

Metaheurísticas

Unidad 6
Metaheurísticas Paralelas

Tema 1: Introducción

Objetivos

- Entender el concepto de las metaheurísticas paralelas
- Conocer los elementos más importantes en el diseño de este tipo de metaheurísticas
- Analizar distintas herramientas y aplicaciones reales que estén utilizando estas técnicas

Bibliografía

- [Alb05] E. Alba (ed.), “Parallel Metaheuristics”, John Wiley & Sons, 2005
- [Cra03] T.G. Crainic, M. Toulouse, Chap. 17: “Parallel Strategies for Metaheuristics.” In F. Glover, G.A. Kochenberber (Eds.) Handbook of Metaheuristics, Kluwer Academic Publisher (2003)
- [Cra09] T.G. Crainic, M. Toulouse, “Parallel Meta-Heuristics” Technical Report (<http://www.cirrelt.ca/DocumentsTravail/CIRRELT-2009-22.pdf>)

Motivación

- La alta complejidad en algunos problemas provocan una pérdida de eficiencia, o incluso, la imposibilidad de obtener resultados en tiempos razonables
- Necesidad de obtener soluciones “de mejor calidad” sin incrementar el tiempo:
 - Incrementamos iteraciones = evaluaciones
 - Mayor diversidad

Índice

1. Sistemas de cálculos paralelos
2. Programación paralela/distribuida
3. Clasificación de heurísticas paralelas
4. MapReduce

Sistemas de cálculos paralelos

Las arquitecturas se han construído sobre el concepto de von Neumann

- La Ley de Moore y su avance en el futuro (<https://hardzone.es/reportajes/que-es/ley-de-moore/>)
- Intentamos minimizar esta ley mediante paralelización

Sistemas de cálculos paralelos

Emplean supercomputadores

<https://www.top500.org>

The main objective of the TOP500 list is to provide a ranked list of general purpose systems that are in common use for high end applications. The authors of the TOP500 reserve the right to independently verify submitted LINPACK results, and exclude systems from the list which are not valid or not general purpose in nature. By general purpose system we mean that the computer system must be able to be used to solve a range of scientific problems. Any system designed specifically to solve the LINPACK benchmark problem or have as its major purpose the goal of a high TOP500 ranking will be disqualified

Sistemas de cálculos paralelos

tendencias de sistemas de cálculos

- En arquitecturas de microprocesadores
 - ★ multihebra
 - ★ multinúcleo
 - ★ combinados como i7
- En arquitecturas de sistemas
 - ★ Multiprocesador
 - ★ Cluster
- Arquitecturas específicas como vectoriales

Sistemas de cálculos paralelos

cálculo paralelo frente a distribuido

- Paralelismo real lleva implícito dos ejecuciones al mismo tiempo, **NO TIEMPO COMPARTIDO**
- Puede ser total (núcleos independientes) o parcial (hebras)
- Estrictamente, el procesamiento paralelo utilizan memoria compartida. Tienen una comunicación rápida
- Procesamiento distribuido emplean una memoria distribuida (nodos interconectados en red). Comunicación más lenta

Programación paralela/distribuida

- El procesamiento paralelo/distribuido podemos decir que **varios procesos** trabajan de forma simultánea en **procesadores independientes**
- Paralelismo lleva asociado:
 - Descomposición de la carga
 - Distribución de los recursos

Programación paralela/distribuida

Los aspectos más destacados en el diseño de este tipo de algoritmos:

- Dependencias temporales entre los datos
- Acceso en exclusión mutua a datos compartidos

Para calcular los rendimientos volvemos a utilizar los conceptos de la Arquitectura de Computadores

$$\text{ganancia} = \frac{t_{\text{secuencial}}}{t_{\text{paralelo}}}$$

Clasificación de heurísticas paralelas

Entornos

Sólo mantienen una solución actual en cada momento

Poblaciones

Mantienen un conjunto de soluciones en cada momento

Clasificación de heurísticas paralelas

Crainic y Nourredine

Las metaheurísticas paralelas se denotan por tres dimensiones:

- A. Cardinalidad del proceso de búsqueda
- B. Intercambio de información de procesos
- C. Diversidad o variedad de los métodos de búsqueda empleados

Clasificación de heurísticas paralelas

CyN – Cardinalidad del proceso de búsqueda

1C

El control del sistema se realiza únicamente por un único proceso
Un proceso monitoriza

pC

El control del sistema se realiza por varios procesos
Control distribuido entre varios procesos independientes

Clasificación de heurísticas paralelas

CyN - Comunicación y control de búsqueda

Síncrona

Existen esperas entre los interlocutores

Asíncrona

Sin esperas, necesita almacenamiento temporal de la información recibida y no procesada

Clasificación de heurísticas paralelas

CyN – Diversidad de los métodos de búsqueda

SPSS

Same initial **Point/Population**
Same search **Strategy**

MPSS

Multiple initial **Point/Population**
Same search **Strategy**

SPDS

Same initial **Point/Population**
Different search **Strategy**

MPDS

Multiple initial **Point/Population**
Different search **Strategy**

Clasificación de heurísticas paralelas

Crainic y Nourredine



Clasificación de heurísticas paralelas

Crainic y Nourredine

Grupo 1: Descomposición funcional de bajo nivel

Clasificación de heurísticas paralelas

Crainic y Nourredine

Grupo 1: Descomposición funcional de bajo nivel

Grupo 2: Descomposición de dominio

Clasificación de heurísticas paralelas

Crainic y Nourredine

Grupo 1: Descomposición funcional de bajo nivel

Grupo 2: Descomposición de dominio

Grupo 3: Múltiples búsquedas independientes

Clasificación de heurísticas paralelas

Crainic y Nourredine

Grupo 1: Descomposición funcional de bajo nivel

Grupo 2: Descomposición de dominio

Grupo 3: Múltiples búsquedas independientes

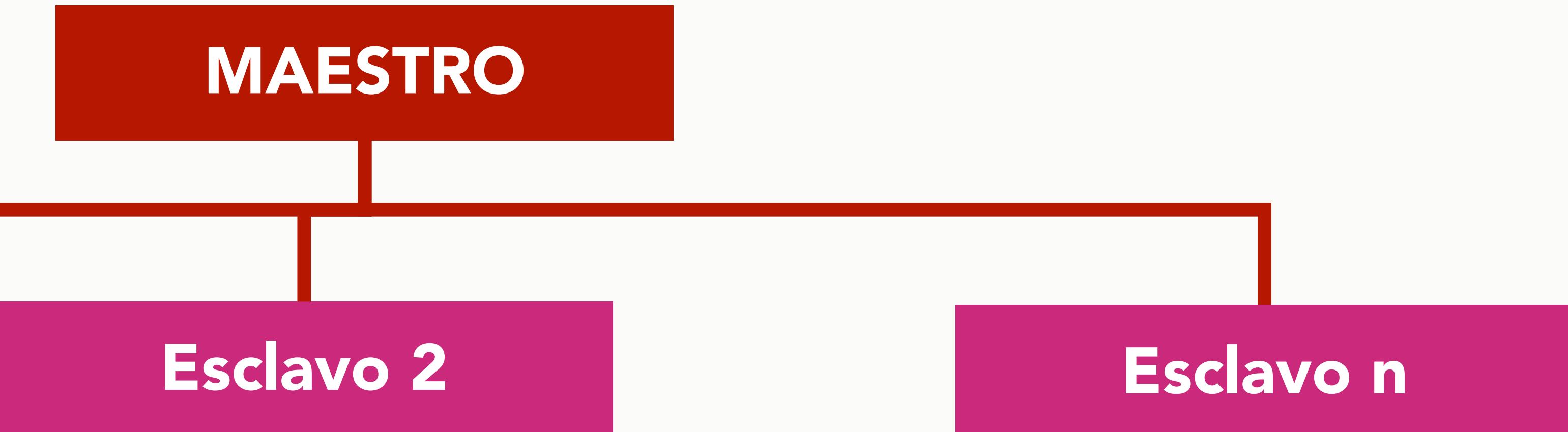
Grupo 4: Múltiples búsquedas cooperativas

Clasificación de heurísticas paralelas

Crainic y Nourredine

Grupo 1: Descomposición funcional de bajo nivel

Modelo 1C/RS/SPSS



Clasificación de heurísticas paralelas

Crainic y Nourredine

Grupo 2: Descomposición de dominio

Espacio 1

Espacio 2

Espacio n

Modelo 1C/KS/MPSS

Modelo 1C/KS/MPDS

Clasificación de heurísticas paralelas

Crainic y Nourredine

Ha obtenido buenos resultados en técnicas multiarranque a pesar de no intercambiar conocimiento entre ejecuciones

Modelo pC/RS/*

Grupo 3: Múltiples búsquedas independientes

Clasificación de heurísticas paralelas

Crainic y Nourredine

- Intercambia información durante el proceso de búsqueda
- Mejores soluciones que el anterior grupo
- Intercambio de información es clave

Grupo 4: Múltiples búsquedas cooperativas

Clasificación de heurísticas paralelas

múltiples búsquedas cooperativas

Hay que satisfacer diferentes cuestiones:

- ¿Qué información intercambiar?
- ¿Entre qué procesos intercambiar?
- ¿Cuándo?
- ¿Cómo?
- ¿Cómo usar esta información?

Clasificación de heurísticas paralelas

múltiples búsquedas cooperativas - ¿qué **intercambiar**?

Lo más sencillo es **enviar la mejor solución encontrada hasta el momento**

Aunque intercambiar solo la mejor solución puede provocar pérdida de diversidad

Interesante **transferir el conocimiento adquirido en el proceso de búsqueda** como las memorias adaptativas

Clasificación de heurísticas paralelas

múltiples búsquedas cooperativas - ¿entre qué procesos?

