



Computación Evolutiva

Jose Aguilar
aguilar@ula.ve

COMPUTACION EVOLUTIVA

- ENFOQUES ALTERNATIVOS DE BUSQUEDA Y APRENDIZAJE BASADOS EN MODELOS COMPUTACIONALES DE PROCESOS EVOLUTIVOS
- *IDEA:* BUSQUEDA ESTOCASTICA EVOLUCIONANDO A UN CONJUNTO DE ESTRUCTURAS Y SELECCIONANDO DE MODO ITERATIVO A LAS MAS APTAS

FINALIDAD: SUPERVIVENCIA DEL MAS APTO

MODO: ADAPTACION AL ENTORNO

COMPUTACION EVOLUTIVA

- EMULACION DE PROCESOS EVOLUTIVOS:
 - POBLACION DE POSIBLES SOLUCIONES => INDIVIDUOS
 - PROCESO DE SELECCIÓN => APTITUD DE LOS
INDIVIDUOS
 - PROCESO DE TRANSFORMACION => GENERACION DE
NUEVOS INDIVIDUOS

Características de la CE

- Se trabaja con la **codificación de un conjunto de parámetros** (o un subconjunto de ellos) y no con los parámetros en si.
- Se usa una **función objetivo o función de evaluación** para definir lo que se desea optimizar.

Esta función permite evaluar la calidad de los individuos en cada generación

=> **función de aptitud**

Características de la CE

- Se trabaja con una **población del problema**, no con un punto simple.
- Se usan **reglas de transición probabilísticas**, no reglas deterministas

COMPUTACION EVOLUTIVA

- OBJETIVOS CONFLICTIVOS QUE SE SIGUEN:
 - EXPLOTAR LAS MEJORES SOLUCIONES
 - EXPLORAR EL ESPACIO DE BUSQUEDA
- PARADIGMAS:
 - ALGORITMOS GENETICOS (HOLLAND)
 - PROGRAMAS EVOLUTIVOS (MICHALEWICZ)
 - PROGRAMACION GENETICA (KOZA)
 - ESTRATEGIAS EVOLUTIVAS (RECHENBERG SCHWEFEL)
 - PROGRAMACION EVOLUTIVA (FOGEL)

Procedimiento General de los Algoritmos Evolutivos

- **Generar** una población inicial.
- **Evaluar** los individuos.
- **Seleccionar**, a través de algún mecanismo, ciertos individuos de la población.
- **Modificar** los genes de los padres seleccionados usando los operadores genéticos.
- **Evaluar** los nuevos individuos.
- **Generar** una nueva población con la existente y los individuos generados en el paso 4.
- **Verificar** el **criterio de convergencia**, o regresar al paso 3.

COMPUTACION EVOLUTIVA

ASPECTOS DE IMPLEMENTACION:

- REPRESENTACION DE LOS INDIVIDUOS
- MEDIDAS DE ADAPTACION (FUNCION ADAPTATIVA)
- MECANISMOS DE SELECCIÓN Y REMPLAZO
- INTERPRETACION DE LOS RESULTADOS
- PARAMETROS Y VARIABLES QUE CONTROLAN EL PROCESO
- OPERADORES GENETICOS A UTILIZAR

Representación de los Individuos

- **Tipos**

- Cadenas binarias $\Rightarrow 2^l$ soluciones en vector de longitud l
- Vector de valores reales: incluye vector de desviación standard
- Elementos del problema
- Estados Finitos
- Arboles

Operadores Genéticos

- Mecanismos de transformación que se utilizan para recorrer el espacio de las soluciones de un problema.

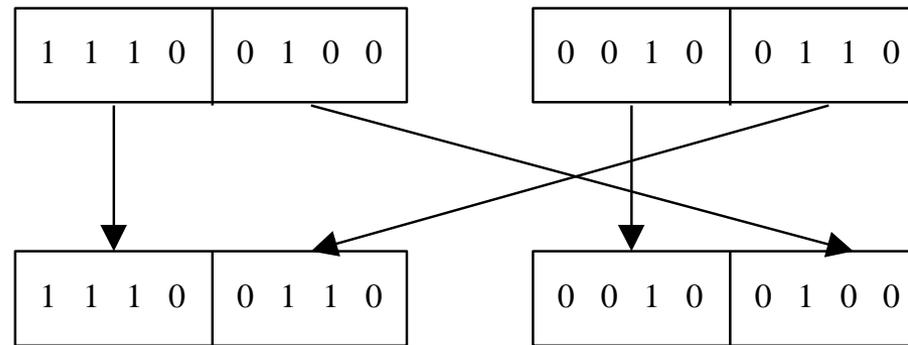
realizan los cambios de la población durante la ejecución de un algoritmo evolutivo.

- En general, se basan en los operadores genéticos biológicos.
- Operadores genéticos clásicos:
 - Mutación
 - Cruce

Operador Cruce

- Operador, normalmente **binario**,
- Permite representar el proceso de **apareamiento natural**
- **Intercambia pedazos de información** entre diversos individuos para producir dos nuevos individuos.
- El **punto de corte** es escogido aleatoriamente.

Operador Cruce

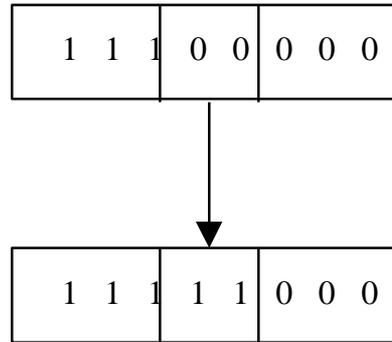


- **Tipos:**
 - Cruce multipunto
 - Cruce segmentado
 - Cruce uniforme

Operador de Mutación

- Operador **unario** que simula el proceso evolutivo que ocurre en los individuos cuando cambian su estructura genética.
- Se **seleccionan aleatoriamente componentes (genes) de un individuo para ser modificados**, también de forma aleatoria.
- A través del operador de mutación se puedan **producir cambios** en la estructura de un individuo

Operador de Mutación



- **Tipos**

- Mutaciones sobre genes
- Mutaciones no estacionarias
- Mutaciones no uniformes

Operadores Avanzados

- Dominación
- Segregación
- Inversión
- Duplicación

Mecanismos de funcionamiento de los algoritmos evolutivos

- **Estrategias de selección** de los individuos susceptibles a reproducirse.
- **Estrategia de apareamiento** de los individuos en la fase de reproducción.
- **Estrategia de reemplazo.**

Mecanismos de Selección

Políticas empleadas para la conformación de los progenitores.

- **Tipos:**

- elitesca
- aleatoria
- Por sorteo
- universal (por ruleta)
- Por torneos

Muestreo por sorteo

$$p_i = \frac{f_i}{\sum_{j=1}^n f_j}$$

- Se calculan las probabilidades acumulativas (q_i):
 - $q_0 = 0$.
 - $q_i = q_{i-1} + p_i$.
- Se genera un número aleatorio α entre 0 y 1.
- Se elige el individuo i que cumpla con la condición:

$$q_{i-1} < \alpha < q_i$$

Muestreo universal

Se genera un número aleatorio α y se utiliza este para crear los números:

$$\alpha_i = \frac{\alpha + i - 1}{k} \quad \forall i = 1, k$$

Una vez obtenidos estos números, se aplica k -veces el paso 3 del muestreo por sorteo.

Muestreo por torneos

Se toman al azar m individuos de la población, y el mejor de estos m individuos formara parte de la muestra.

Mecanismos de Reemplazo

- Tipos:
 - Padre por hijo, o Reemplazo directo
 - Por similitud
 - Por inserción (,), p.e., peores por mejores
 - Por inclusión (+)

Estrategia de Apareamiento

Qué individuos con qué individuos aparear en la fase de reproducción

- Tipos:
 - Aleatorio,
 - Autofertilización,
 - etc.

ALGUNOS PARAMETROS Y VARIABLES QUE CONTROLAN EL PROCESO

- **Probabilidades** de uso de los operadores
- **Criterios de parada**
 - Por ejemplo, Número de generaciones
- **Tamaño y diversidad** de la población

COMPUTACION EVOLUTIVA

- OPTIMIZACION
- ADAPTACION
- MAQUINAS INTELIGENTES
- CONSTRUCCION DE PATRONES

PROBLEMAS MULTIOBJETIVOS

- Muchos problemas de optimización se reducen a la consideración de **un solo criterio**, mientras que en otros problemas es necesario evaluar **varios criterios**.

problemas multicriterios o multiobjetivos

- Las técnicas de optimización convencional, tales como los métodos basados en gradiente, y unos menos convencionales, tales como recocido simulado, son **difíciles de extender a casos verdaderamente multiobjetivos**, ya que estos no fueron diseñados para evaluar esto.

PROBLEMAS MULTIOBJETIVOS

En el mundo real, la mayor parte de los problemas tienen **varios objetivos** (posiblemente en conflicto entre sí) que se desea sean **satisfechos de manera simultánea**.

Muchos de estos problemas suelen convertirse a **mono-objetivo** transformando todos los objetivos originales, menos uno, en restricciones.

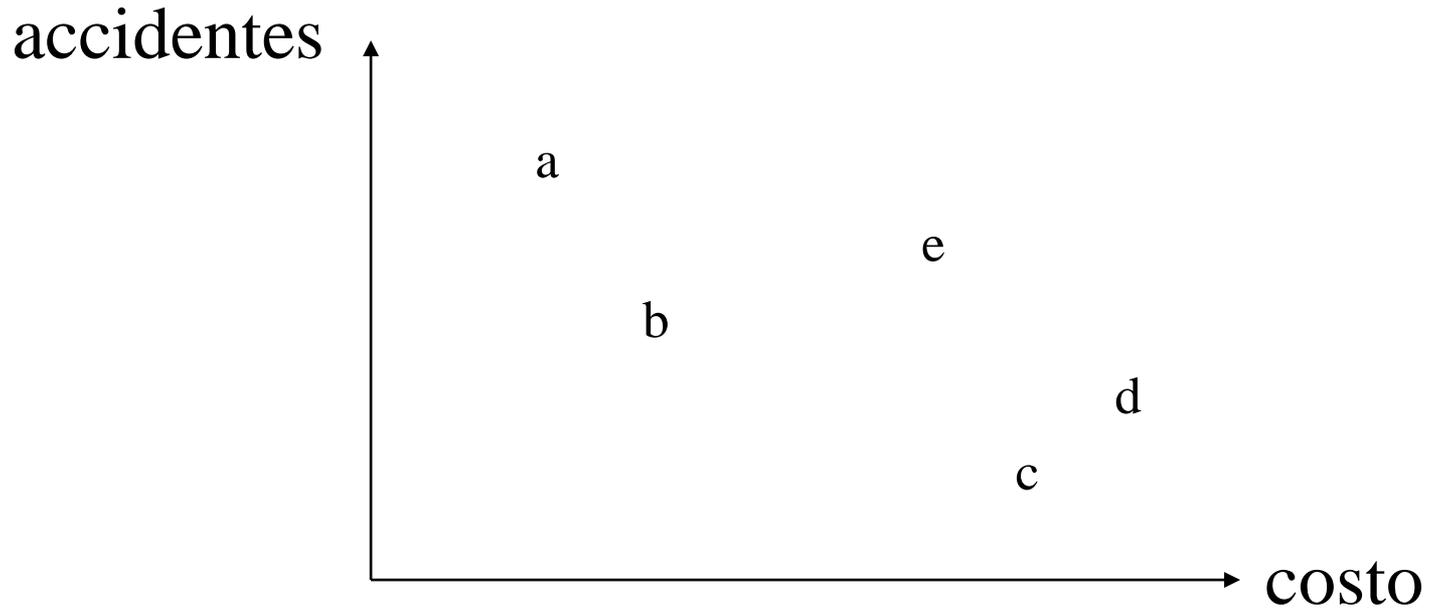
Hay tres tipos de situaciones que pueden presentarse en un problema multiobjetivo:

- **Minimizar todas las funciones objetivo.**
- **Maximizar todas las funciones objetivo.**
- **Minimizar algunas funciones y maximizar otras.**

PROBLEMAS MULTIOBJETIVOS

- Ejemplo: fabrica de cauchos que deben minimizar dos costos:
 - **numero de accidentes y**
 - **costo de producir cauchos**
- Escenarios:
 - $a = (2,10)$
 - $b = (4,6)$
 - $c = (8,4)$
 - $d = (9,5)$
 - $e = (7,8)$

PROBLEMAS MULTIOBJETIVOS



Individuos **no dominados**: a, b, c!!!

Problemas de Localización Multiobjetivo

Escoger un conjunto de lugares para situar varias instalaciones que prestarán su servicio a un conjunto de clientes:

- **La función objetivo** es generalmente el costo de operación (fijo y variable)
- En algunos contextos es necesario considerar **otras funciones objetivo**: distancia, cobertura, tiempo de respuesta, equidad, etc.
- **Criterios en conflicto**
 - Costo
 - Cobertura

Problemas de Localización Multiobjetivo

Variables del modelo

- I, i : Conjunto lugares de localización de las instalaciones
m instalaciones
- J, j : Conjunto e índice de los lugares de demanda, n clientes
- f_i : Costo fijo de operar una instalación en el lugar i.
- c_{ij} : Costo de atender toda la demanda del lugar j desde la instalación i.
- d_j : Demanda del cliente j.
- h_{ij} : Distancia entre i y j

COMPONENTES DEL PROBLEMA DE COBERTURA

- D_{max} : Distancia máxima de cobertura.
- Q_j : $\{i: h_{ij} \leq D_{max}\}$, conjunto de instalaciones que pueden atender la demanda del nodo j cumpliendo con la distancia máxima de cobertura.

