se pueden destacar importantes discrepancias entre ellos:

1. Tipo de  $p(\mathbf{U})$ : para WR está compuesta por varias funciones de densidad puntual, mientras que en el resto es una densidad genérica.
2. Función de muestreo auxiliar  $q(\cdot)$ :
  - a) RS genera las muestras  $\mathbf{u}^{(i)}$  según una función genérica  $q(\mathbf{U})$ , y decide su aceptación o rechazo en función de la discrepancia entre  $p(\mathbf{u}^{(i)})$  y  $q(\mathbf{u}^{(i)})$ , por lo que su eficiencia la marca la cantidad de valores rechazados.
  - b) IS muestrea las  $\mathbf{u}^{(i)}$  por medio de  $q(\mathbf{U})$ , pero asigna un peso  $w(\mathbf{u}^{(i)})$  proporcional a la discrepancia anterior.
  - c) WR se asemeja a IS, solo que utiliza una función muestreo proporcional a una redistribución de las muestras según  $r(\mathbf{u}^{(i)})$ .
  - d) MC genera las muestras siguiendo la cadena  $K(\mathbf{U}|\mathbf{u}^{(i-1)})$ , con ciertas propiedades, rechazando las muestras anteriores a la convergencia.
3. Los pesos  $w(\mathbf{u}^{(i)})$  asociados a las funciones de densidad puntual  $p(\mathbf{U} - \mathbf{u}^{(i)})$ : en RS y MC todas las muestras tienen la misma importancia ( $w(\mathbf{u}^{(i)}) = N^{-1}$ ), mientras que en IS y WR, los pesos dependen de la función de distribución elegida. Por eso, en unos se ignora y en otros se trabaja con el par muestra-peso
4. Si el método funciona cuando desconocemos la expresión de  $p(\mathbf{U})$  pero se dispone de  $Cte \cdot p(\mathbf{U})$ : aplica a RS, IS, WR, pero no a MC, sus cadenas no necesariamente tienen por qué funcionar con ese múltiplo de la función de densidad.

Adicionalmente, también se debe tener en cuenta la paralización de cada uno de los cuatro métodos, pues no todos generan las muestras a la misma velocidad. En IS y WR, el valor de las muestras y de los pesos sin normalizar se realiza de forma independiente, de modo que puede hacerse de manera simultánea. RS funciona de manera similar, pues el valor de cada muestra y su aceptación o rechazo se realiza de forma independiente, salvo que discrepa en la cantidad de muestras generadas, ya que depende de cuántas haya rechazado, hasta llegar a  $N$ . Finalmente, se recuerda que en MC las cadenas generan valores para la muestra de manera condicionada al anterior valor,  $K(\mathbf{U}|\mathbf{u}^{(i-1)})$ , de modo que es necesario hacerlo de manera secuencial. Por otro lado, también es posible generar de forma simultánea varios conjuntos de muestras mediante varias cadenas de Markov ejecutadas en paralelo, de forma independiente.

Se recuerda también que en IS la función óptima de muestreo es proporcional a  $|f(\mathbf{U})|$ , por lo que es inútil generar puntos en regiones donde dicha función sea nula. Esto se puede extender al resto de métodos, pues tales puntos no aportan valor a la aproximación integral.

Para finalizar, se contrastan los métodos RS frente a una combinación de una etapa de IS con una de WR, donde  $r(\mathbf{u}^{(i)})$  sean idénticas a los de la función original  $\sum \alpha^{(i)} \delta(\mathbf{U} - \mathbf{u}^{(i)})$ . En RS, la generación de las posibles muestras  $\mathbf{u}^{(i)}$  generadas con  $q(\mathbf{U})$  se da en la etapa de muestreo de IS con  $q(\mathbf{U})$ . Los pesos  $w(\mathbf{u}^{(i)}) = \frac{p(\mathbf{u}^{(i)})}{q(\mathbf{u}^{(i)})}$  en IS para las muestras son proporcionales a  $\Pr(A|\mathbf{u}^{(i)}) = \frac{p(\mathbf{u}^{(i)})}{Mq(\mathbf{u}^{(i)})}$  en el proceso de aceptación/rechazo para cada una de las muestras de RS. En la última etapa, mientras que RS hace el proceso de aceptación/rechazo de la muestra de forma independiente a las demás, el uso de una etapa de WR tras una de IS con  $r(\mathbf{u}^{(i)}) = \frac{p(\mathbf{u}^{(i)})}{q(\mathbf{u}^{(i)})}$  hace que las muestras admitidas (y posiblemente repetidas) sean aquellas con valores  $\frac{r(\mathbf{u}^{(i)})}{\sum r(\mathbf{u}^{(i)})}$  más significativos. Tras la etapa de WR, los pesos de las  $N$  muestras son equiponderantes, y el mismo valor se les asigna tras la etapa de RS.

