Aplicación de Optimización de Enjambre de Partículas al Problema del Cajero Viajante bi-objetivo

Benjamín Barán Cegla
bbaran@cnc.una.py

Joaquín Quinto Lima Molinari
joaquinlima@gmail.com

Contenido

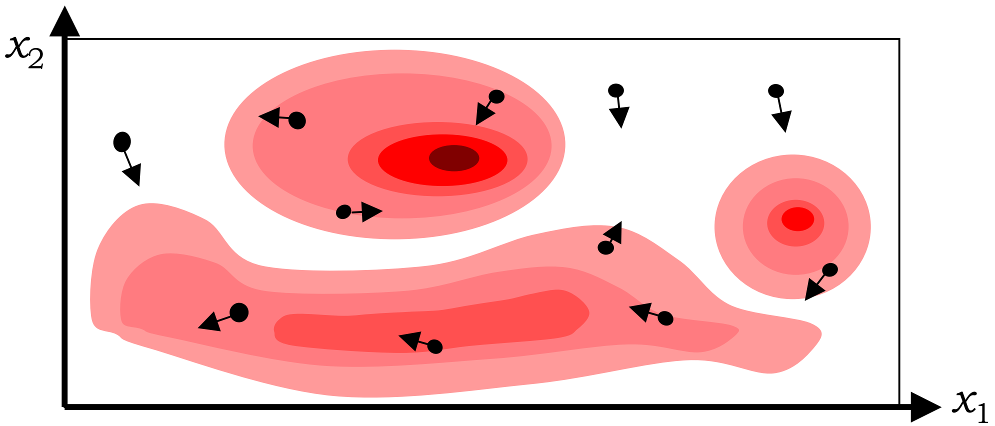
- Optimización de Enjambre de Partículas (PSO)
- Optimización Multiobjetivo
- Problema del Cajero Viajante (TSP)
- Adaptación al PSO para el TSP
- MultiObjective Particle Swarm Optimization (MOPSO)
- Resultados Experimentales
- Conclusiones y Trabajos Futuros

Optimización de Enjambre de Partículas

- *Particle Swarm Optimization* (PSO) es una metaheurística para de optimización de funciones continuas propuesta por R. Eberhart y J. Kennedy en 1995
- Originalmente diseñada para la simulación del movimiento de individuos observado en parvadas de pájaros, bancos de peces y enjambres de insectos.

Optimización de Enjambre de Partículas

- Algoritmo iterativo basado en una población de individuos artificiales que sobrevuela el espacio de decisión en busca de soluciones óptimas

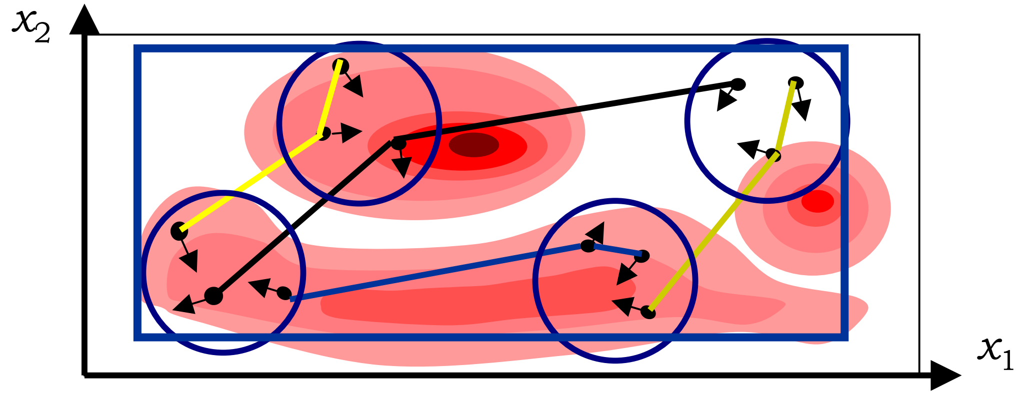


- Cada partícula está compuesta por los siguientes vectores:
  - $X = [x_1, x_2, \ldots, x_N]$, posición actual de la partícula,
  - $V = [v_1, v_2, \ldots, v_N]$, velocidad actual de la partícula,
  - $P = [p_1, p_2, \ldots, p_N]$, posición de la mejor solución encontrada.