Problemas de Localización Multiobjetivo

Variables de decisión

$$y_i = \begin{cases} 1, & \text{si se abre la bodega } i, \\ 0, & \text{de lo contrario.} \end{cases}$$

$$x_{ij} = \begin{cases} 1, & \text{si el cliente } j \text{ es atendido por la bodega } i, \\ 0, & \text{de lo contrario.} \end{cases}$$

Problemas de Localización Multiobjetivo

Función objetivo

$$\text{mín } z_1 = \sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{J}} c_{ij} x_{ij} + \sum_{i \in \mathcal{I}} f_i y_i$$

$$\text{máx } z_2 = \sum_{j \in \mathcal{J}} d_j \sum_{i \in \mathcal{Q}_j} x_{ij}$$

Restricciones

$$\sum_{i \in \mathcal{I}} x_{ij} = 1 \quad , \quad j \in \mathcal{J}$$

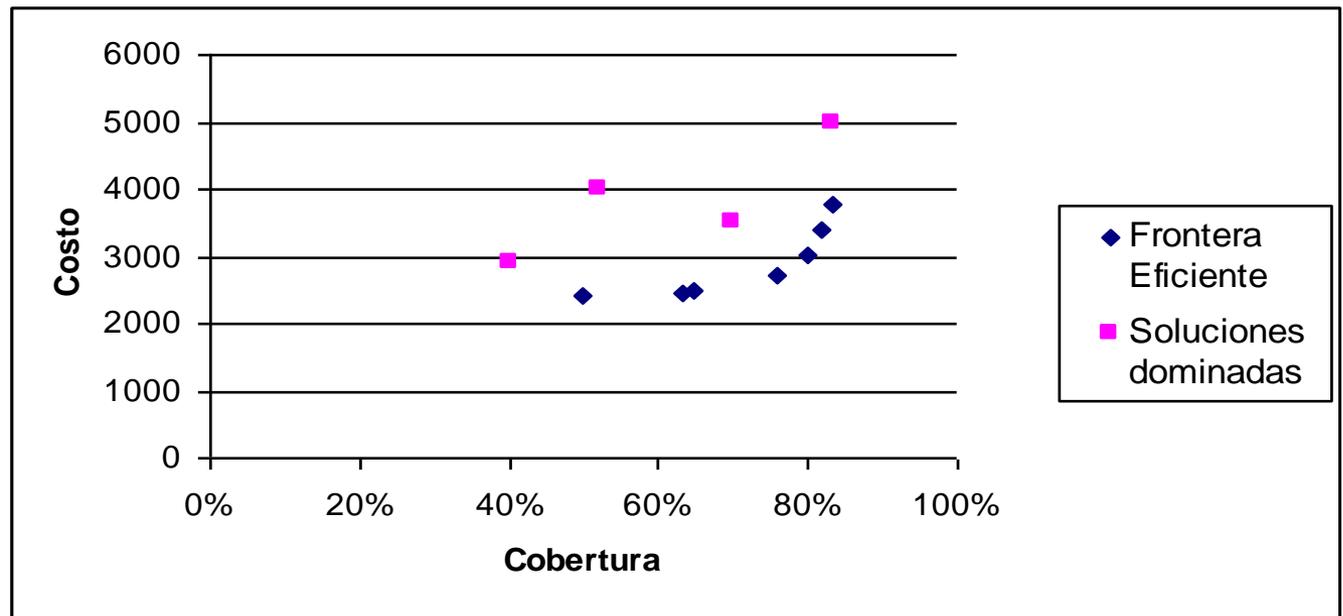
$$x_{ij} \leq y_i \quad , \quad j \in \mathcal{J}, \quad i \in \mathcal{I}$$

$$x_{ij} \in \{0, 1\}, \quad j \in \mathcal{J}, \quad i \in \mathcal{I}$$

$$y_i \in \{0, 1\}, \quad i \in \mathcal{I}$$

Generalidades Optimización Multiobjetivo

- Objetivos en conflicto.
- Se busca un **conjunto de soluciones eficientes**, en las cuales no sea posible **mejorar el valor de una de las funciones objetivo sin deteriorar el desempeño de la otra**.



PROBLEMAS MULTIOBJETIVOS

TEORIA DE PARETO

- A diferencia de la optimización de un único objetivo, la solución a un problema multiobjetivo no es un único punto, sino una familia de puntos conocida como el "**Conjunto Optimo de Pareto**".
- Cada punto en esa superficie **es óptimo** en el sentido de que **no pueden realizarse mejoras en un componente del vector de costo que no conduzca a la degradación en al menos uno de los componentes restantes**.
- La noción de **optimalidad de Pareto** es sólo un primer paso hacia la solución práctica de un problema multiobjetivos; también envuelve la **escogencia de una única solución** compromiso del "**Conjunto Optimo de Pareto**" de acuerdo a alguna **información de preferencia**.

PROBLEMAS MULTIOBJETIVOS: TEORIA DE PARETO

Decimos que un vector de variables de decisión $x^* \in \alpha$ (α es la zona factible) es un **óptimo de Pareto** si no existe otro $x \in \alpha$ tal que

$$f_i(x) \leq f_i(x^*) \text{ para toda } i = 1, \dots, k$$

x^* es un óptimo de Pareto si no existe ningún vector factible de variables de decisión $x^* \in \alpha$ que **decremente algún criterio sin causar un incremento simultáneo en al menos un criterio.**

Desafortunadamente, este concepto casi siempre produce no una solución única sino un conjunto de ellas a las que se les llama **conjunto de Pareto.**

Los vectores x^* correspondientes a las soluciones incluidas en el conjunto de óptimos de Pareto son llamados **no dominados.**

La gráfica de las funciones objetivo cuyos vectores no dominados se encuentran en el conjunto de óptimos de Pareto se denominan **frente de Pareto.**

PROBLEMAS MULTIOBJETIVOS

TEORIA DE PARETO

"**Conjunto Optimo de Pareto**" esta compuesto por elementos que son soluciones **no-inferiores (no-dominados)** para el problema multiobjetivos:

Concepto de Inferioridad: Un vector $\mathbf{u} = (u_1, \dots, u_n)$ se dice es inferior a $\mathbf{v} = (v_1, \dots, v_n)$ si \mathbf{v} es parcialmente menor que \mathbf{u} ($\mathbf{v} p < \mathbf{u}$); es decir:

$$i = 1, \dots, n, v_i \leq u_i \quad i = 1, \dots, n : v_i < u_i$$

Concepto de Superioridad: Un vector $\mathbf{u} = (u_1, \dots, u_n)$ se dice es superior a $\mathbf{v} = (v_1, \dots, v_n)$ si \mathbf{v} es inferior a \mathbf{u} .

ESQUEMAS RESOLUCION PROBLEMAS MULTIOBJETIVOS

- **No basadas en Pareto**
 - Suma lineal de pesos
 - VEGA
 - Método ε -constraint
 - Teoría de juegos
- **Basadas en Pareto**
 - Jerarquización de Pareto “pura”
 - PAES (Pareto Archived Evolution Strategy)
 - NSGA y NSGA II (Nondominated Sorting Genetic Algorithm)
 - SPEA (Strength Pareto Evolutionary Algorithm).

PROBLEMAS MULTIOBJETIVOS

ENFOQUES BASADOS EN LA DIVISIÓN DE LA POBLACIÓN

1.- Dividir población en n subpoblaciones S_1, \dots, S_n .

% se va a iterar un AG con una FO distinta en c/u.

2.- Definir vector de prioridad para las FOs.

3.- Repetir Hasta (Evaluar todas las FOs)

3.1 Para cada S_i , $\forall i = 1, \dots, n$, usar la FO correspondiente según el vector de prioridad.

3.1.1 Iterar el AG para esa FO _{i} según un número γ y dado de iteraciones en cada S_i .

3.1.2 Reemplazar X % de individuos en cada población S_i que cumplen mal la FO actual.

PROBLEMAS MULTIOBJETIVOS

ENFOQUES BASADOS EN POBLACIÓN TOTAL

- 1.- Inicio.
- 2.- Repetir Hasta (Que c/FO sea escogida un mínimo número de veces ó población converja).
 - 2.1 Escoger una FO (aleatoriamente o cíclicamente).
 - 2.2 Iterar el AG en la población usando la FO escogida.

PROBLEMAS MULTIOBJETIVOS

ENFOQUES BASADOS EN LA TEORIA DE PARETO

Para obtener soluciones no-inferiores se pueden usar los AGs:

- manteniendo una población de soluciones, los AGs pueden buscar muchas soluciones no-inferiores en paralelo.

ENFOQUES BASADOS EN LA TEORIA DE PARETO

Enfoque en dos fases:

1. Se clasifican los individuos en **no-inferiores o no (no-dominados o dominados)**, para cada una de las FOs que conforman la FMO.
2. Se utilizan los individuos no-dominados como los **padrotes** para la fase de reproducción.

Macro Algoritmo:

- 1.- Generar población.
- 2.- Repetir Hasta que población generada sea igual a la anterior o sea homogénea.
 - 2.1.- **Fase de Clasificación:** Agrupar individuos en dominados y no-dominados para c/FO.
 - 2.2.- **Fase de Optimización:** Generar nueva población según un AG clásico.

ENFOQUES BASADOS EN LA TEORIA DE PARETO

Fase de Clasificación:

- 1.- Generar población inicial.
- 2.- Evaluar los individuos para c/FO y establecer su clasificación en c/FO.
- 3.- Seleccionar individuos no-dominados (Según c/u de las FOs).
- 4.- Asignar rango 1 a los individuos no-dominados y remover de la disputa, luego se encuentra un nuevo conjunto de individuos no-dominados con rango 2, y así sucesivamente. Esto se hace para c/FOs.

Fase de Optimización:

- 1.- Seleccionar los individuos para reproducción por algún mecanismo clásico (ruleta, elitesco).
- 2.- Reproducir con operadores clásicos, usando los individuos no-dominados como los padrotes.

ESQUEMAS RESOLUCION

PROBLEMAS MULTIOBJETIVOS

Input : N: population size

N': archive size

T: maximum number of generations

Output: A: nondominated set

- 1. Initialization :** Generate an initial population P_0 and create the empty archive (external set) $P_0' = ;$. Fix $t = 0$ (i.e. first generation).
- 2. Fitness Assignment:** Compute fitness values for individuals in P_t and P_t' .
- 3. Environmental Selection:** Find the nondominated individuals in P_t and P_0' and copy them in P_{t+1}' . If the size of P_{t+1}' exceeds N' , then reduce P_{t+1}' by a truncation operator. Conversely, if the size of P_{t+1}' is smaller than N' , then fill P_{t+1}' with dominated individuals in P_t and P_t' .
- 4. Termination:** If $t \geq T$ or any other stopping criteria is satisfied, then A is the set of nondominated individuals, i.e. optimal solutions. **Stop.**
- 5. Mating Selection:** perform binary tournament selection with replacement on P_{t+1}' to fill the mating pool.
- 6. Variation:** apply crossover and mutation to the mating pool, generating population P_{t+1} . Increase the generation counter and **go to step 2.**



UNIVERSIDAD
DE LOS ANDES
MÉRIDA VENEZUELA



ESCUELA
POLITÉCNICA
NACIONAL

PROGRAMACIÓN EVOLUTIVA

PROGRAMACIÓN EVOLUTIVA

Evoluciona algoritmo que opera secuencia de símbolos tomados de un **alfabeto finito**

Símbolo de salida maximiza rendimiento si **algoritmo predice próximo símbolo**

Énfasis en el **comportamiento**

Por ejemplo: un conjunto de *maquinas de estado finito* se expone a un ambiente.

PROGRAMACIÓN EVOLUTIVA

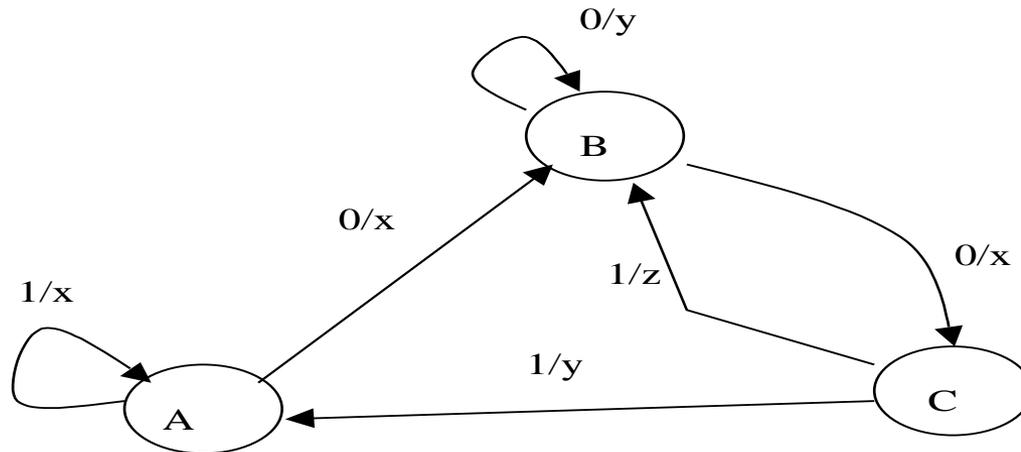
Para cada Máquina de Estados Finitos, su **predicción es medida** con respecto al ambiente, como una **función error**.