Podemos utilizar distintos tipos de comunicación como
síncrona o asíncrona

Se pueden utilizar repositorios centrales o distribuidos (*pools*,
data warehouse) mediante una comunicación asíncrona

Clasificación de heurísticas paralelas

múltiples búsquedas cooperativas - **¿cuándo y cómo?**

Hay que tener cuidado con los posibles retrasos que pueden provocar los intercambios

Equilibrio entre eficiencia y eficacia

- Cooperación síncrona: conseguir información completa del proceso de búsqueda global
- Cooperación asíncrona: totalmente distribuida, más flexible y permite el desarrollo de un comportamiento emergente más efectivo

Clasificación de heurísticas paralelas

múltiples búsquedas cooperativas en **Poblaciones**

Tienen un buen funcionamiento en cualquiera de los modelos vistos hasta ahora

Especial interés en las múltiples búsquedas cooperativas

**Tipo de
descentralización**

Modelos celulares

Modelos distribuidos

**Relación entre
ambos modelos**

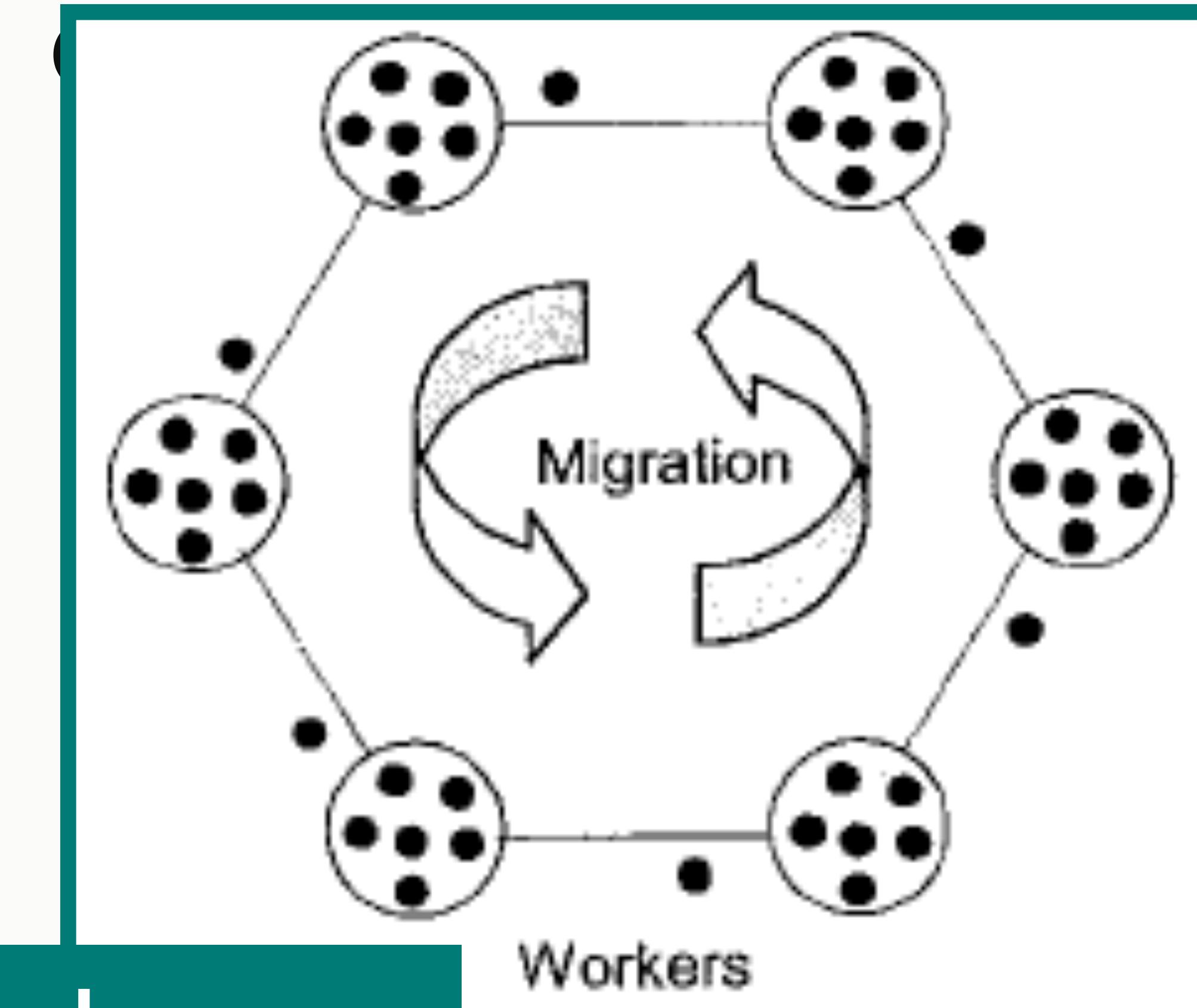
Clasificación de heurísticas paralelas

múltiples búsquedas cooperativas en **Poblaciones**

Distribuidos

- Definen subpoblaciones
- Comunicación de intercambio de individuos

Tipo de
descentralización

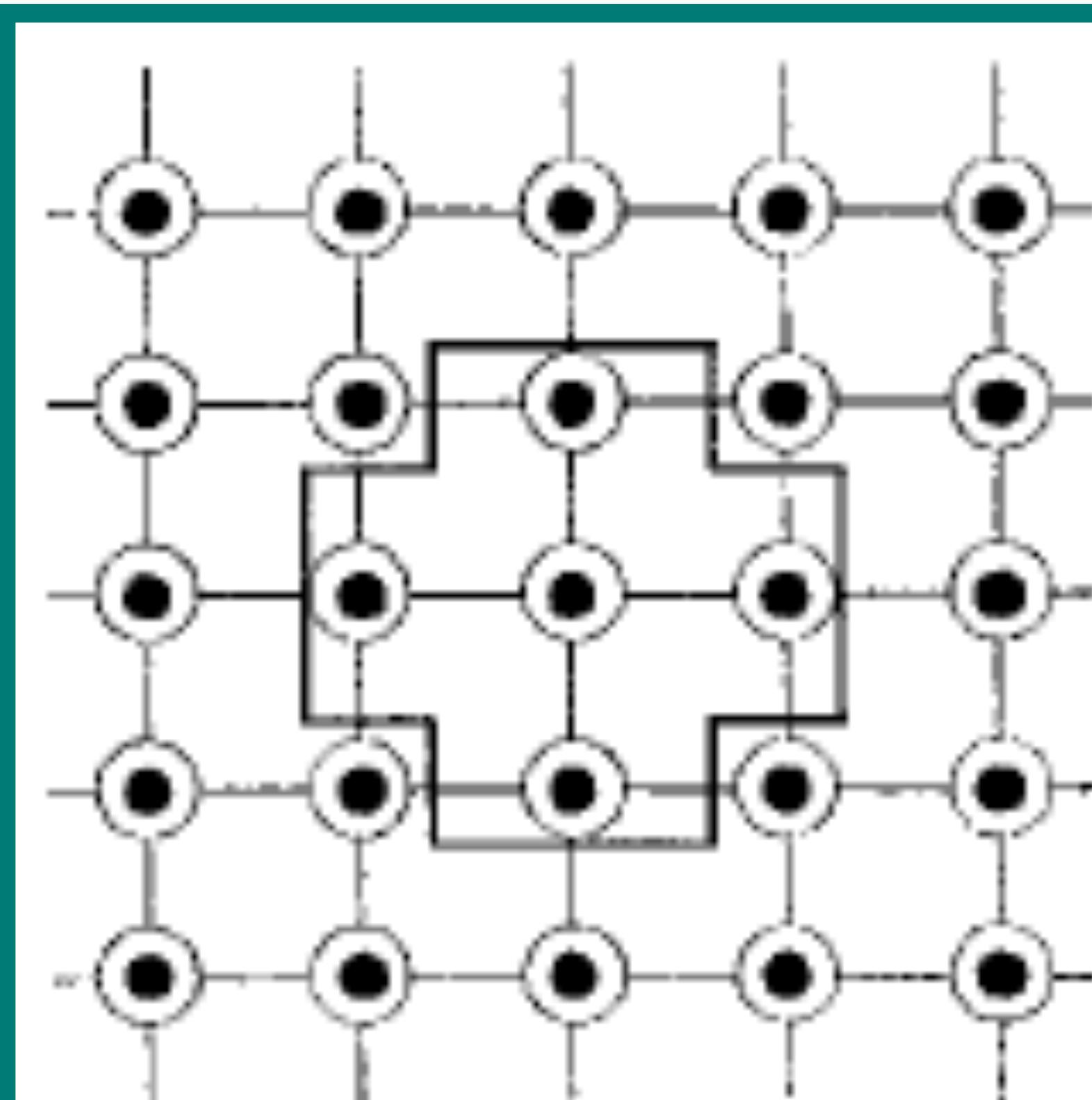


Clasificación de heurísticas paralelas

múltiples búsquedas cooperativas en **Poblaciones**

Distri

- Definidas por la población
- Comunicación entre individuos



Workers

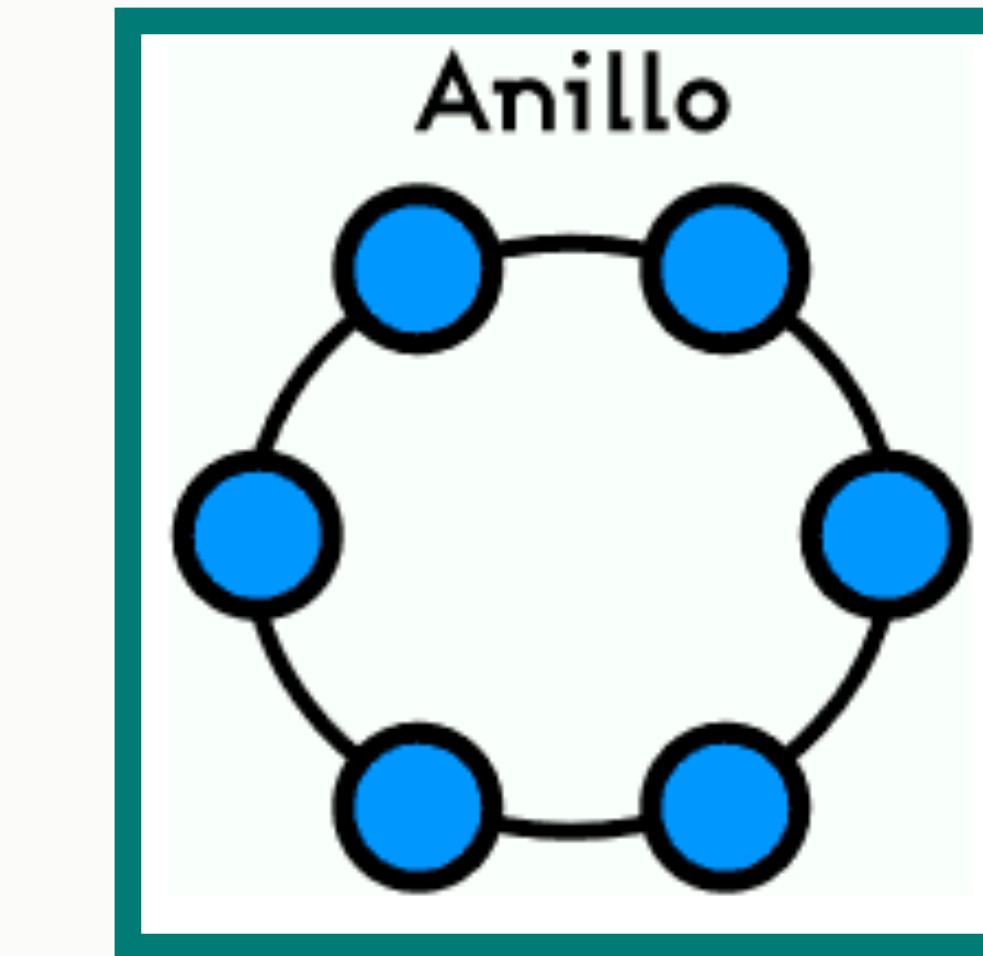
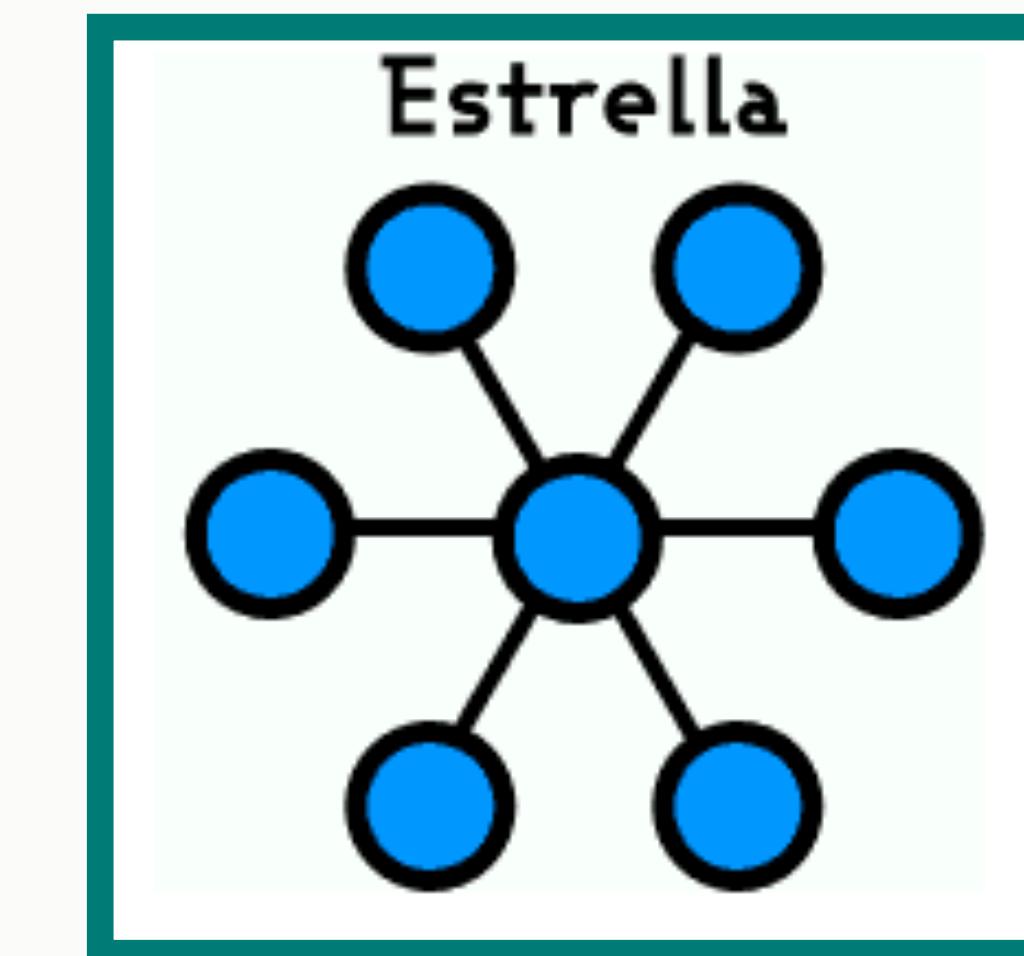
Tipo de
descentralización

Clasificación de heurísticas paralelas