Ambos métodos se comportan de forma parecida para  $N \rightarrow \infty$  debido a que, en RS, sólo se usa un valor  $M$  a lo largo de todo el soporte de  $p(\mathbf{U})$ , al igual que su análogo  $\sum_{i=1}^N r(\mathbf{u}^{(i)})$  en WR.

## Capítulo 4

# Técnicas de simulación estocástica en algoritmos de optimización heurística

Este capítulo se centra en la utilización de los métodos de Monte Carlo para abordar problemas de optimización desde una perspectiva heurística. Un ejemplo representativo es el célebre problema del vendedor ambulante o del viajero (*Traveling Salesman Problem*, *TSP*), donde la explosión combinatoria desempeña un papel crucial. Los algoritmos deterministas disponibles hasta el momento se caracterizan por su gran complejidad y falta de generalidad. Además, existen otros tipos de problemas que presentan dificultades inherentes debido a la presencia de numerosos mínimos (máximos) locales, los cuales requieren un examen exhaustivo. Aunque es posible explorarlos de manera eficiente, persiste la incertidumbre en cuanto a la adecuada transferencia de estos hallazgos para identificar el óptimo global.

Así, es necesario apoyarse en los métodos de Monte-Carlo, los cuales son sencillos de programar y ofrecen un muy buen rendimiento computacional. Caben destacar los algoritmos genéticos (Algorithms, GA), GA) y el temple simulado (*Simulated Annealing*), los cuales se estudiarán más en profundidad en la sección 4.1

El código completo está disponible en el repositorio de Github: <https://github.com/mavice07/TFG-Mates.git>.

### 4.1. Métodos de optimización de Monte-Carlo

El método de temple simulado se basa en la experiencia previa de la mecánica estadística, a veces referida como termodinámica estadística, en la que se estudian las propiedades macroscópicas de sistemas que tienen un enorme número de estados. Este campo enlaza las propiedades macroscópicas de los materiales en equilibrio termodinámico y los comportamientos y movimientos microscópicos que ocurren dentro del material.

Las moléculas de un material pueden tener distintos niveles de energía, siendo el menor de



ellos el estado fundamental. A mayor temperatura, mayor es la probabilidad de encontrar moléculas a un nivel más alto de energía, configurándose en distintos microestados. El conjunto de todos los posibles microestados se representará como  $\Omega$ , y  $n_i$  hará referencia al número de partículas en el estado  $i$ .

Disminuir la temperatura de una sustancia hasta el cero absoluto no garantiza que ésta logre su estado de energía más bajo. Por tanto, se usa la técnica de templado, que consiste en elevar la temperatura del ambiente, al menos hasta conseguir que el sólido se funda, para después bajar la temperatura de manera gradual hasta que las partículas se estabilicen, volviendo al estado sólido (*congelación*). El proceso contrario del templado se conoce como oscurecimiento (*quenching*) en el que se hace descender la temperatura del ambiente instantáneamente. Si no se aplica un enfriamiento suave, el material alcanza una estructura meta-estable con mayor valor energético.

La técnica del templado es el mejor procedimiento para evitar estancarse en un mínimo local de energía y alcanzar uno global, ya que, conforme la temperatura disminuye, también lo hace la probabilidad de aceptación de una solución peor que la actual.

El equilibrio térmico está caracterizado por la *ecuación de Boltzmann*. Esto es, la probabilidad de que un material alcance un estado  $i \in \Omega$  con energía  $E_i$  a la temperatura  $T$ .

$$P_T(X = i) = \frac{1}{Z(T)} e^{\frac{-E_i}{k_B T}} \quad \text{con} \quad Z(T) = \sum_{j \in \Omega} e^{\frac{-E_j}{k_B T}} \quad (4.1)$$

$X$  representa la variable aleatoria que indica el estado del sólido,  $T$  es la temperatura de ambiente,  $k_B$  es la constante de Boltzmann y  $Z(T)$  es la función de partición. Hay que observar que su suma se aplica a todos los estados de  $\Omega$ .