Después que se hace la ultima predicción, un **error promedio** para cada máquina es calculado para indicar la **aptitud de ella**.

Las máquinas que presenten el **mejor resultado** (mínimo error) se retienen para **ser padres** en la próxima generación.

PROGRAMACIÓN EVOLUTIVA

Edo. Presente	C	B	C	A	A	B
Entrada	0	1	1	1	0	1
Prox.	B	C	A	A	B	C
Salida	X	Z	Y	X	X	Z



PROGRAMACIÓN EVOLUTIVA

MACROALGORITMO

1. DEFINIR SECUENCIA DE SIMBOLOS Y FA
2. INICIAR POBLACION MEF
3. INTRODUCIR SIMBOLO ENTRADA Y VER SU SALIDA
4. EVALUAR SI PREDIJO BIEN
5. CREAR NUEVOS HIJOS USANDO MUTACION: **CAMBIAR SIMBOLOS, ESTADO DE TRANSICION, O EDO. INICIAL, ANADIR O ELIMINAR SIMBOLO**
6. SELECCIONAR MEJORES PADRES
7. REGRESAR A 3

Esquema de generación de descendientes

Por cada maquina padre se obtiene un descendiente aplicando sobre esta una mutación aleatoria:

- Cambiar un símbolo de salida.
- Cambiar un estado de transición.
- Agregar un estado.
- Eliminar un estado.
- Cambiar el estado inicial.

PROGRAMACIÓN EVOLUTIVA

EJEMPLO: PREDICCIÓN DE TIEMPO

{SOL, LLUVIA, SOL, LLUVIA, TORMENTA, ...}

REPRESENTADO POR {0, 1, 0, 1, 2, ...}

MATRIZ DE PAGO

	0	1	2
0	1	-10	-100
1	-10	5	-40
2	-25	-5	10



UNIVERSIDAD
DE LOS ANDES
MÉRIDA VENEZUELA



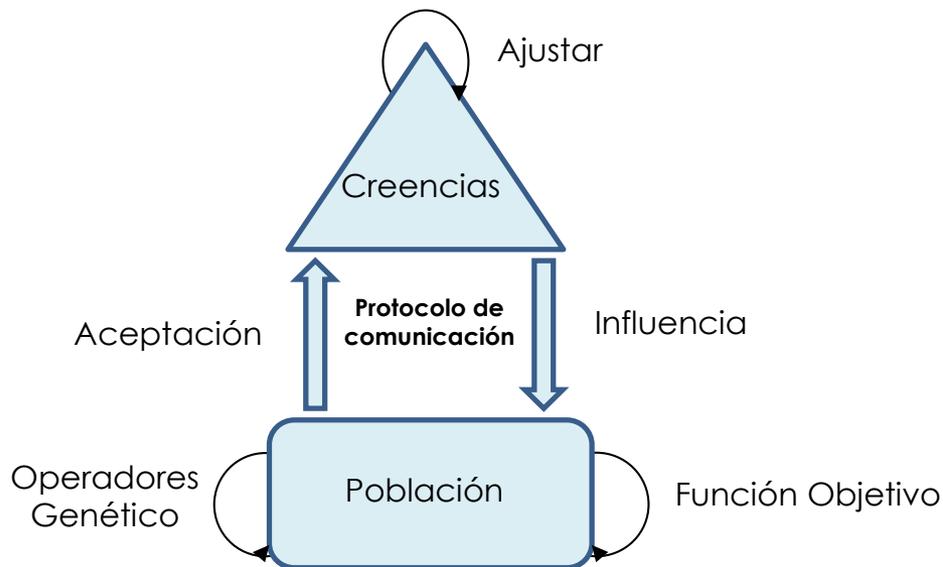
ESCUELA
POLITÉCNICA
NACIONAL

Algoritmos Culturales

Algoritmos culturales (AC):

Basado en la **evolución cultural**, vista como un proceso de herencia a dos niveles:

- **el nivel micro-evolutivo**, que consiste en el *material genético heredado* por los padres a sus descendientes, y
- **el nivel macro-evolutivo**, que es el conocimiento adquirido por los individuos a través de las generaciones, y que una vez codificado y almacenado, sirve para *guiar el comportamiento* de los individuos que pertenecen a una población



```
Begin
t = 0;
Iniciar Pt
Iniciar Bt
Repetir
    Evaluar Pt
    Ajuste (Bt, Acepte (Pt))
    Variación (Pt, influencia (Bt))
    t = t + 1;
    Seleccionar Pt de Pt-1
Hasta (haber alcanzado la condición de finalizar)
End
```

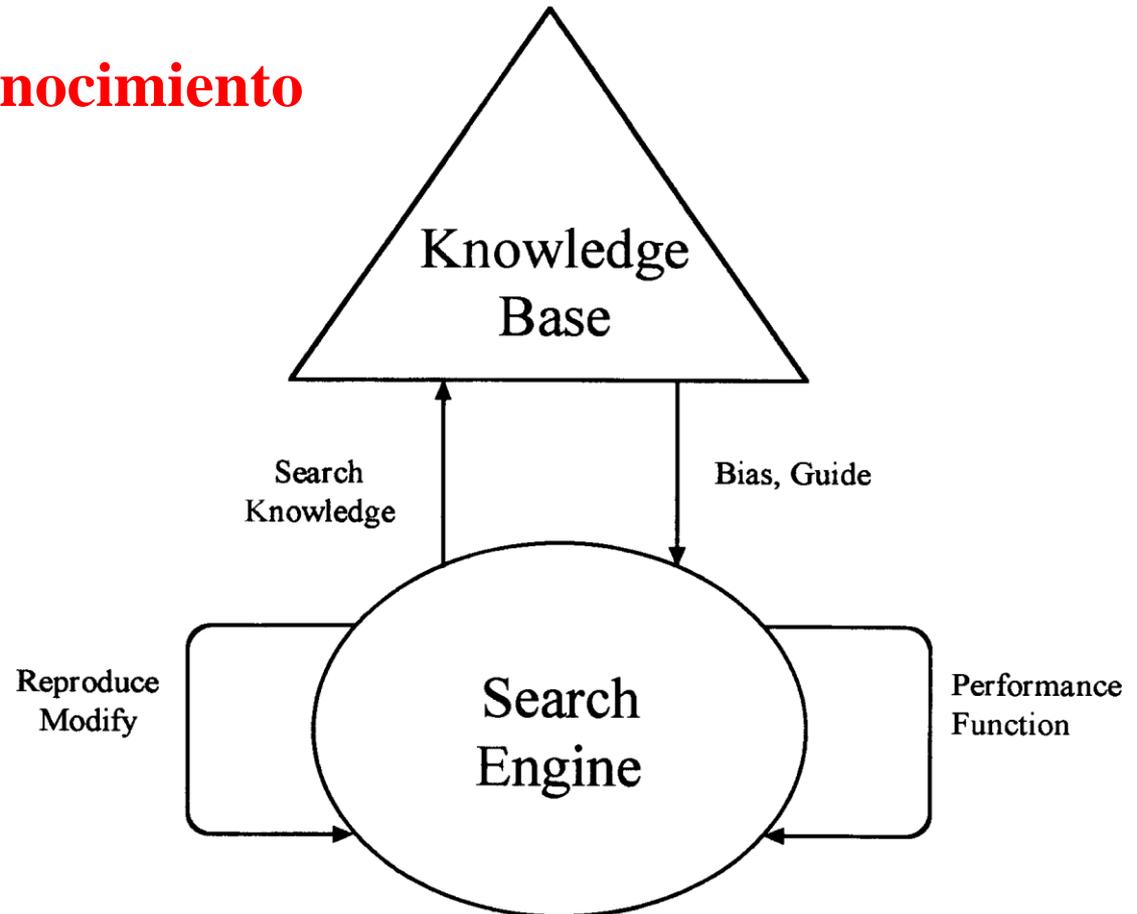
Pseudo-código de un AC

Sistema Híbrido

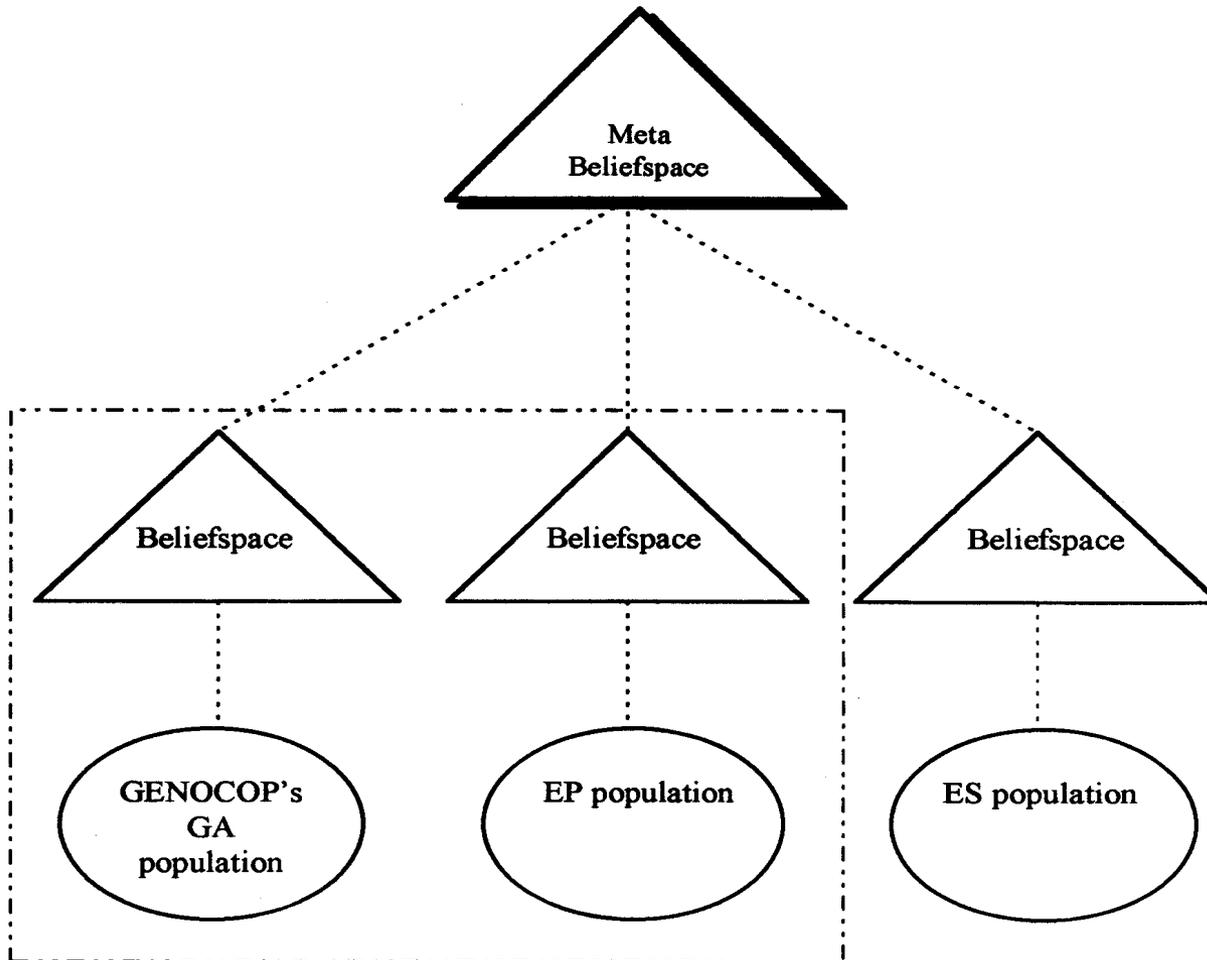
Búsqueda

+

Basado en conocimiento



Permite la aparición de estructuras jerárquicas tanto en la creencia como en los espacios de población



Problemas adecuados

- Cantidad significativa de conocimiento de dominio (por ejemplo, problemas de optimización con restricciones).
- Sistemas complejos en los que la adaptación puede ser en varios niveles a diferentes ritmos en la población y el espacio de creencias.
- El conocimiento está en diferentes formas y necesita ser razonado de diferentes maneras.
- Sistemas híbridos que requieren una combinación de búsqueda basada en conocimiento.
- La solución de problemas requiere múltiples poblaciones y múltiples espacios de creencias e interacción.
- Entornos problemáticos estructurados jerárquicamente donde pueden surgir población y conocimiento estructurados jerárquicamente.