múltiples búsquedas cooperativas en **Poblaciones**

El diseño de la estructura de intercomunicación entre individuos es clave

- Estrella
- Anillo
- Red



Modelos distribuidos

Clasificación de heurísticas paralelas

múltiples búsquedas cooperativas en **Poblaciones**

Paralelizar las operaciones de la población pero con individuos vecinos de la propia celda

- Evaluación
- Cruce
- Mutación

Modelos celulares

El paradigma Map Reduce

MapReduce

Surge en paralelo al problema Big Data

Explorar 100 TB en 1 nodo @ 50MB/sec equivalen a 23 días

Explorar 100 TB en 1 clúster de 1000 nodos equivalen a 33 minutos

MapReduce

Surge en paralelo al problema Big Data

Explorar 100 TB en 1 nodo @ 50MB/sec equivalen a 23 días

Explorar 100 TB en 1 clúster de 1000 nodos equivalen a 33 minutos

Divide y Vencerás

MapReduce

divide y vencerás

Entonces, ¿aumentamos los recursos de hardware y solucionamos el problema?

¿Cuánta energía podemos necesitar y espacio?

¿Cuánto cuesta mantener eso?

Google lanza el Paradigma MAPREDUCE en 2004

MapReduce

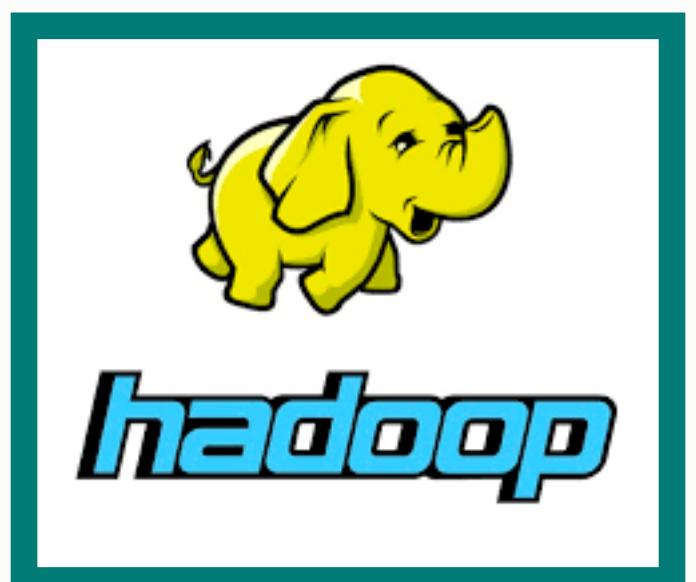
Llegar a procesar en sus comienzos unos 20 PB por día

Es un modelo de programación de datos paralela

Concepto simple, elegante y extensible para múltiples aplicaciones

Popularizó por el proyecto de código abierto Hadoop

<https://hadoop.apache.org/>



MapReduce

- Oculta la complejidad de la distribución y tolerancia a fallos
- Gestión de comunicación entre máquina
- Completamente transparente para el programador, analista y usuario
- MapReduce no es adecuado para todos los problemas, pero cuando funciona, puede ahorrar mucho tiempo