Optimización de Enjambre de Partículas

- El movimiento de las partículas esta determinado en base a:
  - La velocidad actual de la partícula,
  - La mejor posición personal (información cognitiva), y
  - La mejor posición grupal (información social).

- La mejor posición grupal puede ser determinada de manera:
  - Global, o
  - Mediante un esquema de vecindad:
    - Geográfica, o
    - Social.

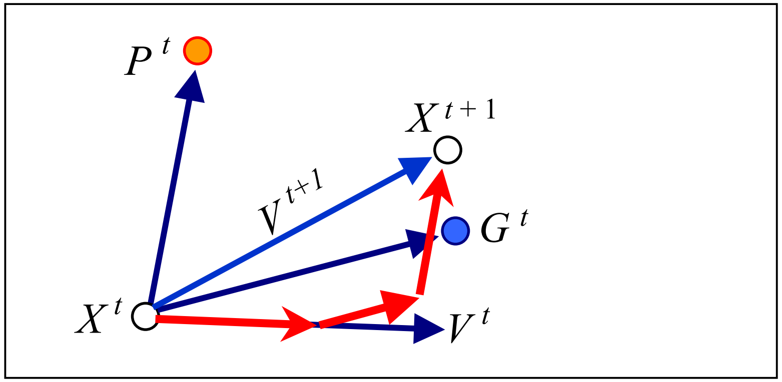


Optimización de Enjambre de Partículas

- La posición y velocidad se actualizan a cada iteración conforme a:

$$(1) \quad V^{t+1} = \boxed{\omega \times V^t} + \boxed{rand\left(0, C_1\right) \times \left(G^t - X^t\right)} + \boxed{rand\left(0, C_2\right) \times \left(P^t - X^t\right)}$$