Diseño de Algoritmos Culturales

- Diseño del componente de conocimiento
 - Conocimientos ontológicos (conceptos comunes compartidos para un dominio)
 - Representación del conocimiento de restricción
 - Representación de la solución
 - ¿Cuál será modificado? **Función de actualización** para cada componente modificable.
 - Mantenimiento del Conocimiento
- 2. Diseño del componente de población
 - **Variables de estado** que determinan la solución
 - Cómo se usan esas variables para producir una estrategia o comportamiento de resolución de problemas.
 - **¿Cómo se evalúa tal comportamiento?**

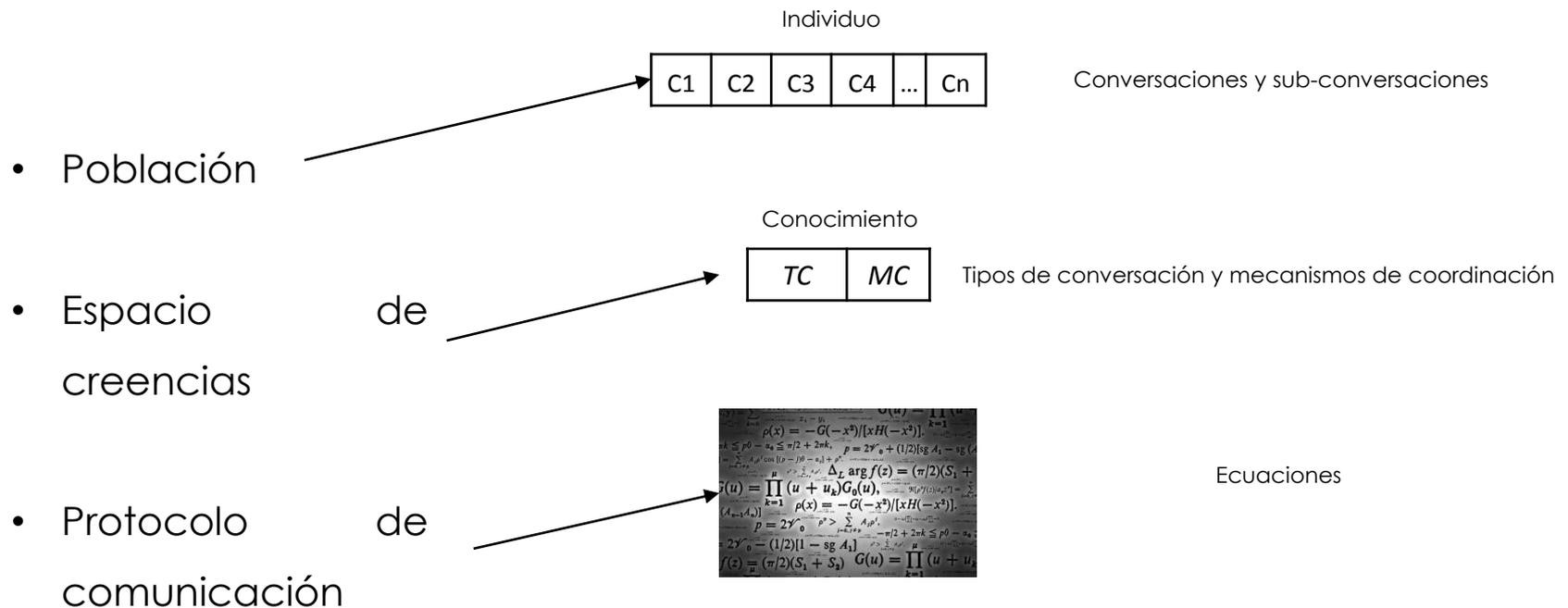
Algoritmos culturales (AC):

Espacio de creencias

- **Conocimiento normativo:** colección de rangos de valores deseables para los individuos de la población
- **Conocimiento del dominio específico:** información acerca del dominio del problema
- **Conocimiento situacional:** ejemplos específicos de eventos importantes,
- **Conocimiento temporal:** historial del espacio de búsqueda (registra lo recorrido del espacio de búsqueda), por ejemplo, patrones temporales del proceso de búsqueda
- **Conocimiento espacial:** información acerca de la topografía del entorno o espacio de búsqueda.

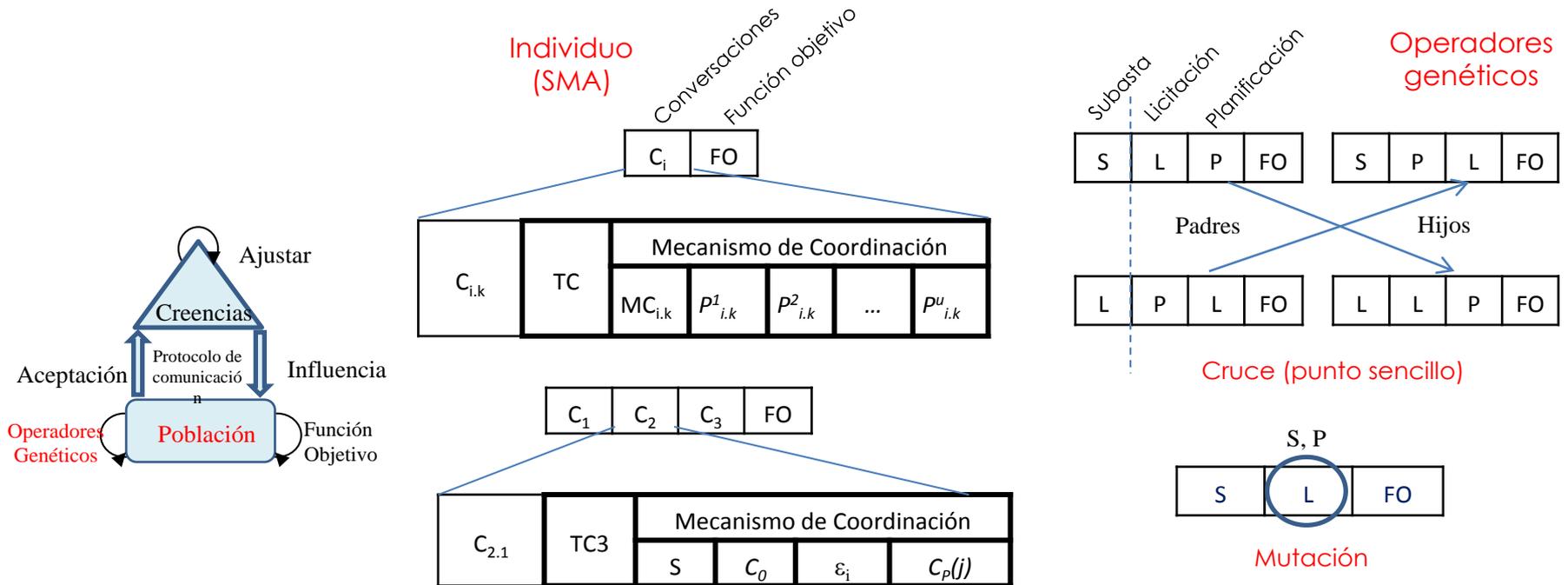
Algoritmos culturales (AC):

Caracterizar los componentes del AC



Algoritmos culturales (AC):

Población



Algoritmos culturales (AC):

Función objetivo

Permite evaluar el performance o desempeño de los individuos, esta basada en el costo de procesamiento (CP) y en el costo comunicacional (CC) de cada mecanismo de coordinación usado por el individuo (la idea es minimizar la función objetivo)

$$FO = \sum_{i=1}^n \sum_{k=1}^m (a * CP_{i,k} + b * CC_{i,k})$$

- Donde a y b son constantes definidas por el usuario y permiten normalizar las unidades de la función
- Conversaciones $i = \{1 \dots n\}$
- Sub-conversaciones $k = \{1 \dots m_i\}$

- CP y CC

Costo de Procesamiento

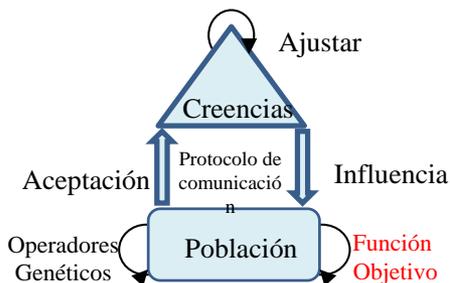
$$CP_{i,k} = PI_k + PE_k + \sum_{l=1}^j \sum_{q=1}^{m_l} A_{l,q}$$

- PI fijación del precio inicial; especificación de condiciones en las que se requiere un servicio, generación de planes
- PE proceso de selección del agente ganador, asignación de planes
- $A_{l,q}$ tiempo para preparar propuestas, tiempo para generar planes parciales

Costo de Comunicación

$$CC_{i,k} = \sum_{l=1}^j (\sum_{r=1}^{N-1} CEP_{l,r} + \sum_{s=1}^{n_j} CEO_{l,s}) + \sum_{r=1}^{N-1} CS_r$$

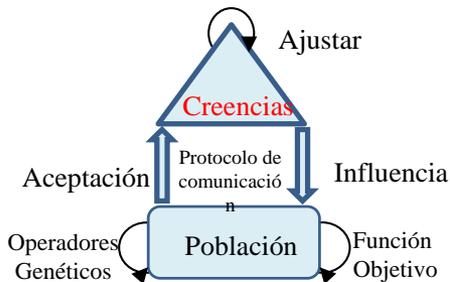
- CEP costo de envío de propuesta
- CEO costo de envío de ofertas, envío de planes parciales
- CS costo de informar al ganador, de enviar el plan global



Algoritmos culturales (AC):

Espacio de creencias

- **Conocimiento normativo:** colección de rangos de valores deseables para los individuos de la población
- **Conocimiento situacional:** ejemplos específicos de eventos importantes,



Conocimiento Circunstancial

Ejemplos específicos de eventos importantes, e. g., experiencias exitosas

TC	MC	$IO_{(TC, MC, t-1)}$	$TO_{(TC)}$
------	------	----------------------	-------------

- IO es el índice de ocurrencias
- TO es el total de ocurrencias

Conocimiento Normativo

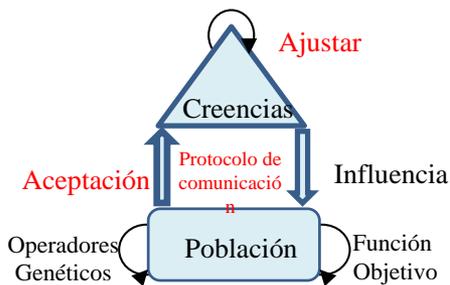
Son rangos de valores idóneos de cada una de las variables que integran al mecanismo de coordinación usado por los individuos

	p^1		p^2		...	p^u	
MC	LI	LS	LI	LS		LI	LS

- LI, LS son los límites inferiores y superiores de cada variable P^u

Algoritmos culturales (AC):

Protocolo de comunicación: función aceptación



Conocimiento Circunstancial

$$IO_{(TC,MC,t)} = IO_{(TC,MC,t-1)} + \left(\frac{NO_{(TC,MC,t)}}{TO_{(TC)}} \right)$$

$$TO_{(TC)} = TO_{(TC)} + \sum_{i=1}^k NO_{(TC,MC,t)i}$$

TC, tipo de conversación
MC, mecanismo de coordinación
t, estado actual

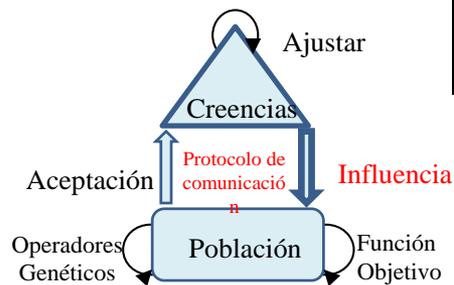
Conocimiento Normativo

$$Lac(P^u) = \left[\frac{(lv + \bar{P})}{2} \right]$$

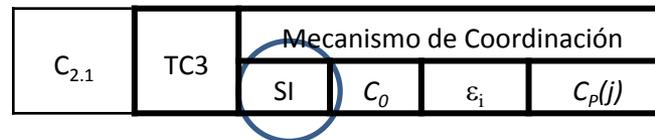
Lac, limite actual
Lv, limite anterior

Algoritmos culturales (AC):

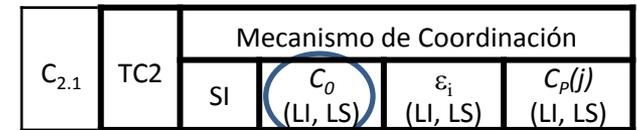
Protocolo de comunicación: función influencia



Conocimiento Circunstancial en el Individuo



Conocimiento Normativo en el Individuo

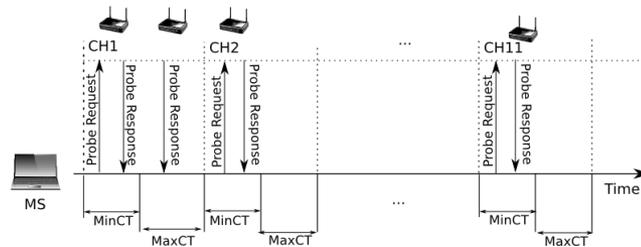


Mutación dirigida

Algoritmos culturales (AC):

Scanning sequences optimization

- The IEEE 802.11 Standard does not specify values for MinCT, MaxCT, neither the order in which the channels of 2.4GHz band are scanned.
- This is the main reason because the wireless network interface manufacturers implement different strategies to scan the channels.
- A complete sequence of all channels in a particular order is known as a full scanning.