La simulación de este fenómeno físico viene dado a través del algoritmo de Metrópolis, entre otros, generando los estados de la manera siguiente:

- Se tiene un estado del material  $i$ , con energía  $E_i$ .
- Tras alterar el estado actual, se genera al estado  $j$  con energía  $E_j$ .
- Se transiciona del primero al segundo si  $E_i - E_j \geq 0$ .
- Se transiciona del primero al segundo con una probabilidad  $P = e^{\frac{E_i - E_j}{k_B T}}$ , si se da que  $E_i - E_j \leq 0$ .

Este criterio de aceptación se conoce como *criterio de Metrópolis*. Más tarde se estableció un paralelismo entre el fenómeno físico y la optimización:

<b>Simulación termodinámica</b>	<b>Optimización</b>
Estados del material	Soluciones del problema
Energía	Función objetivo
Cambios de estado	Soluciones vecinas
Estados metaestables	Mínimo local
Estado de congelación	Solución heurística

Cuadro 4.1: Paralelismo entre simulación termodinámica y optimización.

Se define también el criterio de aceptación para un problema mono-objetivo de minimización, que jugará el papel del de Metrópolis, que da la probabilidad de aceptar una solución  $x$  generada a partir de las vecindades de una solución  $x^*$ . Si definimos  $\Delta f(x, x^*) = f(x) - f(x^*)$ :

$$P(\text{aceptar } x \text{ a partir de } x^*) = \begin{cases} 1 & \text{si } f(x) \leq f(x^*) \\ e^{\frac{-\Delta f(x, x^*)}{c}} & \text{si } f(x) > f(x^*) \end{cases} \quad (4.2)$$

El parámetro  $c \in \mathbb{R}^+$ , análogo a la temperatura, debe reducirse lentamente para que se dé el efecto deseado. Dichas estrategias se denominan *planes de templado*. Esto se puede traducir en una matriz  $g$  de dimensión  $|\Omega| \times |\Omega|$ , con las propiedades de irreducibilidad ( $x_0 = x^*, x_1, x_2, \dots, x_n = x$ ,  $n$  finito, si  $g_{x_{i-1}, x_i} \geq 0$ ) y simetría ( $g_{x_i x_j} = g_{x_j x_i}$ ). Dado que  $c$  varía en cada iteración, la ecuación 4.2 hará que las probabilidades de transición de la cadena de Markov varíen con el tiempo, generando así una de tipo no homogéneo.

Por tanto, un algoritmo de búsqueda local (2.3.1) puede transformarse en uno de templado al seleccionar aleatoriamente una solución de las vecindades en cada iteración  $k$ , y utilizar el criterio de aceptación para determinar si se reemplaza la solución actual por la del entorno seleccionada, considerando el valor del parámetro de control,  $c$ , en esa iteración. Esto se repetirá en cada iteración un número suficiente de veces (ciclo de procesamiento,  $L$ ) como para alcanzar el equilibrio térmico para cada una de las temperaturas  $c$ .