$$(2) \quad X^{t+1} = \boxed{X^t + V^{t+1}}$$

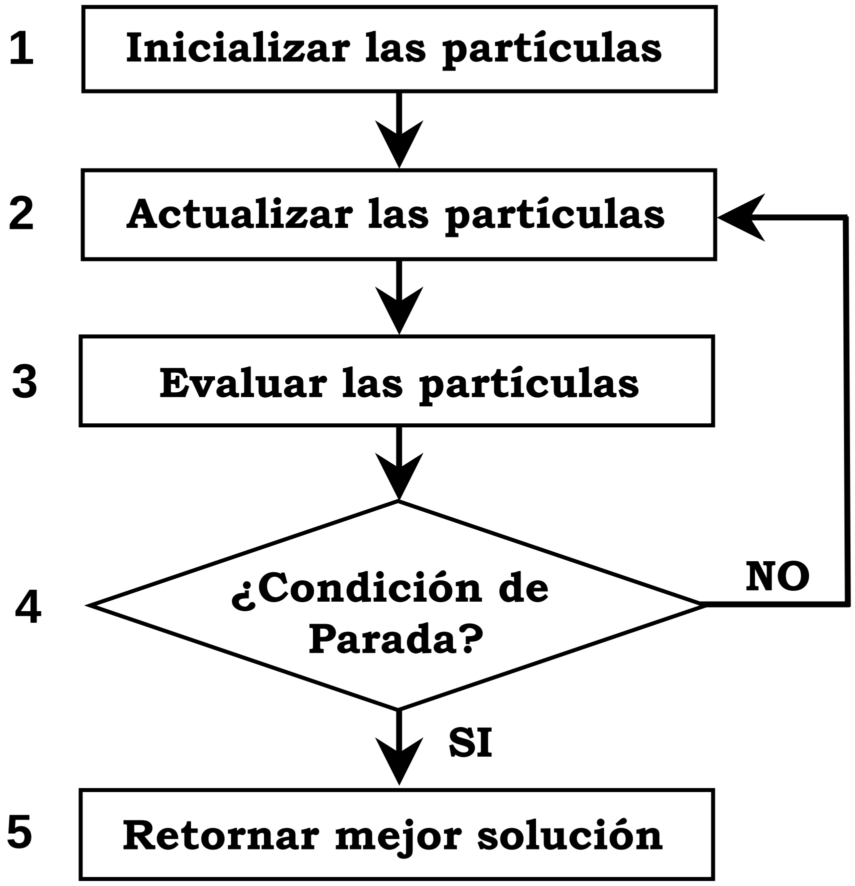


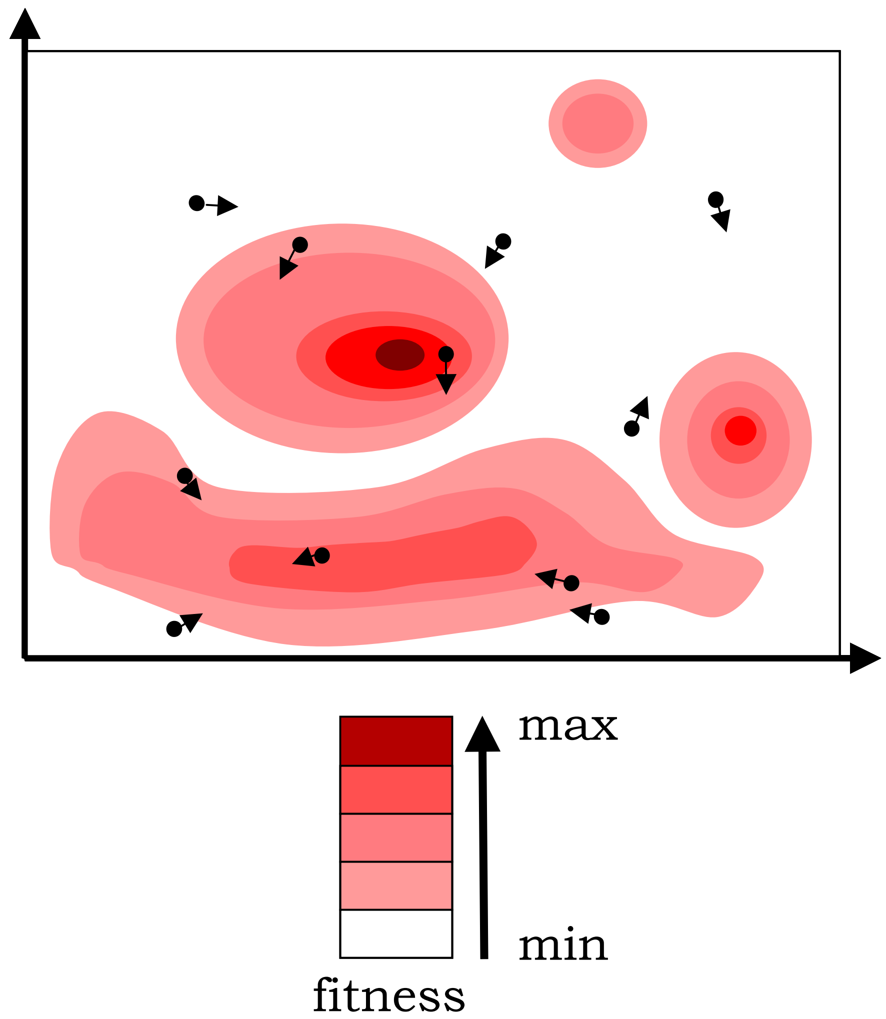
- Las mejores posiciones se actualizan:

$$Si\,(X^{t+1}\ es\ mejor\ \ P) \quad entonces \quad P := X^{t+1}$$

$$Si\ (X^{t+1}\ es\ mejor\ \ G) \quad entonces \quad G := X^{t+1}$$

Optimización de Enjambre de Partículas





Optimización Multiobjetivo

- Un problema de optimización multiobjetivo se define como:

$$\begin{array}{ll} \text{Optimizar} & \vec{f}(\vec{x}) = [f_1(\vec{x}), f_2(\vec{x}), \ldots, f_M(\vec{x})] \\ \text{Sujeto a} & \vec{g}(\vec{x}) = [g_1(\vec{x}), g_2(\vec{x}), \ldots, g_J(\vec{x})] \geq 0 \\ \text{y} & \vec{h}(\vec{x}) = [h_1(\vec{x}), h_2(\vec{x}), \ldots, h_K(\vec{x})] = 0, \\ \text{donde :} & \vec{x} = [x_1, x_2, \ldots, x_N]^{\mathrm{T}} \end{array}$$

- Espacio solución factible:

$$\Omega = \left\{ \vec{x} \in \Re^N \mid \vec{g}(\vec{x}) \geq 0 \wedge \vec{h}(\vec{x}) = 0 \right\}$$

$$\Omega_0 = \left\{ \vec{y} \in \Re^M \mid \vec{y} = \vec{f}(\vec{x}) \ \forall \vec{x} \in \Omega \right\}$$

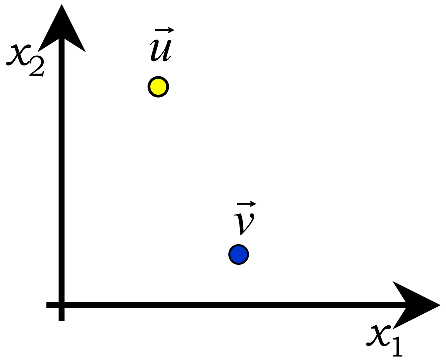
Optimización Multiobjetivo

- **Dominancia Pareto:**
  - En un caso de minimización se dice que:
    - ***u*** *domina a* ***v****,* si y solo si:

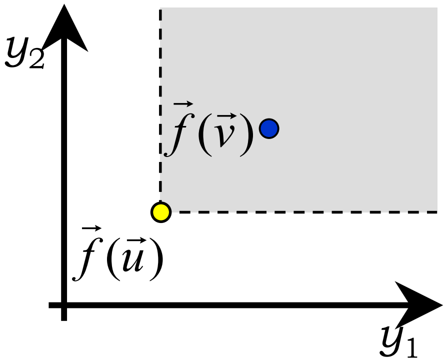
$$f_i(u) \leq f_i(v) \;\; \forall\, i \,\{1, \ldots, M\} \text{ y}$$
$$f_i(u) < f_i(v) \text{ para al menos un valor de } i$$