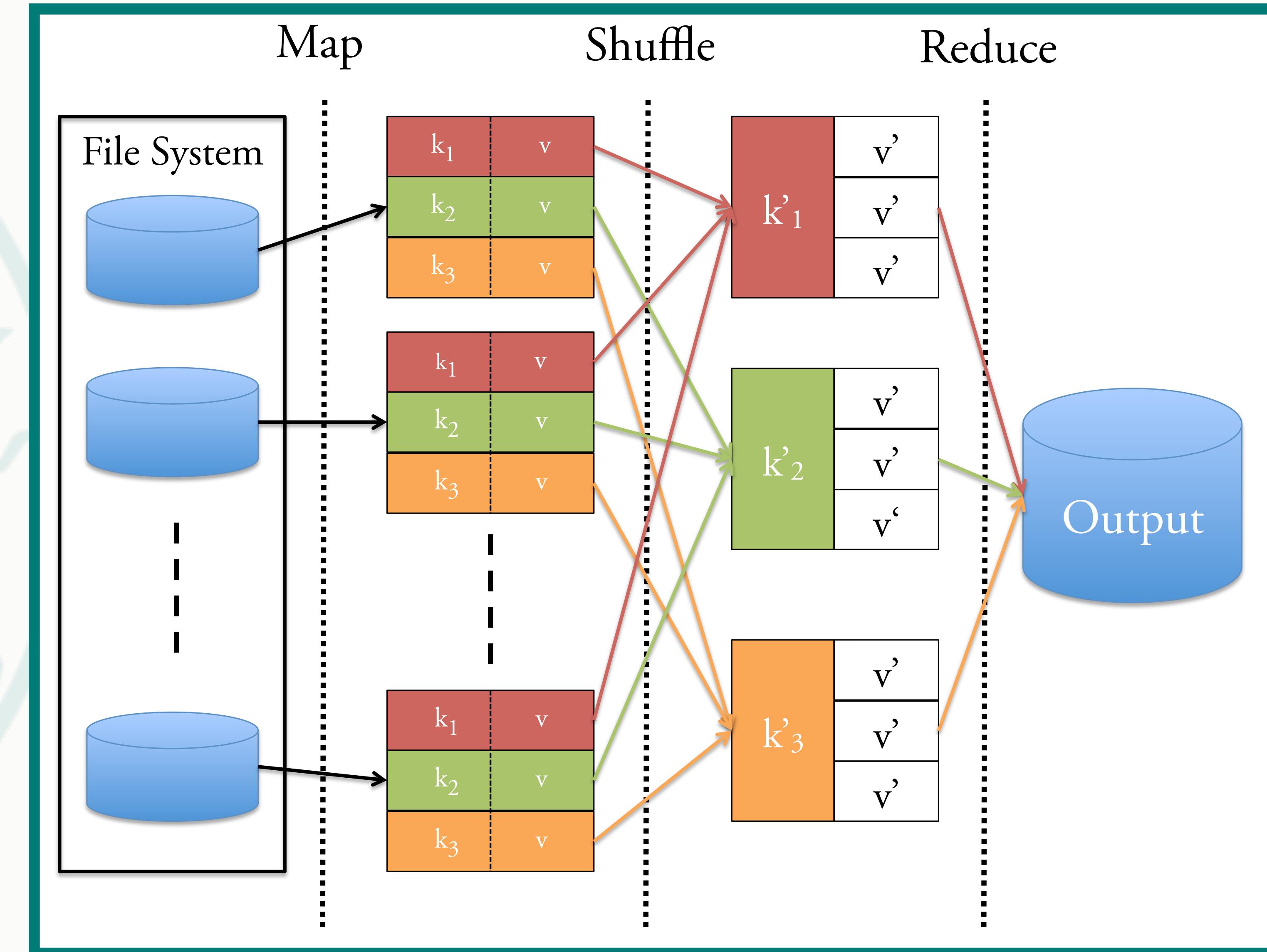
MapReduce

- La filosofía de MapReduce es:

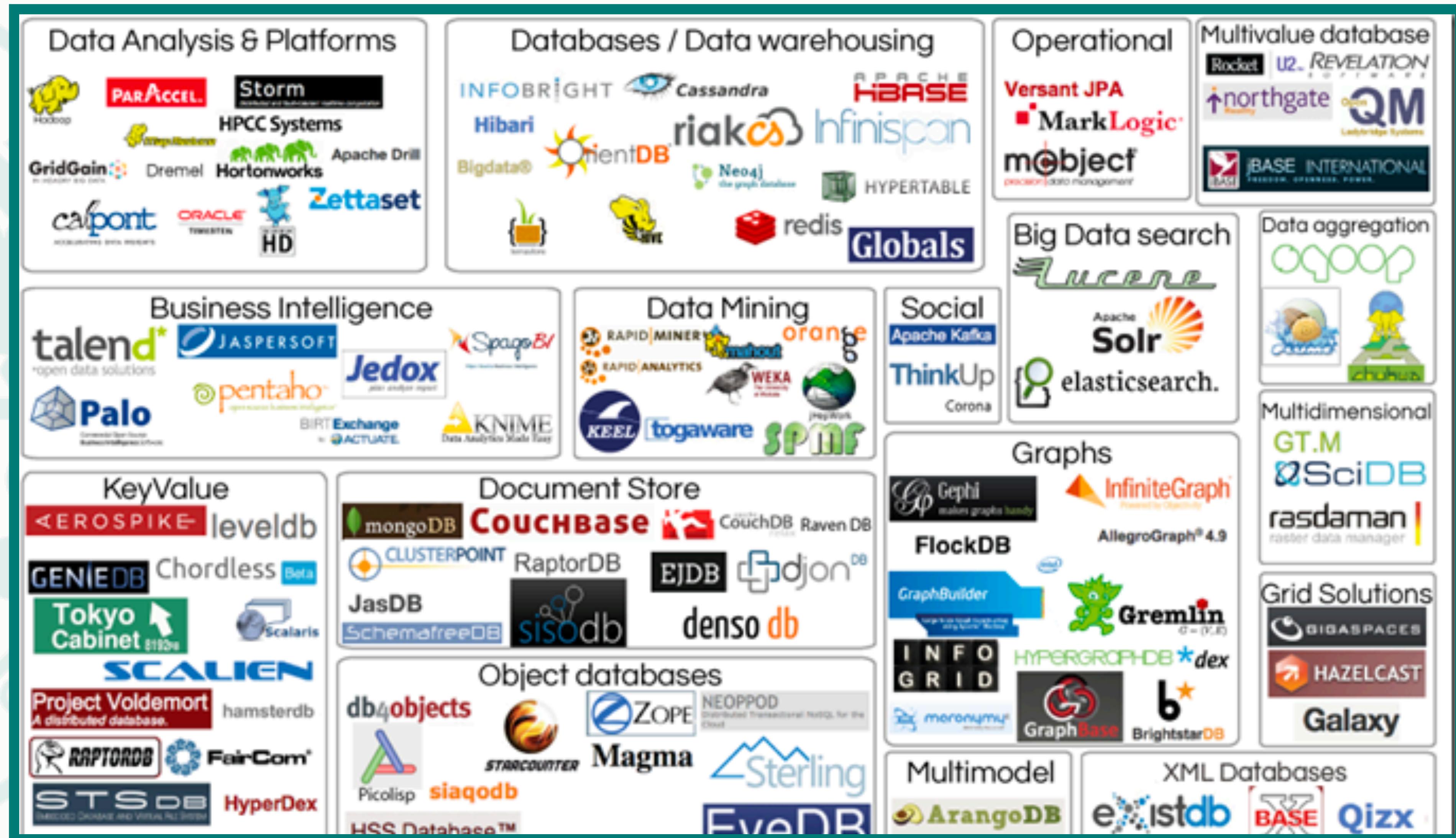
★ **make it scale**

★ **make it cheap**

MapReduce



MapReduce



Enfoques Big Data

Descubrimiento de reglas descriptivas
para aprendizaje supervisado

Descubrimiento de reglas descriptivas para aprendizaje supervisado

FENOTIPO

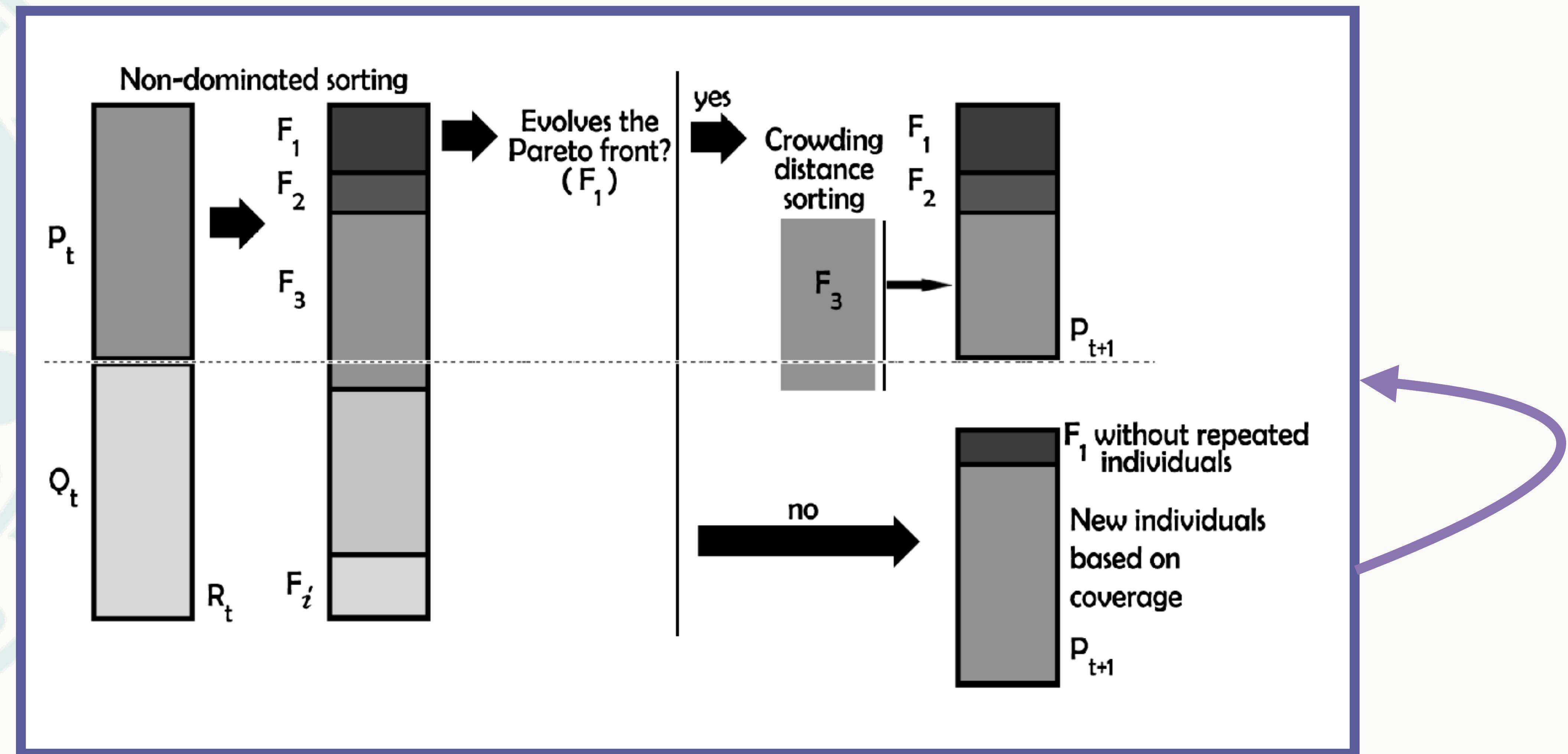
Si ($v_1=3 \text{ Y } v_2=5 \text{ Y } \dots \text{ Y } v_n=1$) entonces enfermo

	v_1	v_2	v_3	...	v_n
enfermo	1	4	1		1
enfermo	3	5	1		1
enfermo	1	4	0		3
NO enfermo	4	5	0		3
NO enfermo	3	5	0		1
NO enfermo	4	5	1		1