Optimisation model:

$$\text{Max Discovery Rate} = \sum_{i=1}^{11} N_{\text{min}_{C_i}} / \text{MinCT}_{C_i} + N_{\text{max}_{C_i}} / \text{MaxCT}_{C_i}$$

$$\text{Min Latencia} = \sum_{i=1}^{11} (\text{MinCT}_{C_i} + p_i \text{MaxCT}_{C_i}) \quad \forall C_i \in [1; 11]$$

$N_{\text{min}_{C_i}}$ and $N_{\text{max}_{C_i}}$: number of APs discovered in channel i

p_i probability of having at least one response channel I

C_i represents the i channel from a total of 11 channels in the 2.4 GHz band

Algoritmos culturales (AC):

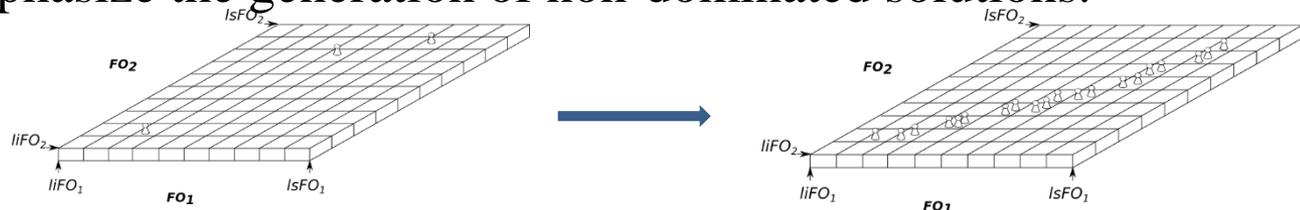
Structure of the population:

$$\langle X1; \dots; X11 \rangle \langle OF1; OF2 \rangle,$$

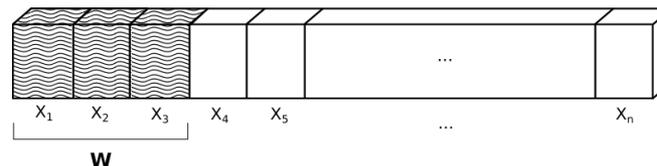
each gene is $\langle Ci; \text{MinCTi}; \text{MaxCTi}; \text{APi} \rangle$

Structure of the belief space:

- a phenotype normative to keep the lowest and highest limits of the ranges for each objective function,
- a grid to emphasize the generation of non-dominated solutions.



- a super-individual, in which the best values of genes of all individuals in the population are stored



Algoritmos culturales (AC):

Data: wireless topology model

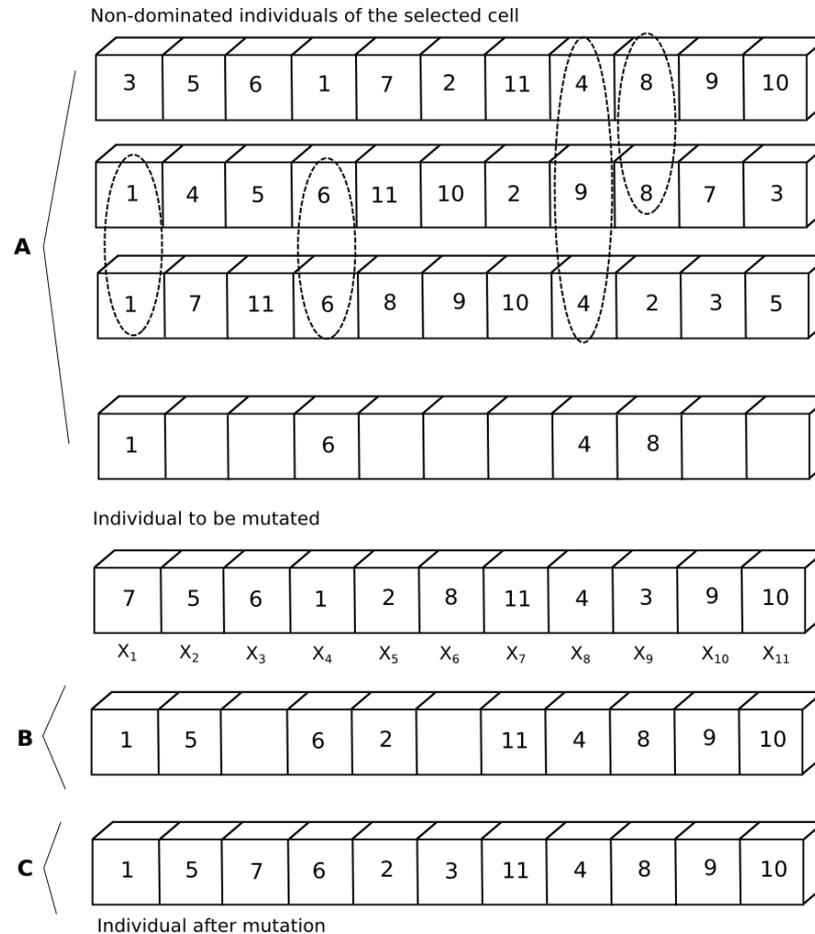
Result: adapted (individuals) scanning sequences

- 1 Initialise with P random individuals;
- 2 Compute OF1 and OF2 for every individual;
- 3 Compute Superindividual;
- 4 Initialise belief space;
- 5 while $i \leq \text{Total Generations}$ do
 - 6 Mutate individuals into $2P$ population P_{new} ;
 - 7 Filter P_{best} individuals through tournaments on P_{new} ;
 - 8 Add non-dominated individuals (P_{nd}) of P_{best} into elite-DB;
 - 9 Update grid using P_{nd} ;
 - 10 Update phenotypical component every $M < \text{Total Generations}$;
 - 11 Update Superindividual;
 - 12 end

Algoritmos culturales (AC):

NEW MUTATION PROCESS

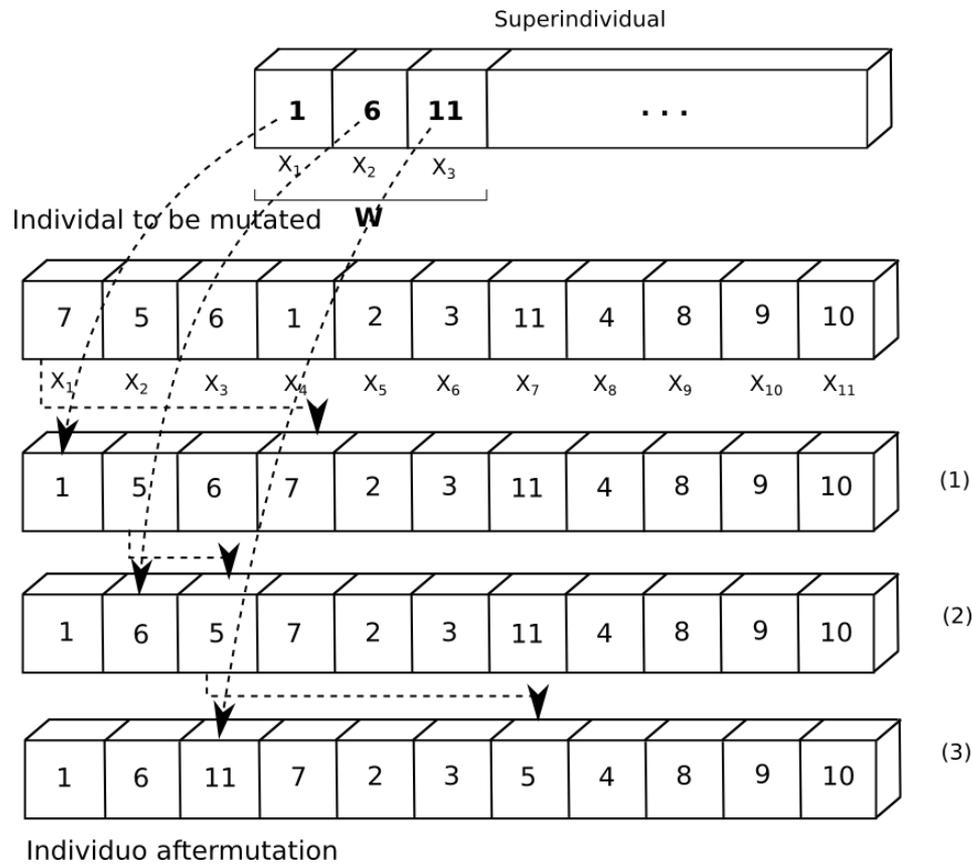
A. Mutation to Explore the Pareto Front



Algoritmos culturales (AC):

NEW MUTATION PROCESS

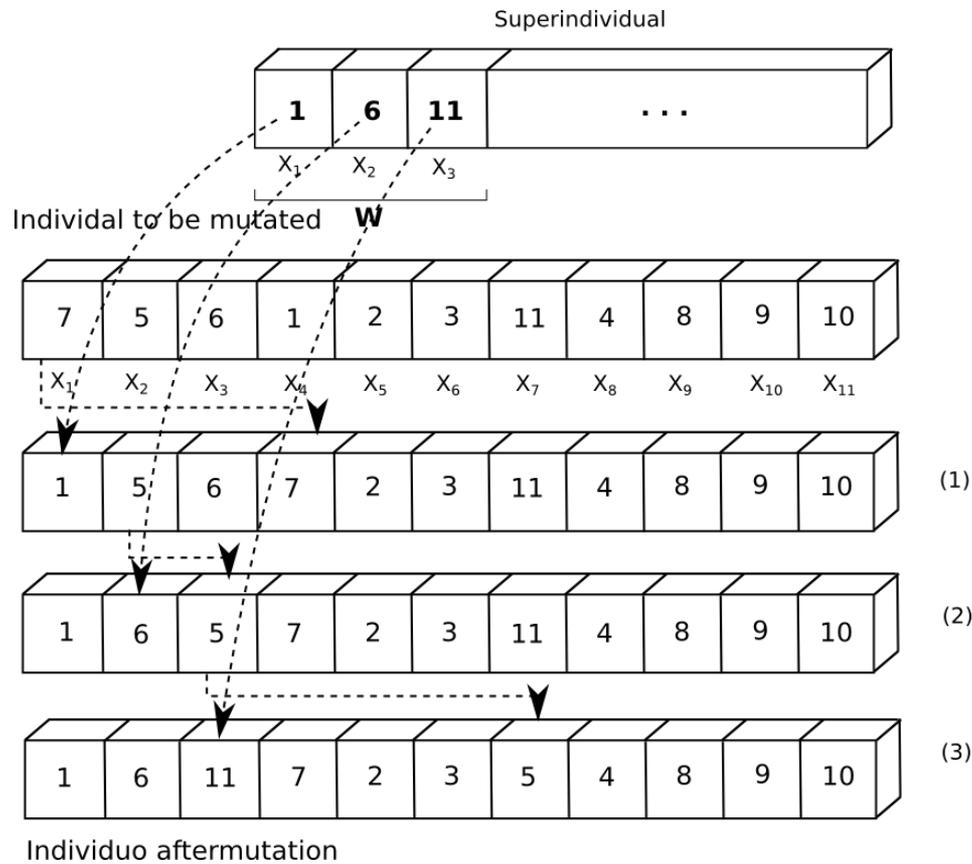
B. Directed mutation using superindividuals



Algoritmos culturales (AC):