- ***u*** *es mejor que* ***v*** *en por lo menos un objetivo*

Denotado como: $\vec{u} \succ \vec{v}$



Espacio decisión

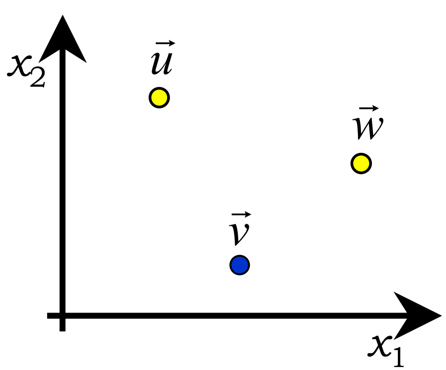


Espacio solución

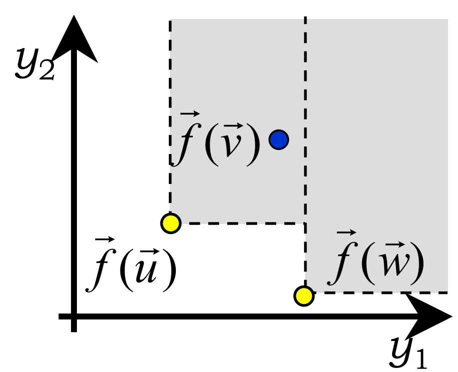
Optimización Multiobjetivo

- **Dominancia Pareto:**
  - En un caso de minimización se dice que:
    - ***u*** es *no comparable* con ***w***, si $\boldsymbol{u} \not\succ \boldsymbol{w}$ y $\boldsymbol{w} \not\succ \boldsymbol{u}$
    - ***u*** es mejor que ***w*** en algunos objetivos y ***w*** es mejor que ***u*** en otros objetivos

Denotado como: $\vec{u} \sim \vec{w}$



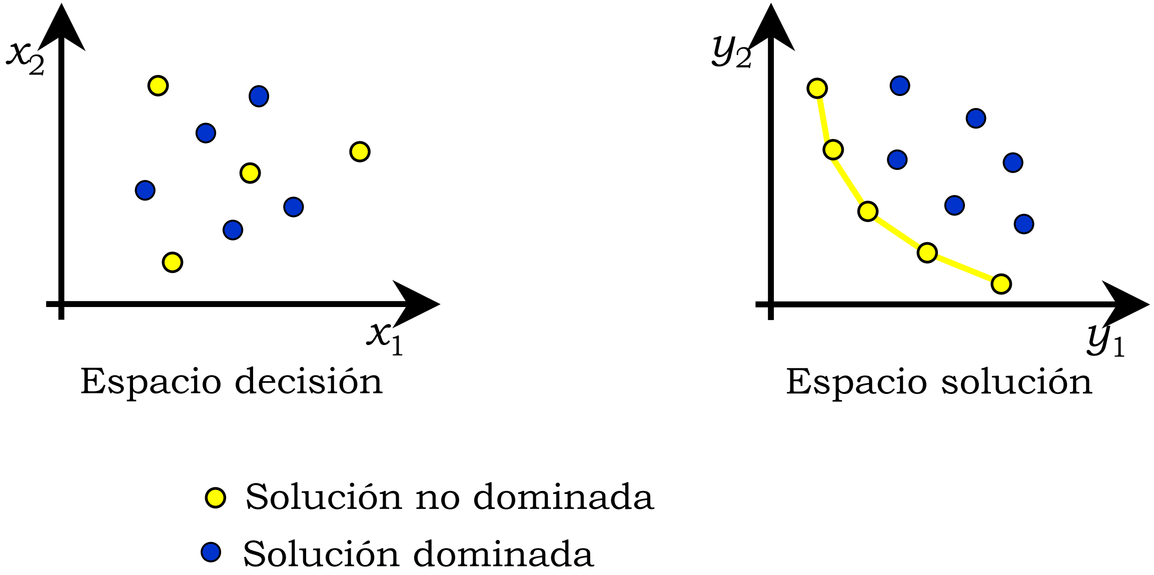
Espacio decisión



Espacio solución

Optimización Multiobjetivo

- Conjunto Pareto: $P = \left\{ \vec{x} \in \Omega \mid \nexists \vec{x}\,' \in \Omega \;\; para\ el\ cual \;\; \vec{x}\,' \succ \vec{x} \right\}$
- Frente Pareto: $FP = \left\{ \vec{y} \in \Omega_0 \mid \vec{y} = \vec{f}(x) \;\; \forall \vec{x} \in P \right\}$