Cubiertos bien	Cubiertos mal
No cubiertos mal	No cubiertos bien

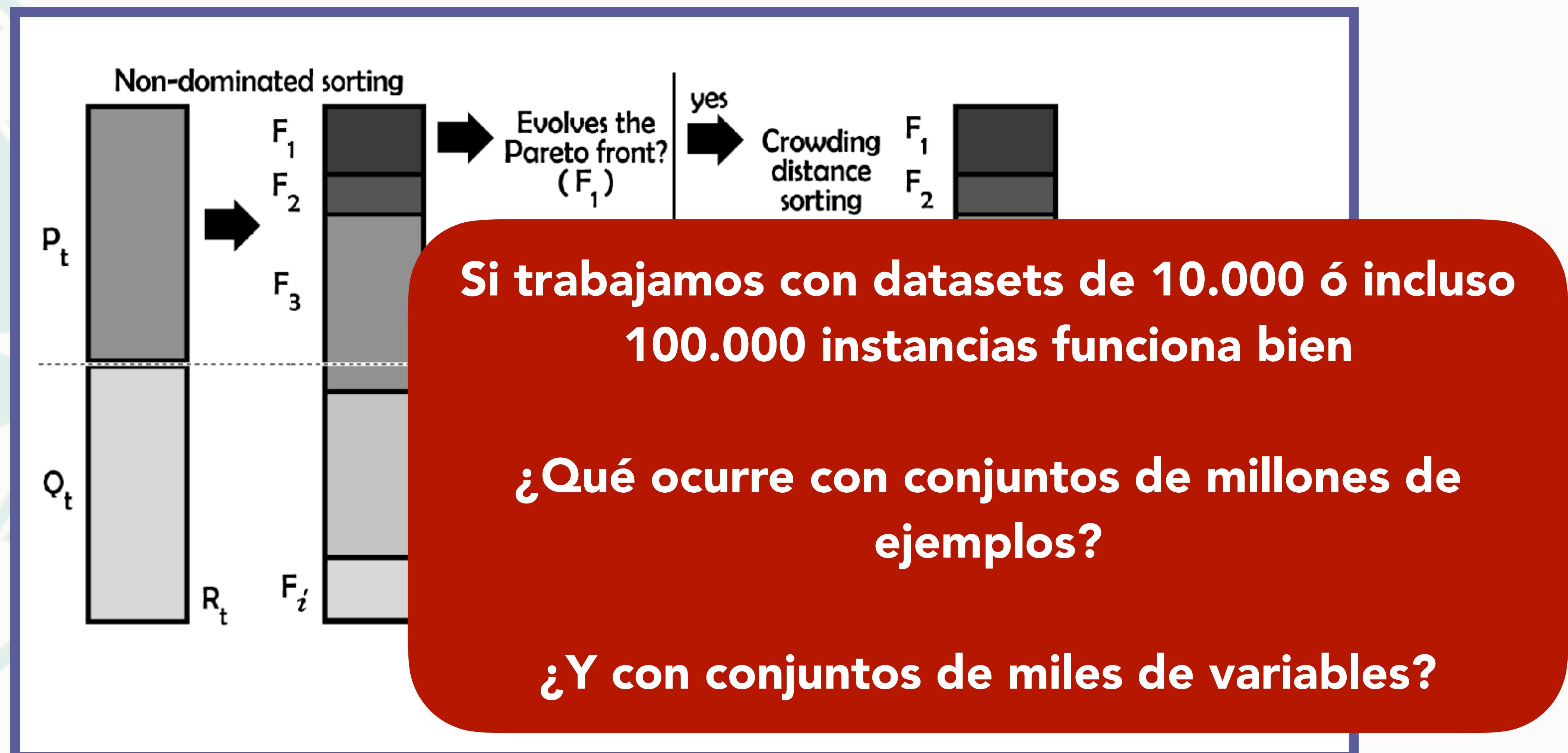
Descubrimiento de reglas descriptivas para aprendizaje supervisado

esquema del algoritmo



Descubrimiento de reglas descriptivas para aprendizaje supervisado

esquema del algoritmo



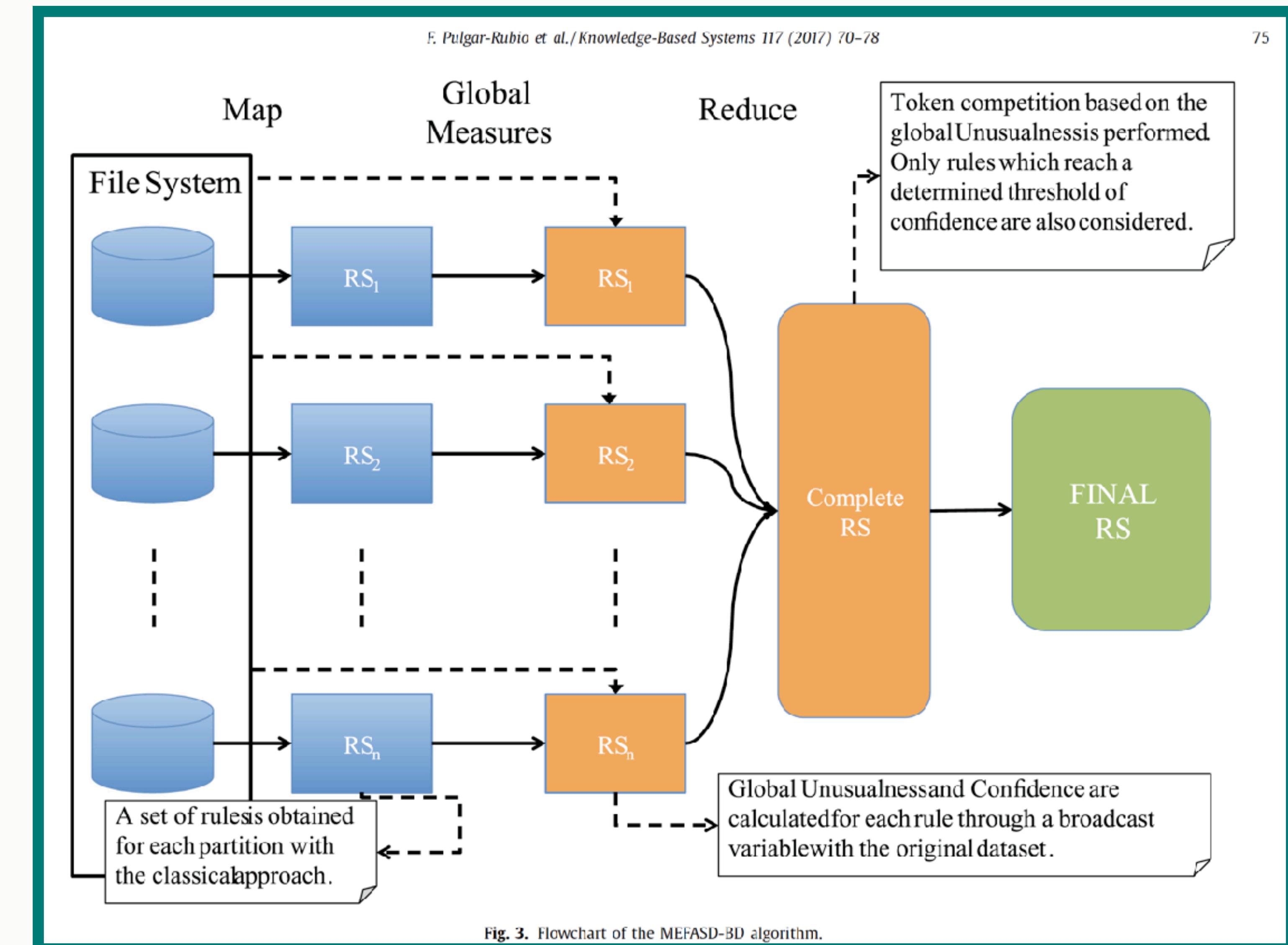
Descubrimiento de reglas descriptivas para aprendizaje supervisado