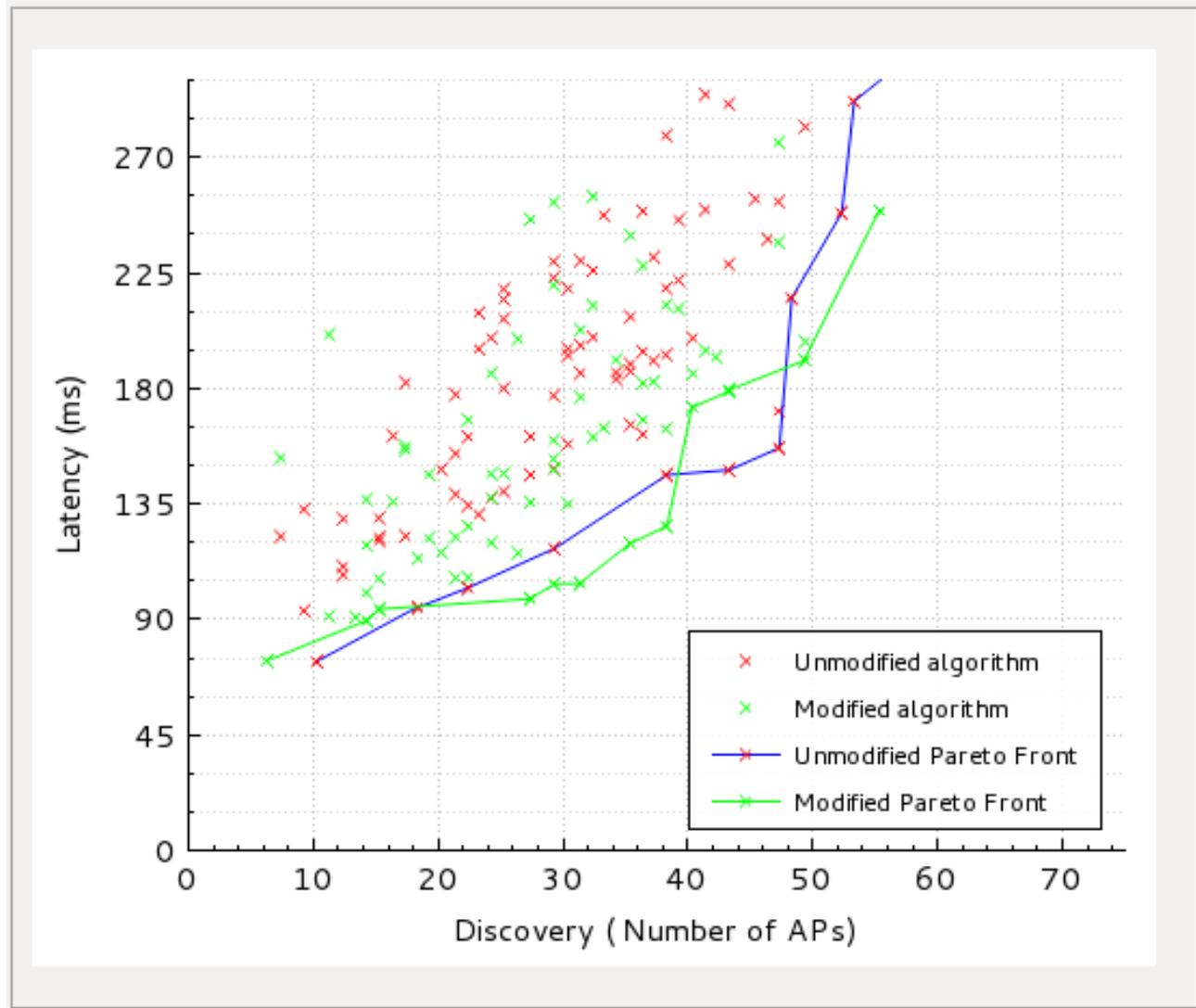
NEW MUTATION PROCESS

B. Directed mutation using superindividuals

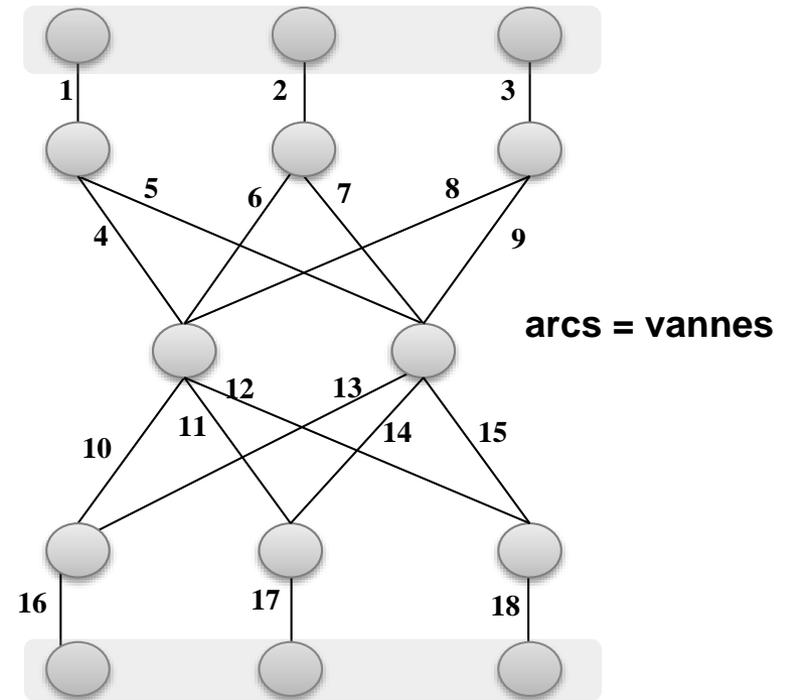
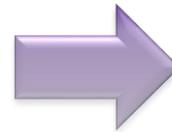
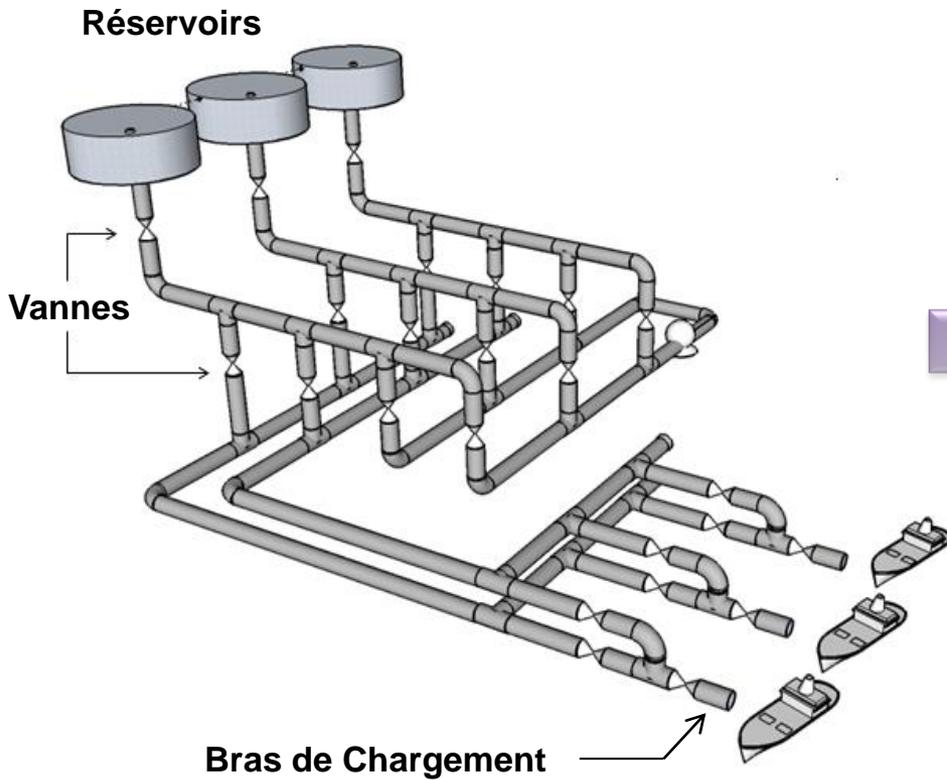


Algoritmos culturales (AC):

Resultados



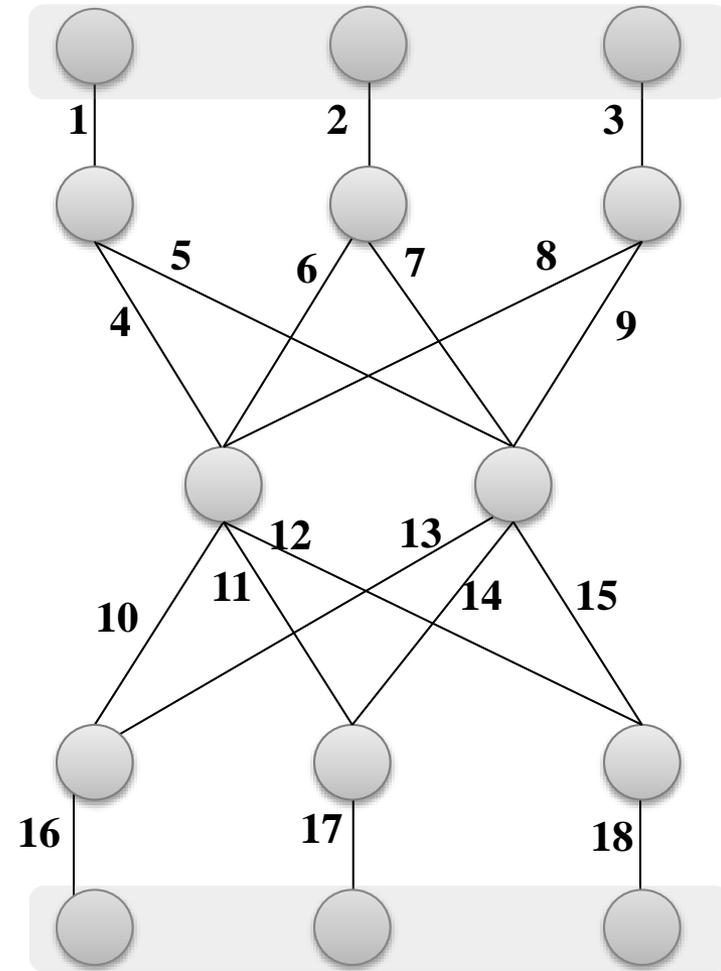
Optimisation de l'allocation des ressources pour les opérations de livraison dans un terminal maritime pétrolier: ordonnancement des opérations et sélection des alignements pour ces livraisons



Cas d'Étude et Problématique

Ressources:

Vannes

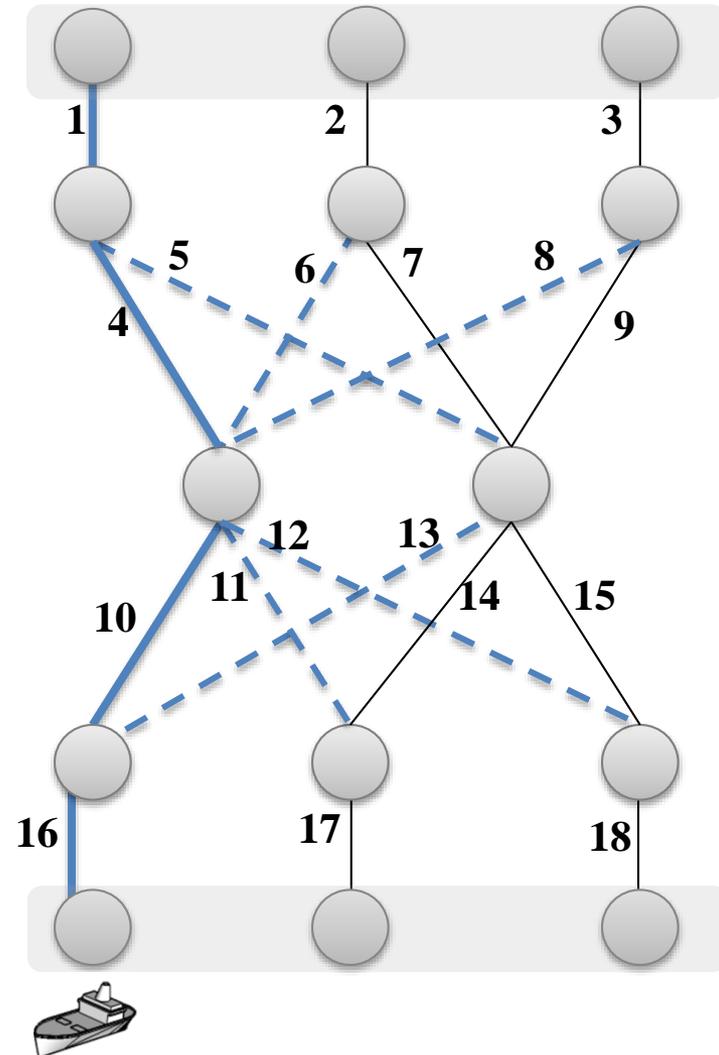


Cas d'Étude et Problématique

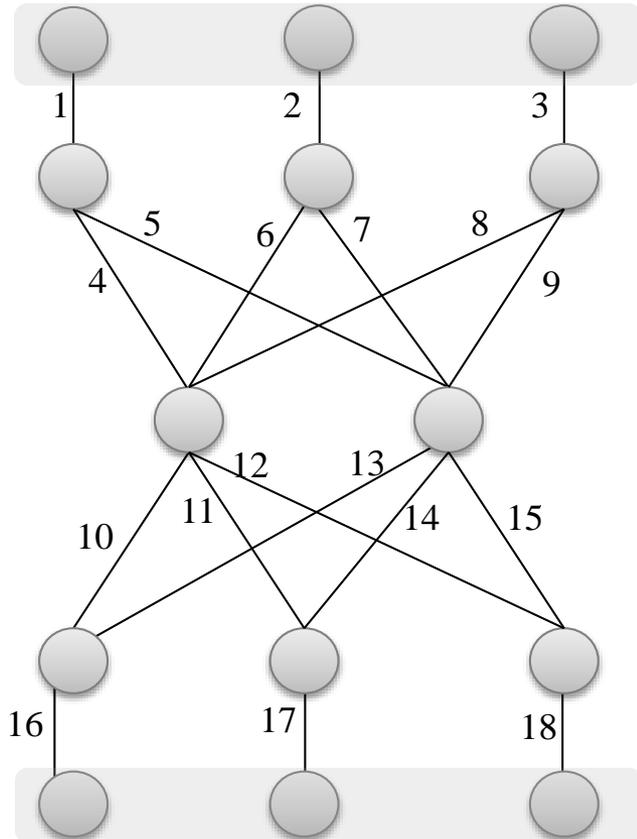
Alignement:

Engagement des vannes par:

- Ouverture des vannes dans l'alignement
- Fermeture des vannes adjacentes

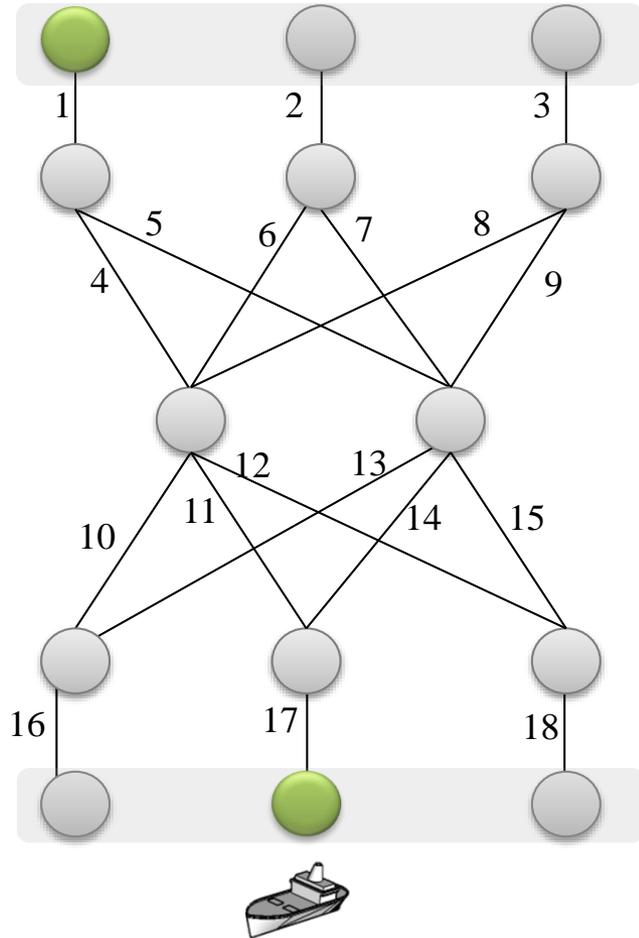


Cas d'Étude et Problématique



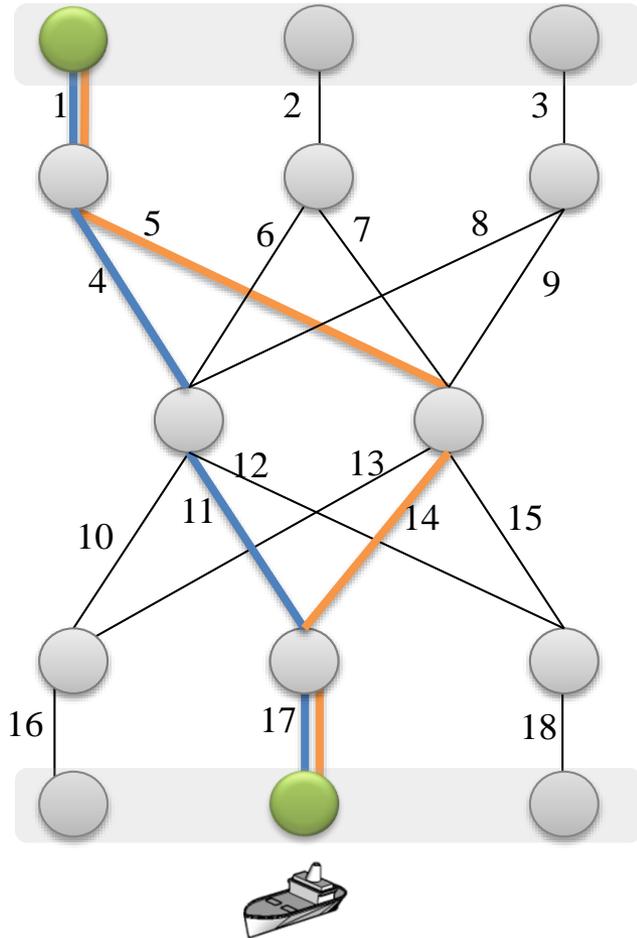
- N requêtes / fenêtres de temps pour arrivée des clients / ressources limitées
- Spécifications pour chaque requête
- Sélection d'alignement par requête

Cas d'Étude et Problématique



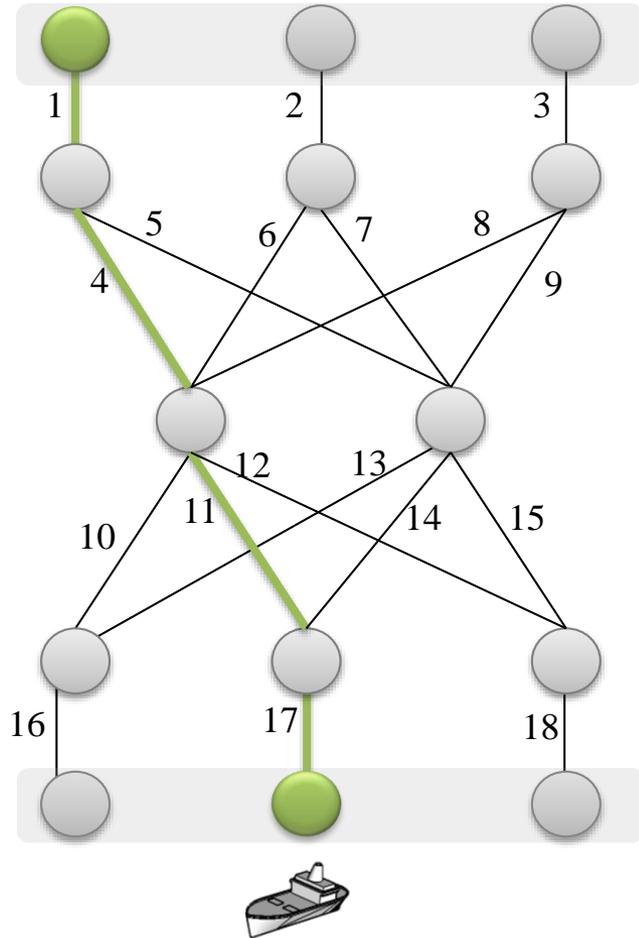
- N requêtes / fenêtres de temps pour arrivée des clients / ressources limitées
- Spécifications pour chaque requête
- Sélection d'alignement par requête

Cas d'Étude et Problématique



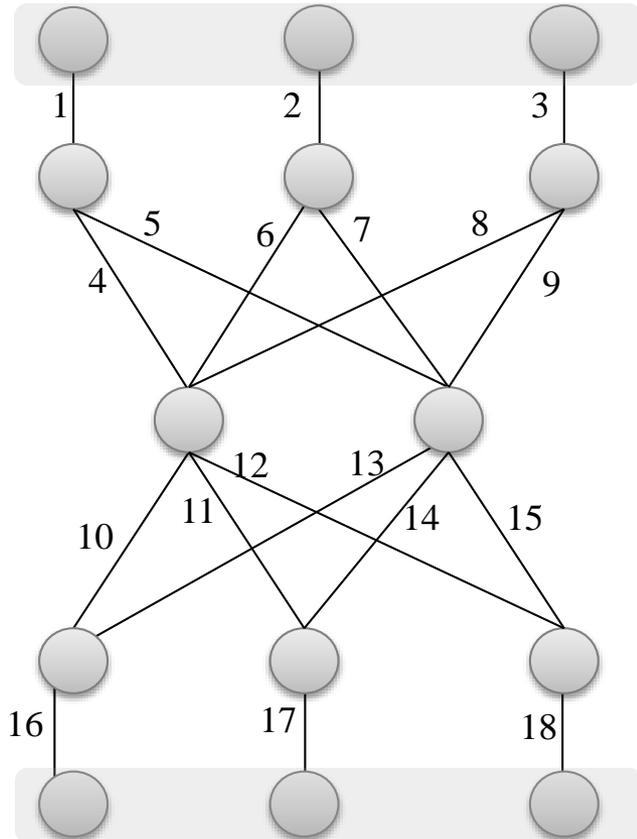
- N requêtes / fenêtres de temps pour arrivée des clients / ressources limitées
- Spécifications pour chaque requête
- Sélection d'alignement par requête

Cas d'Étude et Problématique



- N requêtes / fenêtres de temps pour arrivée des clients / ressources limitées
- Spécifications pour chaque requête
- Sélection d'alignement par requête

Cas d'Étude et Problématique

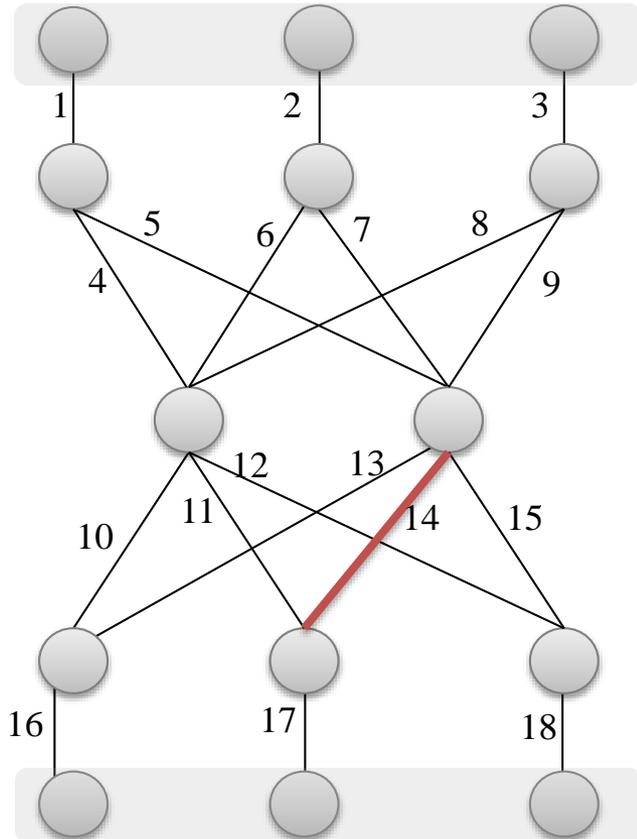


- N requêtes / fenêtres de temps pour arrivée des clients / ressources limitées
- Spécifications pour chaque requête
- Sélection d'alignement par requête
- Pénalités pour service tardif



Milliers de \$/heure par client

Cas d'Étude et Problématique



- N requêtes / fenêtres de temps pour arrivée des clients / ressources limitées
- Spécifications pour chaque requête
- Sélection d'alignement par requête
- Pénalités pour service tardif
- Opérations de Maintenance



Milliers de \$/heure par client

Solución

Contraintes: Dateurs des Opérations

$$x_i = \left(t_0 \oplus u_i \oplus \left(\oplus_k (x_{m_k} \otimes t_{m_k} \otimes V_{i,j;k}) \right) \oplus \left(\oplus_{i'j'} (x_{i'j'} \otimes p_{i'} \otimes V_{i,j;i'j'}) \right) \right) \otimes W_{ij}$$

$$x_{m_k} = x_{f_k} \oplus \left(\oplus_{i,j} (x_{i,j} \otimes p_i \otimes V_{k;i,j}) \right)$$

Objectifs: Minimisation des Pénalités et des Retards dans les Opérations de Maintenance

$$TCP = \sum_i (delay_i \times c_i)$$

$$TME = \sum_k (x_{m_k} - x_{f_k})$$

TME: « Total Maintenance Error »

Approche



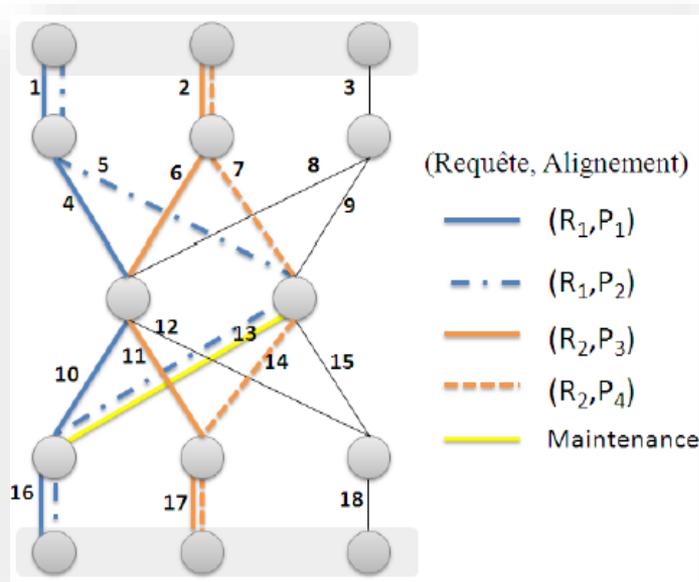
Solución

Modélisation (max,+) des contraintes fondamentales

Définition de la Structure Génomique

Définition du Comportement des Opérateurs Génétiques

Définition de la Stratégie Évolutionnaire



$$x_{1,1} = ((x_{2,3} \otimes p_2 \otimes V_{1,1;2,3}) \oplus (x_{m_{13}} \otimes tm_{13} \otimes V_{1,1;13})) \otimes W_{1,1}$$

$$x_{1,2} = ((x_{2,4} \otimes p_2 \otimes V_{1,2;2,4}) \oplus (x_{m_{13}} \otimes tm_{13} \otimes V_{1,2;13})) \otimes W_{1,2}$$

$$x_{2,3} = (x_{1,1} \otimes p_1 \otimes V_{2,3;1,1}) \otimes W_{2,3}$$

$$x_{2,4} = ((x_{1,2} \otimes p_1 \otimes V_{2,4;1,2}) \oplus (x_{m_{13}} \otimes tm_{13} \otimes V_{2,4;13})) \otimes W_{2,4}$$

$$x_{m_{13}} = x_{f_{13}} \oplus (x_{1,1} \otimes p_1 \otimes V_{13;1,1}) \oplus (x_{1,2} \otimes p_1 \otimes V_{13;1,2}) \oplus (x_{2,4} \otimes p_2 \otimes V_{13;2,4})$$

Solución

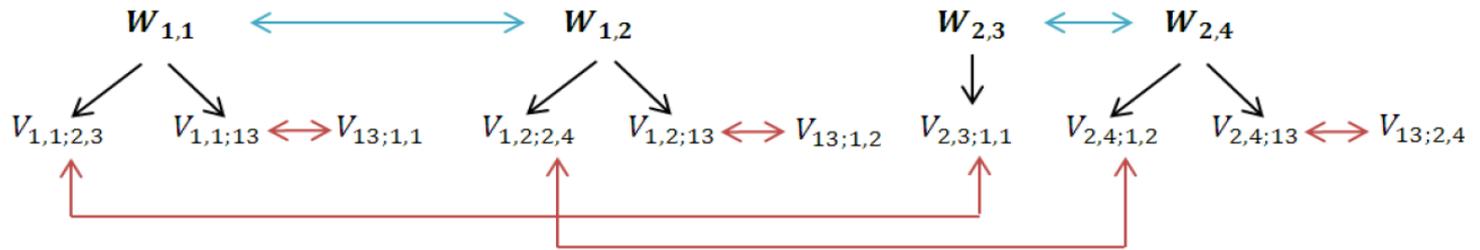
Modélisation (max,+) des contraintes fondamentales