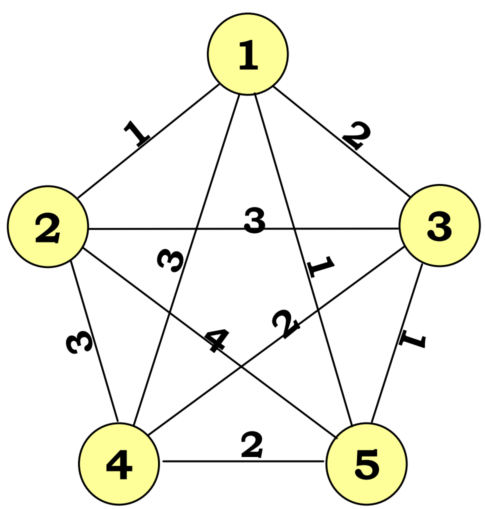


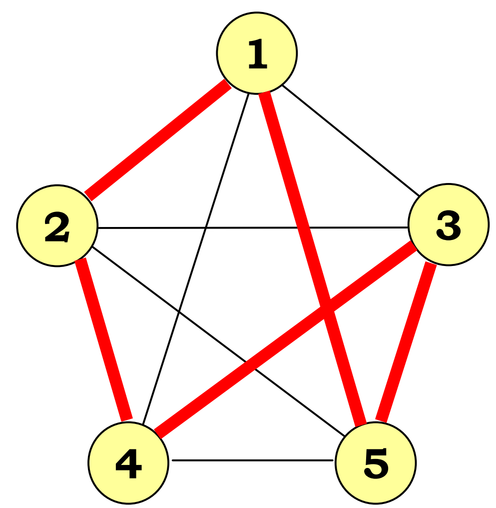
Problema del Cajero Viajante

- Dado un conjunto $C = \{ c_1, c_2, \ldots, c_N \}$ de N ciudades totalmente interconectadas, encontrar el camino de menor costo que inicie en una ciudad cualquiera, pase una vez por cada una de las ciudades restantes y finalice en la ciudad inicial.

$$\Omega = permutaciones(C)$$

$$minimizar \quad f(x) = \sum c_{ij} \quad \forall (v_i, v_j) \Rightarrow x \in \Omega.$$