enfoque local

Local porque en cada nodo teníamos un algoritmo

Datos se dividían por nodos

Compartían reglas globales



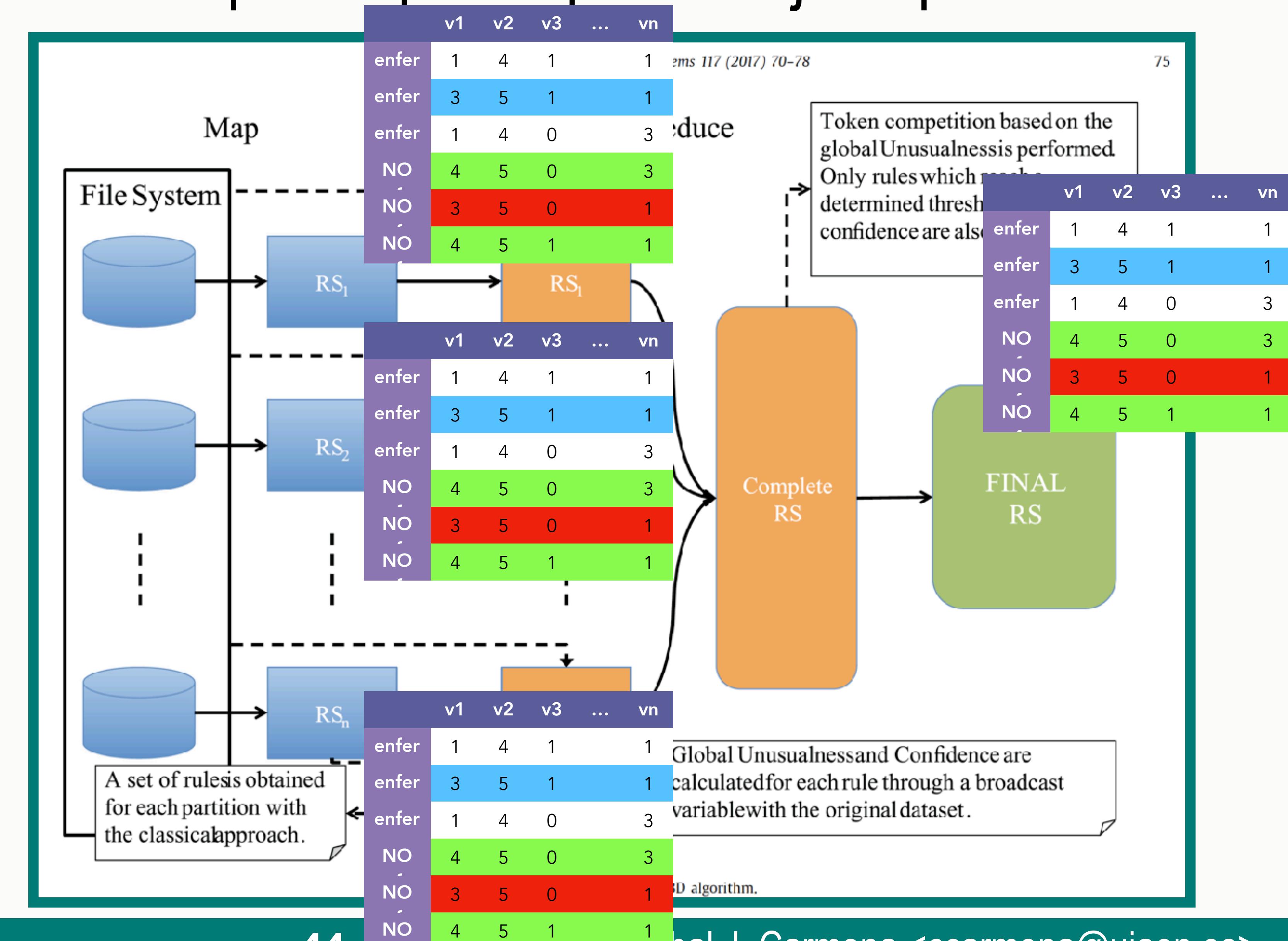
Descubrimiento de reglas descriptivas para aprendizaje supervisado

enfoque local

Local porque en cada nodo teníamos un algoritmo

Datos se dividían por nodos

Compartían reglas globales



Descubrimiento de reglas descriptivas para aprendizaje supervisado

enfoque local

PROBLEMA: Enorme intercambio de información en el proceso final.

Eficiencia baja cuando aumentábamos mucho los datos

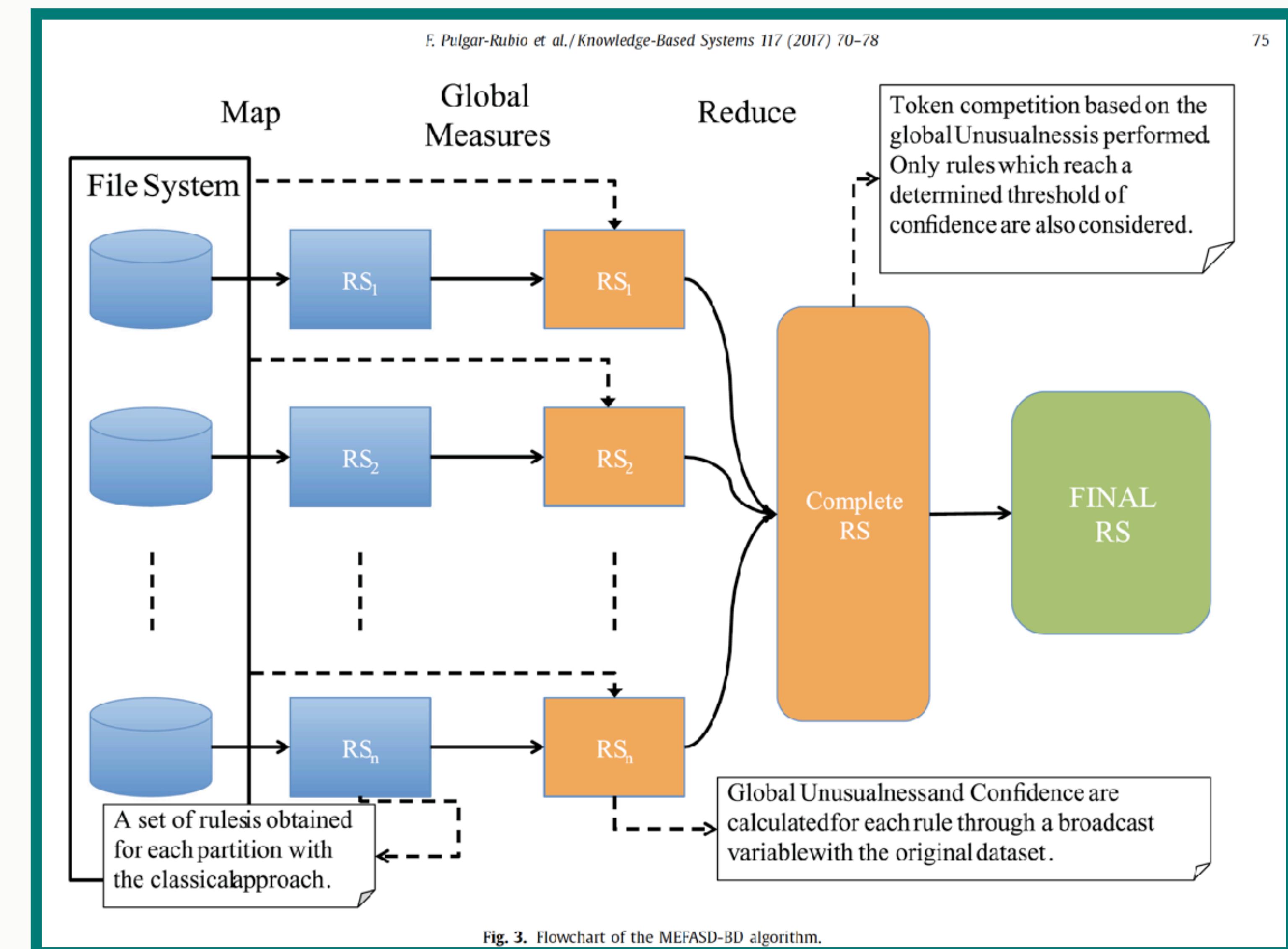


Fig. 3. Flowchart of the MEFASD-BD algorithm.

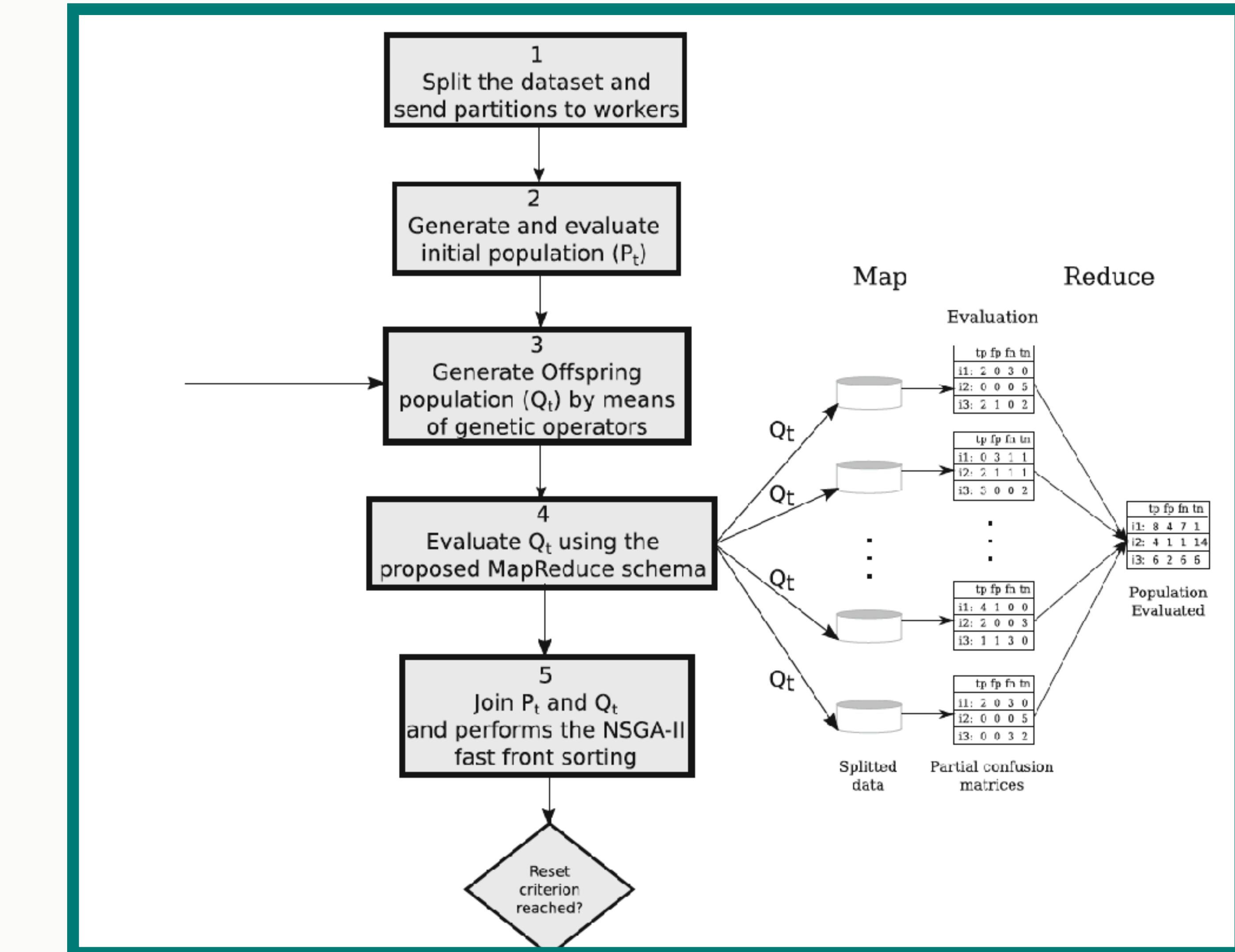
Descubrimiento de reglas descriptivas para aprendizaje supervisado