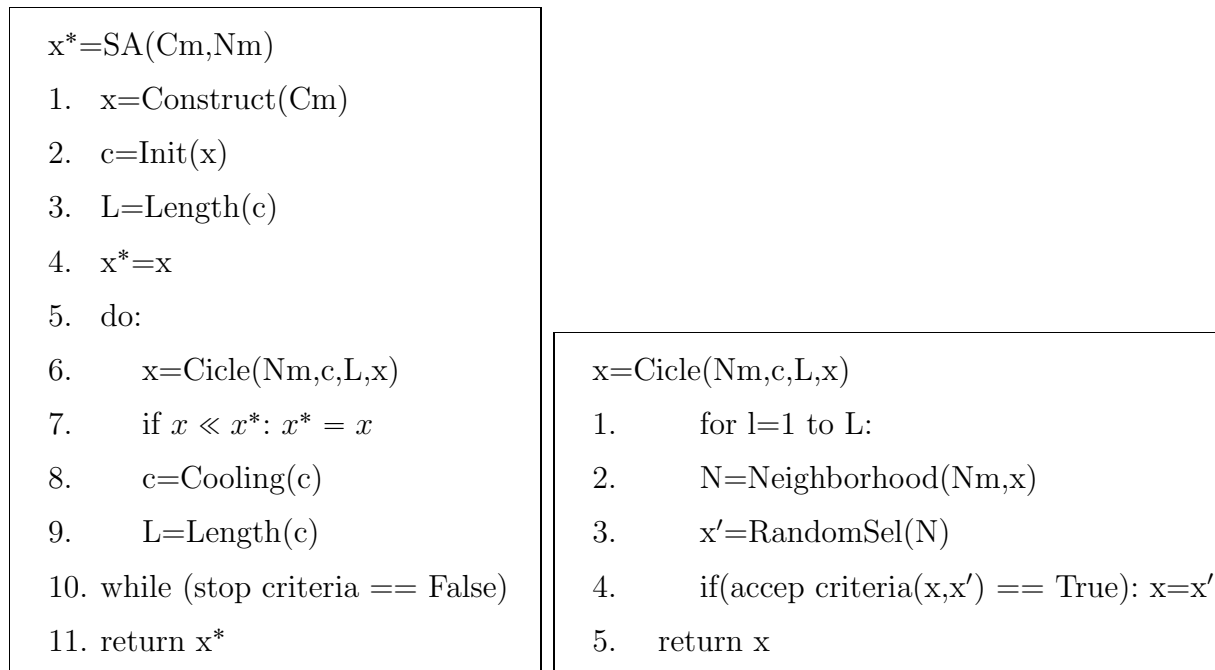


Figura 4.1: Temple simulado.

Figura 4.2: Ciclo de procesamiento.

El permitir aceptar soluciones peores a la actual (línea 4. Figura 4.2) es la clave para que, a diferencia de los de búsqueda local, el SA no quede atrapado en óptimos locales. Según disminuya  $c$ , menor será la probabilidad de aceptar dicha solución. De hecho, cuando  $c = 0$ , se ve claramente que SA es una generalización de la búsqueda monótona.

El algoritmo SA converge con una rapidez sujeta tanto al número de transformaciones (generación de una solución vecina y la aplicación del criterio de aceptación) para cada temperatura  $c$ , como por la forma en la que aminora (enfía) este parámetro. Teóricamente, esta convergencia al óptimo sucede de manera asintótica. Sin embargo, en la práctica es inviable ejecutar tal número de iteraciones, de modo que se rebajan sustancialmente.

Para ello, se sigue un plan de templado determinando un valor inicial para el parámetro de control  $c$ , una función de enfriamiento, un valor  $L$  de transiciones en cada valor de  $c$ , y un valor final de  $c$  como criterio de parada.

Valor inicial del parámetro  $c$ : siguiendo la analogía termodinámica, se requiere un valor alto para permitir aceptar cambios sustanciales. Una propuesta para  $c_0$ , valor inicial de  $c$ , en la línea 2 del algoritmo 4.1, es igualarlo a la diferencia del mayor y menor valor de la función de coste en las vecindades de la solución inicial, es decir,  $c_0 = f_{max} - f_{min}$ . Otra propuesta es comenzar con un  $c_0$  pequeño y calcular a través de iteraciones el llamado radio de aceptación  $\chi$  para luego multiplicar  $c_0$  por una constante mayor que 1 y volver a calcular  $\chi$ . Esto se repite hasta que el radio es próximo a 1. Otra posibilidad más es el esquema de enfriamiento en tiempo polinomial. Si  $m_1$  representan las iteraciones aceptadas,  $m_2$  las iteraciones rechazadas y  $\Delta f^+$  la media de diferencias de energía de  $m_2$ , el valor inicial de  $c = c_0$  se calcula ahora como

$$\chi = \frac{m_1 + m_2 \cdot e^{-\frac{\Delta f^+}{c}}}{m_1 + m_2} \implies c = \frac{\Delta f^+}{\ln \left( \frac{m_2}{m_2 \cdot \chi - m_1 \cdot (1 - \chi)} \right)}$$

estableciendo el valor de  $\chi$  cercano a 1, habiendo medido el resto de parámetros.

Función de enfriamiento: relativo a la línea 8 del algoritmo 4.1, el método para conseguir la suavidad del descenso de temperatura se obtiene principalmente multiplicando  $c$  por una constante  $\alpha \in [0.80, 0.99]$ , la velocidad de enfriamiento, de modo que  $c_{k+1} = \alpha \cdot c_k$ . Menos extendidas pero también válidas son la opción  $c_{k+1} = \frac{c_k}{1+\beta}$ , con  $\beta \in [0.1, 0.3]$ , la variante adaptativa  $c_{k+1} = \frac{c_k}{1+\beta}$  si el movimiento ha sido aceptado mientras que si no lo fue, se realiza  $c_{k+1} = \frac{c_k}{1+\gamma}$ , con  $\beta = \gamma \cdot r$ ; o la estrategia de enfriamiento en tiempo polinomial,  $c_{k+1} = \frac{c_k}{1 + \frac{c_k \cdot \ln(1+\delta)}{3\sigma_c}}$ , donde  $\delta$  denota la distancia y hace decrecer el parámetro  $c$ , y donde  $\sigma_c$  denota la desviación estándar de la función objetivo al parámetro  $c$ .