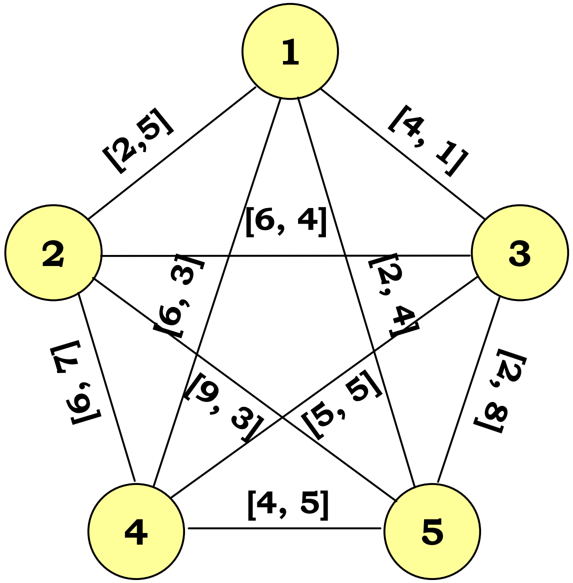


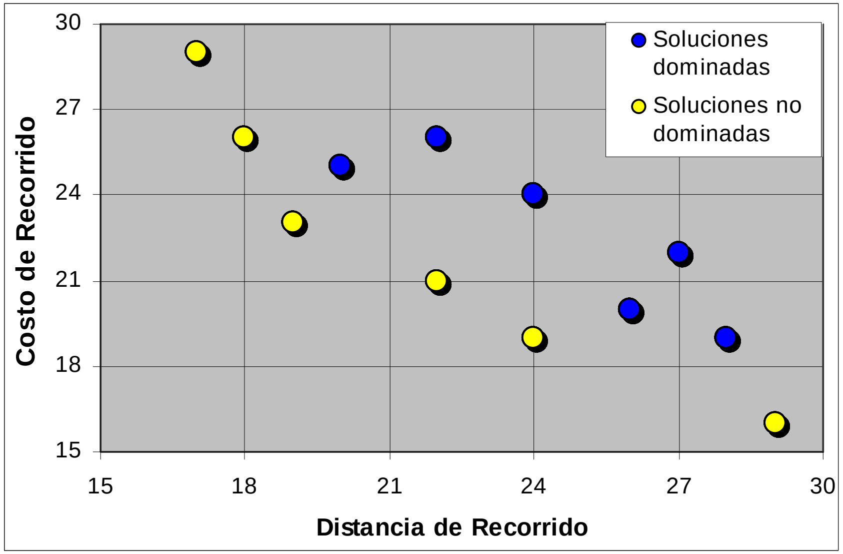
Problema del Cajero Viajante

- En problemas TSP de M objetivos (distancia, costo de recorrido, tiempo de recorrido, etc.), cada arista tiene asociada un vector de costo $\vec{c}_{ij} \geq 0$ de dimensión M.

$$minimizar \quad \vec{f}(x) = [f_1(x), f_2(x), \ldots, f_M(x)]$$

$$f_k(x) = \sum c_{ij}^k \quad \forall (v_i, v_j) \Rightarrow x \in \Omega \quad \forall k \in [1 \ldots M].$$

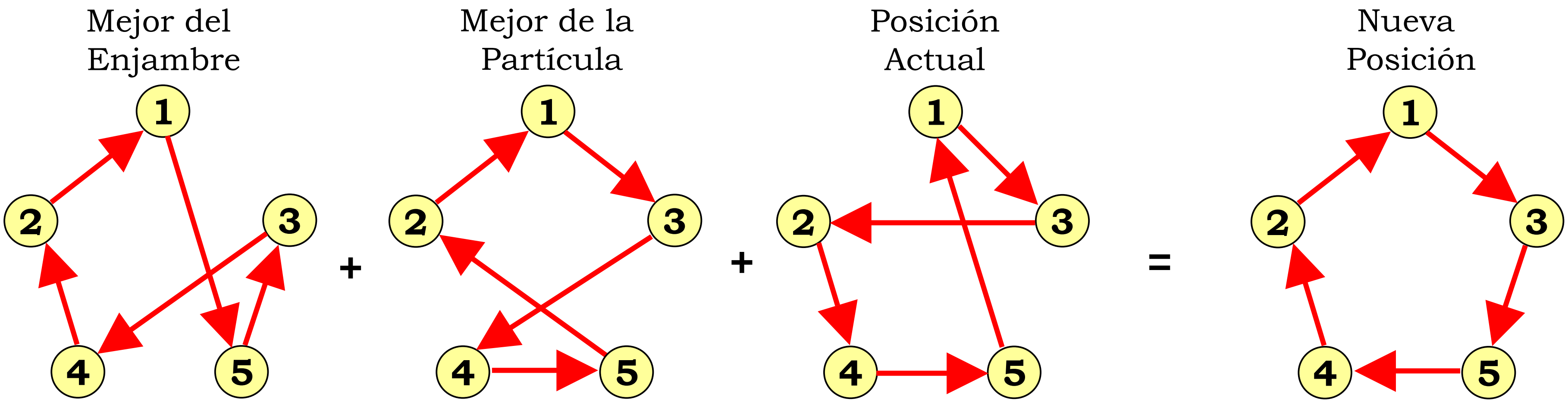




Adaptación del PSO para el TSP

Adaptación propuesta por Secrest (2001) :