enfoque global

**Global porque solo
evoluciona un algoritmo**

**Datos se dividían por
nodos**

**Pero solo un algoritmo
en el nodo central**



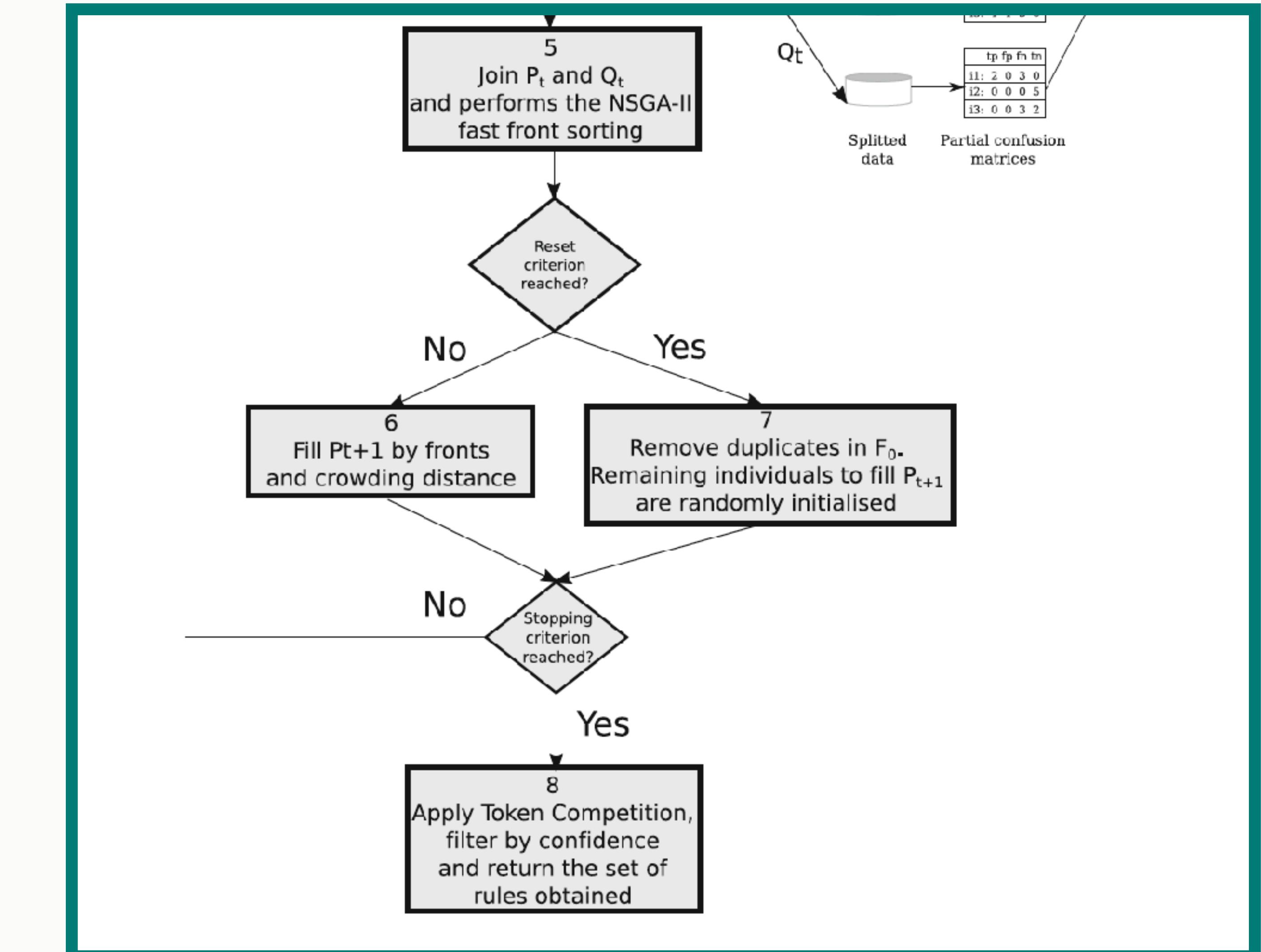
Descubrimiento de reglas descriptivas para aprendizaje supervisado

enfoque global

**Global porque solo
evoluciona un algoritmo**

**Datos se dividían por
nodos**

**Pero solo un algoritmo
en el nodo central**



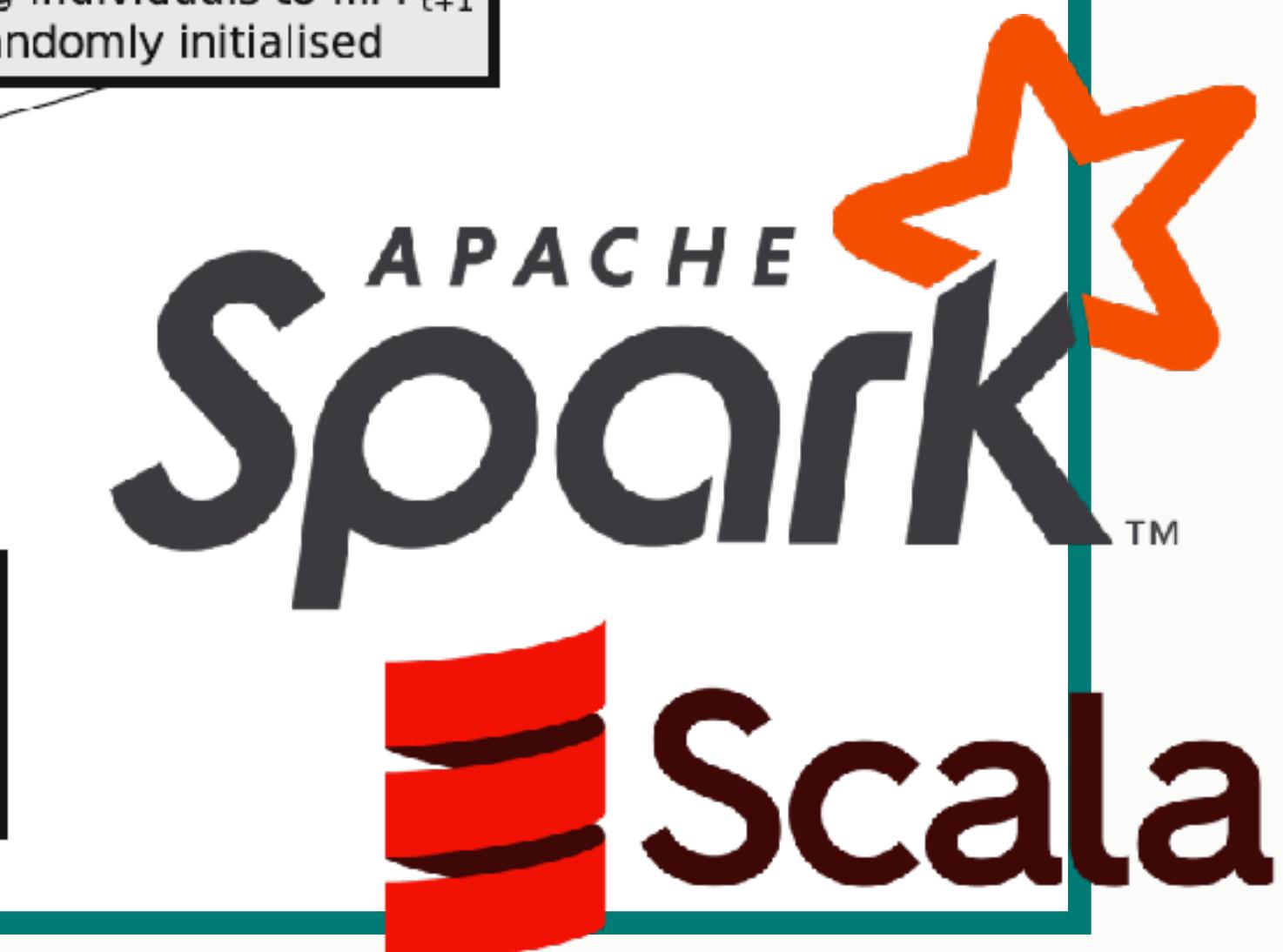
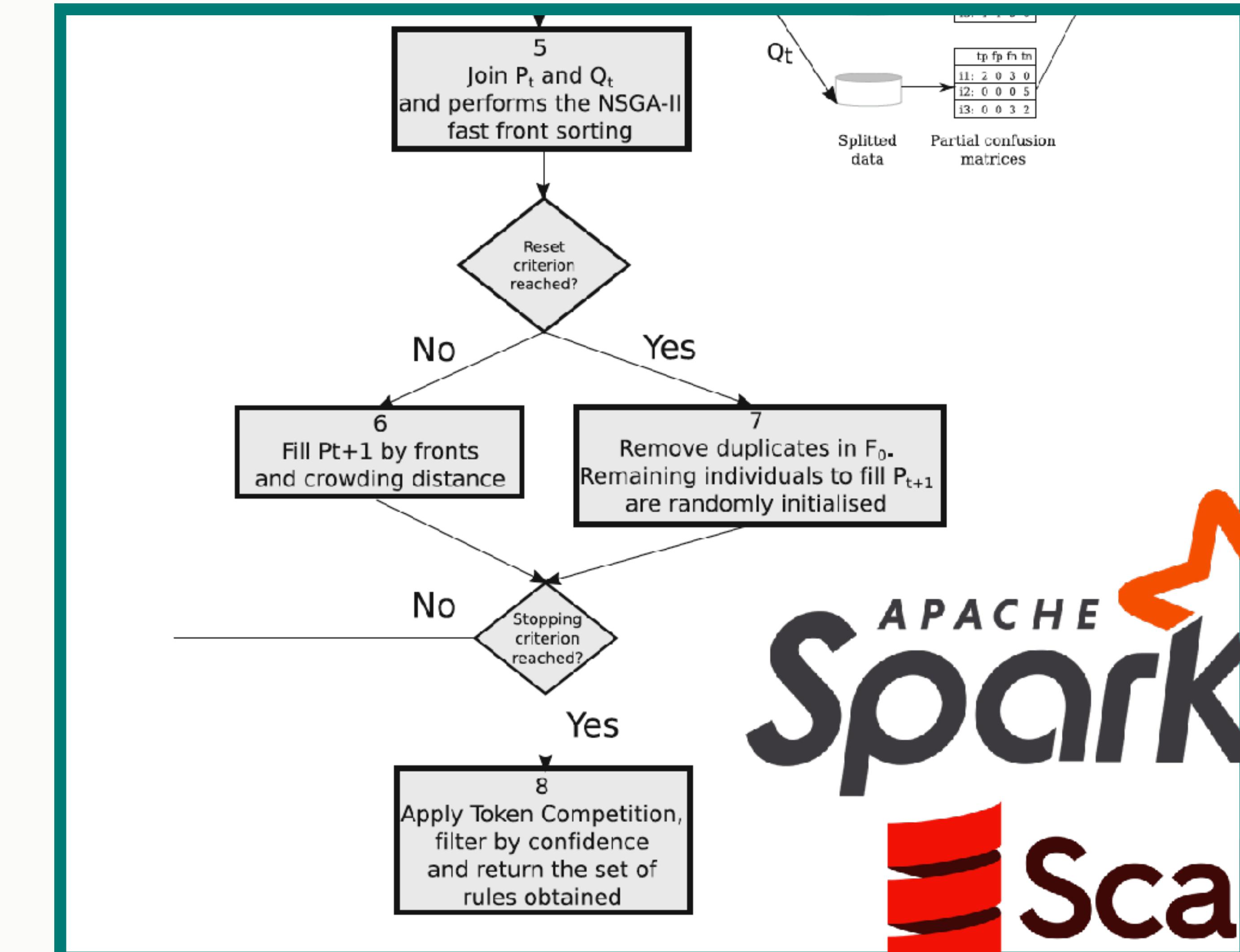
Descubrimiento de reglas descriptivas para aprendizaje supervisado