Définition de la Structure Génomique

Définition du Comportement des Opérateurs Génétiques

Définition de la Stratégie Évolutionnaire

Structure Hiérarchique



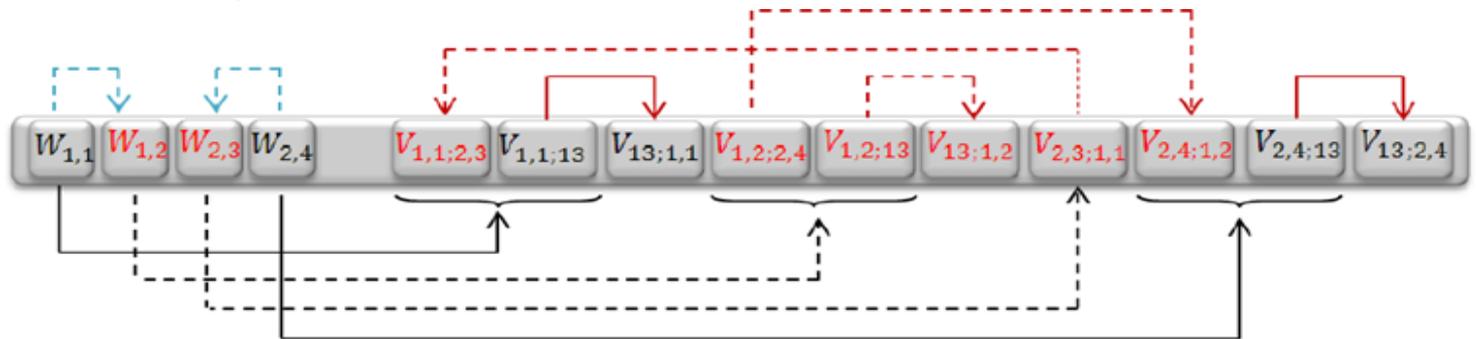
Gènes de Contrôle

Gènes de Paramétrage

Sélection d'Alignements

Ordonnancement

Génome

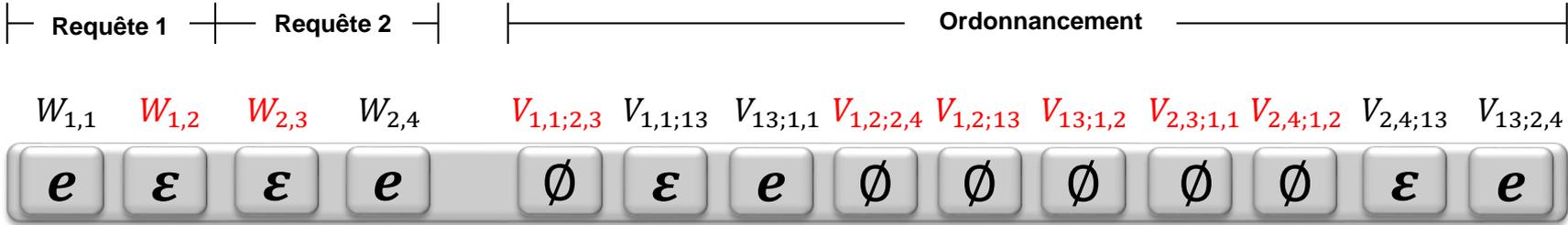


- $V_{i;j;j'}$: variable binaire et déterminant la précedence entre l'opération de transfert i (sur un certain alignement j) et l'opération en conflit i' (sur un certain alignement j').
- $W_{i;j}$: variable binaire déterminant l'affectation (ou pas) de l'alignement j à l'opération de transfert i .
- $V_{i;j;k}$: détermine la précedence entre l'opération de transfert i (sur un certain alignement j) et l'opération de maintenance sur la vanne k .

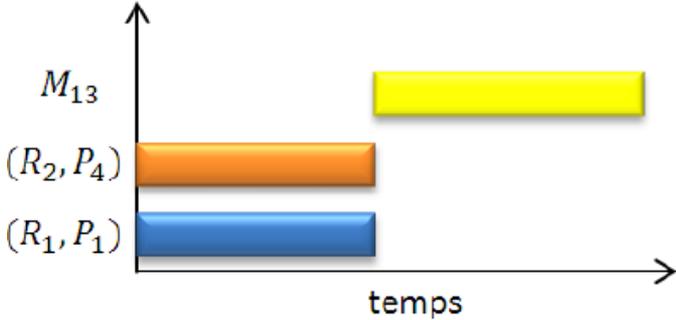
Solución



Codage



Solution Qualitative Encodée



$\varepsilon = \infty \quad e = 0$

Solución

Modélisation (max,+) des contraintes fondamentales

Définition de la Structure Génomique

Définition du Comportement des Opérateurs Génétiques

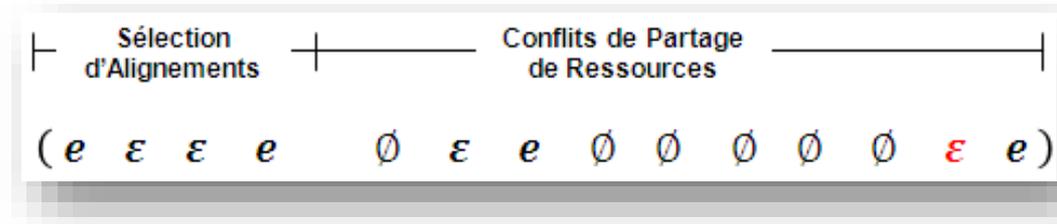
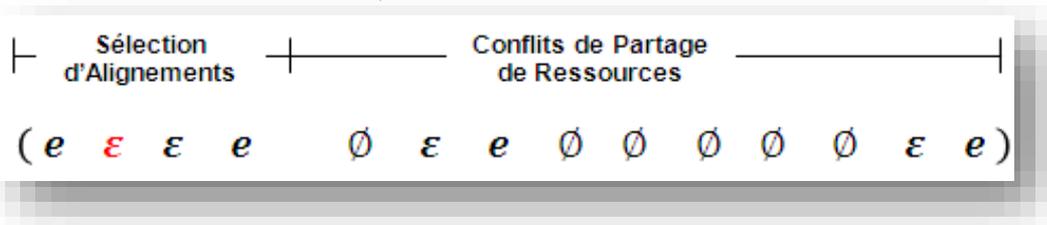
Définition de la Stratégie Évolutionnaire

Opérateur de Mutation



Mutation d'un gène de contrôle

Mutation d'un gène de paramétrage



$$\varepsilon = \infty \quad e = 0$$

Solución

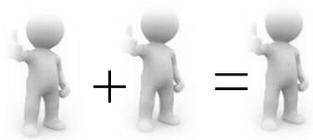
Modélisation (max,+) des contraintes fondamentales

Définition de la Structure Génomique

Définition du Comportement des Opérateurs Génétiques

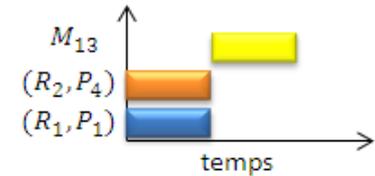
Définition de la Stratégie Évolutionnaire

Opérateur de Croisement



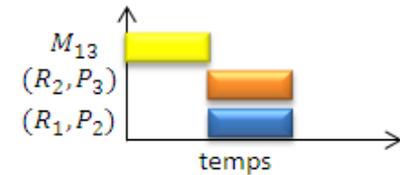
Parent P_1

$(e \ \varepsilon \ | \ \varepsilon \ e \ \emptyset \ \varepsilon \ e \ \emptyset \ \emptyset \ \emptyset \ \emptyset \ \emptyset \ \varepsilon \ e)$



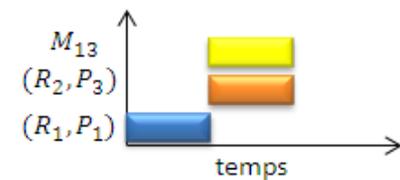
Parent P_2

$(\varepsilon \ e \ | \ e \ \varepsilon \ \emptyset \ \emptyset \ \emptyset \ \emptyset \ e \ \varepsilon \ \emptyset \ \emptyset \ \emptyset \ \emptyset)$



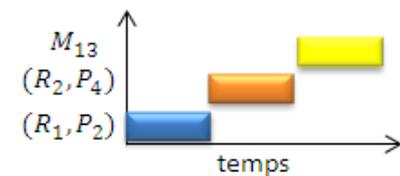
Descendant D_1

$(e \ \varepsilon \ e \ \varepsilon \ \varepsilon \ \varepsilon \ e \ \emptyset \ \emptyset \ \emptyset \ e \ \emptyset \ \emptyset \ \emptyset)$



Descendant D_2

$(\varepsilon \ e \ \varepsilon \ e \ \emptyset \ \emptyset \ \emptyset \ \varepsilon \ \varepsilon \ e \ \emptyset \ e \ e \ \varepsilon)$



Solución

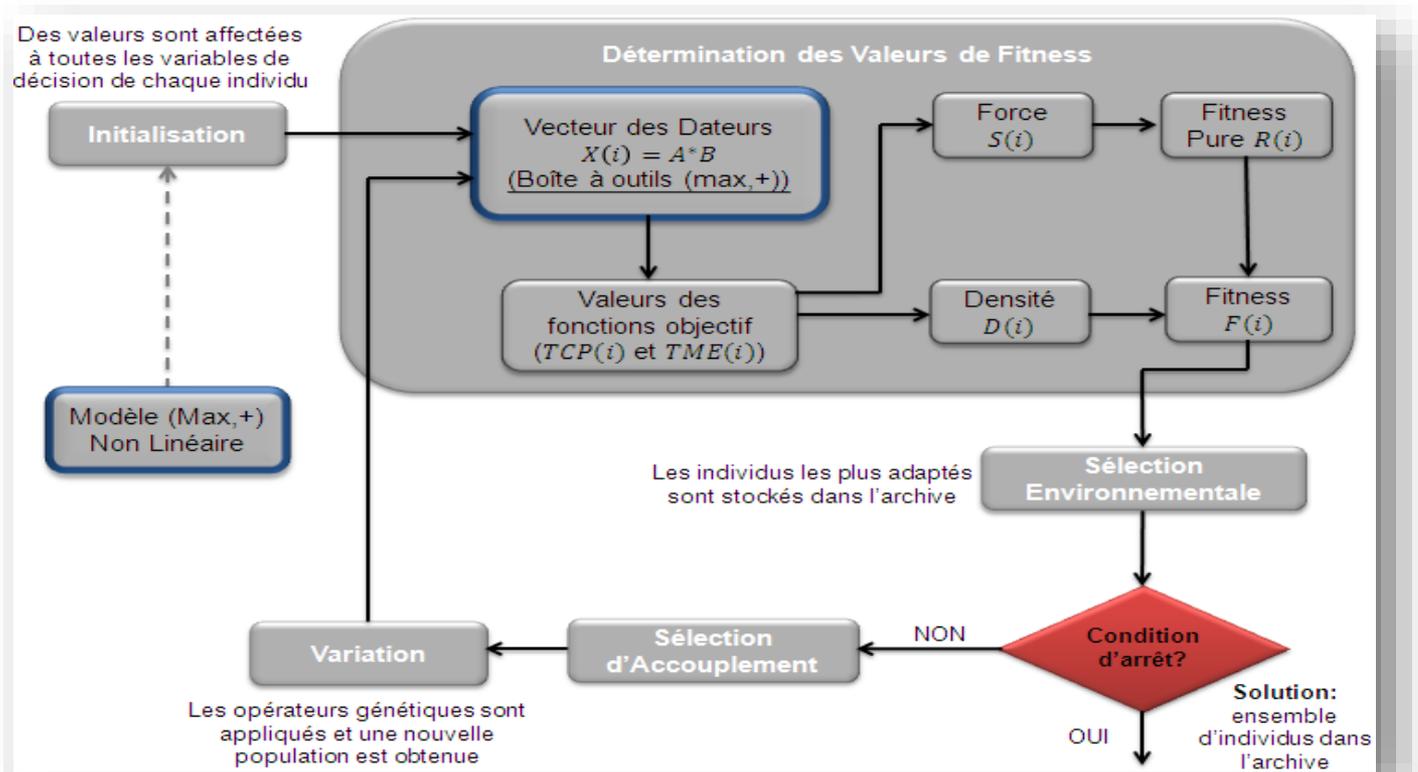
Modélisation (max,+) des contraintes fondamentales

Définition de la Structure Génomique

Définition du Comportement des Opérateurs Génétiques

Définition de la Stratégie Évolutionnaire

SPEA2 + Approche (max,+)



Profit d'un modèle analytique:



- Définition concise des contraintes

- Définition d'un contrôle original des gènes

SPEA2: Strength Pareto Evolutionary Algorithm

Solución

Modélisation (max,+) des contraintes fondamentales

Définition de la Structure Génomique

Définition du Comportement des Opérateurs Génétiques

Définition de la Stratégie Évolutionnaire

Validation