Longitud del ciclo de procesamiento: haciendo referencia a las líneas 3 y 9 de 4.1, se trata del número  $L$  de iteraciones en cada ciclo y se ve influida de forma proporcional por la función de enfriamiento, es decir, un descenso más pronunciado de la temperatura permite ciclos más largos. Aún así, la longitud crece según decrece  $c$ , pues se requieren más intentos por escapar de mínimos locales. Para cada temperatura, la estrategia puede ser bien fijar un valor manualmente y estudiar su convergencia para decidir si mantenerlo o cambiarlo, bien establecer una relación explícita con la función de enfriamiento o bien no estipular un valor base, sino realizar las iteraciones necesarias para que se acepten un número concreto de cambios. Con esta estrategia, al aceptarse más cambios cuando  $c$  se acerca a cero,  $L$  tenderá a infinito, luego es necesario establecer una cota superior.

Criterio de parada: referenciando a la línea 10 del algoritmo 4.1, a diferencia del proceso

termodinámico, no es necesario llegar al estado sólido, sino que el algoritmo puede parar antes debido al uso de probabilidades, ya muy cercanas a cero antes de que se dé la analogía con la congelación. Los distintos criterios de parada posibles pueden atender a la baja probabilidad de aceptación supeditada al valor de  $c$ , o a razón del número de iteraciones sin producirse un cambio. En ocasiones, este número se toma como ciclo de procesamiento. No obstante, es posible que estos dos criterios puedan tomar demasiadas iteraciones e incluso ciclos enteros en un entorno de un mínimo local, evitando así la congelación pero sin salir de dicho entorno. Por tanto, se proponen otros dos criterios que consisten en finalizar el algoritmo cuando los desplazamientos aceptados descienden de una determinada proporción o en recalentar una vez alcanzada la congelación y detenerlo tras unas ciertas iteraciones sin que se hayan producido cambios.

## 4.2. Algoritmos genéticos

El concepto de algoritmo genético (*Genetic Algorithm*, *GA*), también conocido como evolutivo, enmarca un conjunto de técnicas bioinspiradas, estableciendo clásicamente un paralelismo con la teoría darwiniana. Se establece una analogía entre iteraciones y generaciones, población (conjunto de soluciones) e individuos de una especie, mejorando estos hacia la mejor solución a través de selección (selección natural) junto a operadores genéticos (crce y mutación).

### 4.2.0.1. Tipos de aprendizaje

Se distinguen tres tipos de inteligencia en los seres vivos, en los cuales se inspiran estos algoritmos:

Filogenética:

## 4.3. Caso práctico: problema de rutas de vehículos

# Capítulo 5

## Conclusiones y Trabajo Futuro

El presente capítulo pretende dar una visión global de las conclusiones a las que se ha llegado durante el desarrollo de esta memoria. Esto nos servirá para poder evaluar las contribuciones personales y el impacto de los resultados obtenidos, y, al mismo tiempo, descubrir las líneas de investigación y trabajo que quedan abiertas, y que podrían ser la base de nuevos proyectos en el futuro.

### 5.1. Contribuciones

### 5.2. Conclusiones

### 5.3. Trabajos relacionados

A continuación, presentaremos brevemente algunos trabajos relacionados con el nuestro. Estos trabajos han servido, desde el inicio, para obtener inspiración en el punto de partida, para usarlos como comparación, o incluso como propuestas de lecturas complementarias a lo largo de todo este trabajo.

- Trabajo1
- Trabajo2
- Trabajo3

### 5.4. Trabajo futuro

En cuanto a las líneas de investigación que quedan abiertas, y sobre las que se podrían desarrollar nuevos trabajos en el futuro, destacan las siguientes:

- Trabajo1:
- Trabajo2:

# Bibliografía

- [1] Vicente Quesada Paloma y Alfonso García Pérez. *Lecciones de cálculo de probabilidades*. Ediciones Díaz de Santos, 1988.
- [2] Sheldon M Ross. *Simulation*. academic press, 2022.
- [3] MG Pajares y J De La Cruz. “Aprendizaje automático un enfoque práctico”. En: *RA-MA SA Editorial y Publicaciones. ISBN-10: 8499640117* (2010).

# Anexos

## Info Anexo 1

## Info anexo 2



## **Info anexo 3**