enfoque global

**Global porque solo
evoluciona un algoritmo**

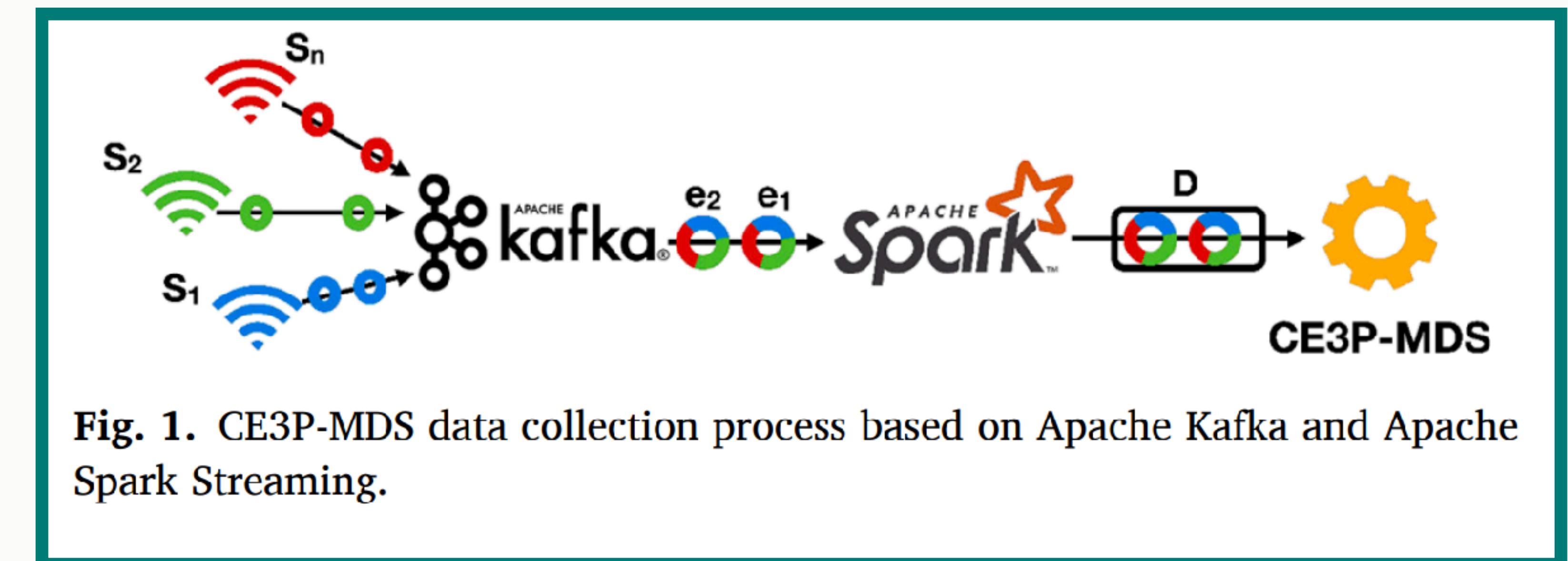
**Datos se dividían por
nodos**

**Pero solo un algoritmo
en el nodo central**



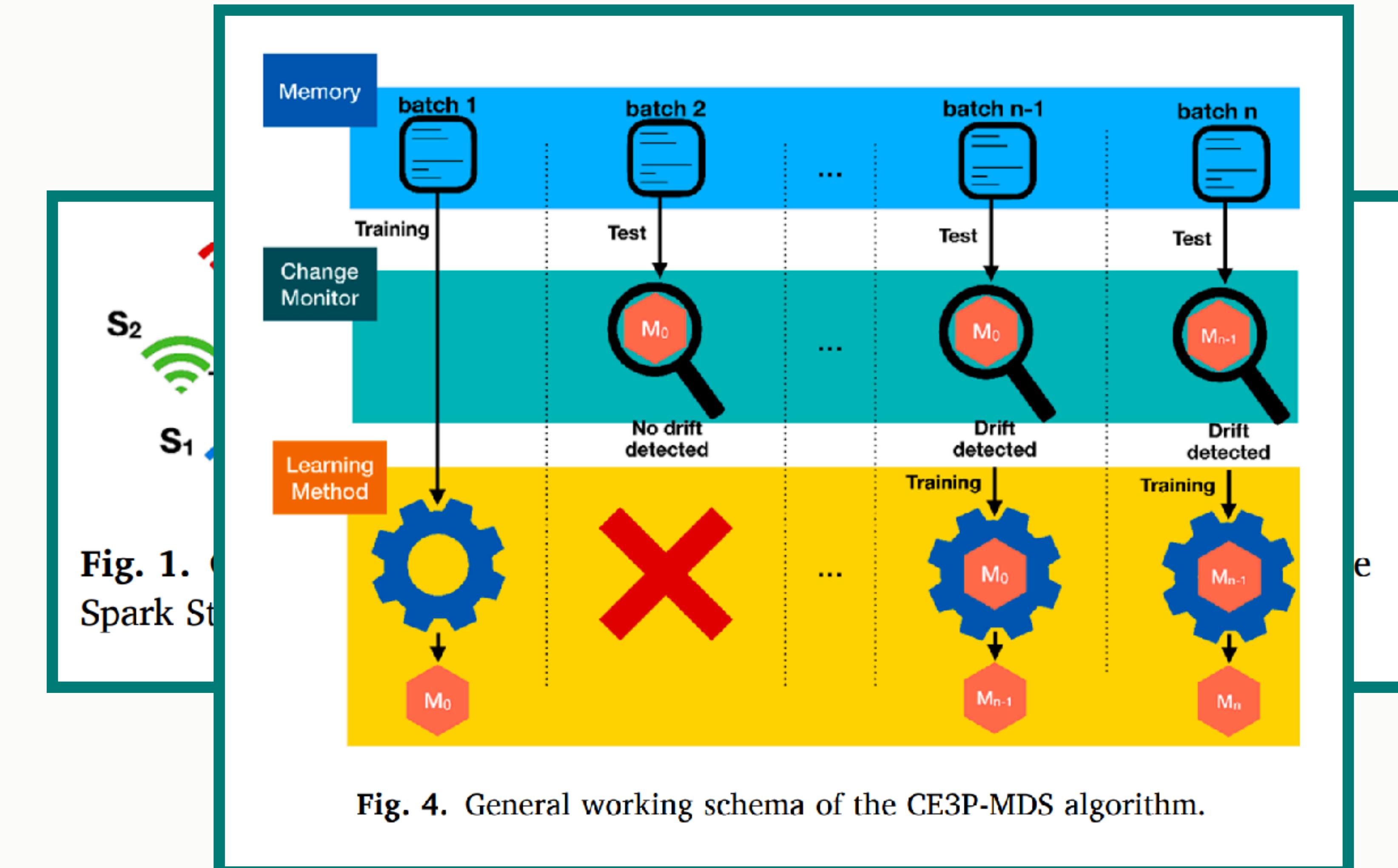
Descubrimiento de reglas descriptivas para aprendizaje supervisado

análisis en flujos continuos de datos



Descubrimiento de reglas descriptivas para aprendizaje supervisado

análisis en flujos continuos de datos





Metaheurísticas

Grado en Ingeniería Informática

Universidad de Jaén

Cristóbal J. Carmona

Curso 2023/2024

Esta obra está protegida con licencia
Creative Commons Atribución-NoComercial 4.0 Internacional