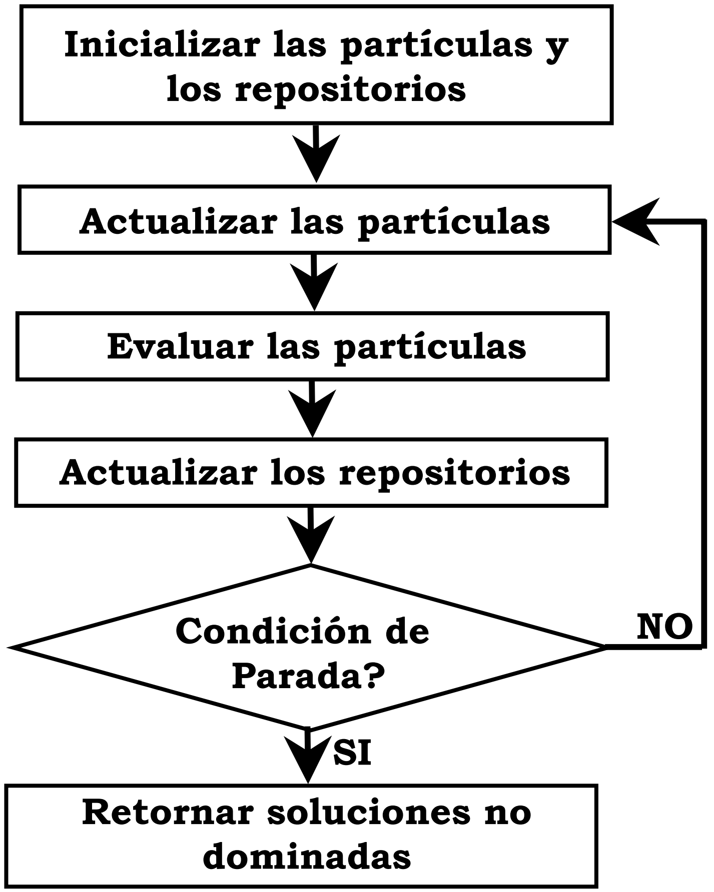
- Se reemplazan el factor social (C_1) y el factor cognitivo (C_2) por:
  - K_1 ➔ prob. de selec. sgte. vértice de la mejor pos. del Enjambre
  - K_2 ➔ prob. de selec. sgte. vértice de la mejor pos. de la partícula
  - K_3 ➔ prob. de selec. sgte. vértice de la posición actual
  - $K_1 + K_2 + K_3 := 100$



[3, 4, 2, 1, 5] [5, 2, 1, 3, 4] [4, 5, 1, 3, 2] [2, 1, 3, 5, 4]

Multiobjective Particles Swarm Optimization – MOPSO

- El resultado de la evaluación de las partículas es un vector que contiene la evaluación en todos objetivos
- Se tiene un repositorio global que almacena todas las soluciones no dominadas.
- Las partículas pueden recordar:
  - La última posición no dominada encontrada
  - Todas las posiciones no dominadas encontradas



Multiobjective Particle Swarm Optimization – MOPSO

- La diferencias entre los distintos MOPSOs son:
  - La manera en como se escoge la mejor solución grupal para las partículas.
  - El método en que se gestionan los distintos repositorios de las partículas y el repositorio global
- Los métodos MOPSOs implementados corresponden a:
  - Moore y Chapman (1999)
  - Hu y Eberhart (2002)
  - Coello y Lechuga (2002)
  - Fieldsend y Singh (2002)
  - Mostaghim y Teich (2003)

Resultados experimentales

- Se realizaron implementaciones de los MOPSOs y de los algoritmos basados en colonias de hormigas: MOACS, M-MMAS y PACO.
- El problema resuelto fue el KROAB100 obtenido de la librería TSPLIB
- Se realizaron 10 corridas de 200 s. para cada algoritmo. Los resultados fueron combinados para obtener el frente generado por cada algoritmo.
- Los resultados de todas las corridas fueron combinados de manera a obtener una aproximación al Frente Pareto real del problema.
- Se utilizaron las métricas:
  - M^*_1, M^*_2 y M^*_3 propuestas por Zitzler et al. (2000), que evalúan la calidad, distribución y extensión del frente generado por cada algoritmo
  - *Error* propuesta por Veldhiuzen (2000) que indica el porcentaje de soluciones que no pertenecen al frente Pareto real aproximado.

Resultados experimentales

Ranking de Evaluaciones:

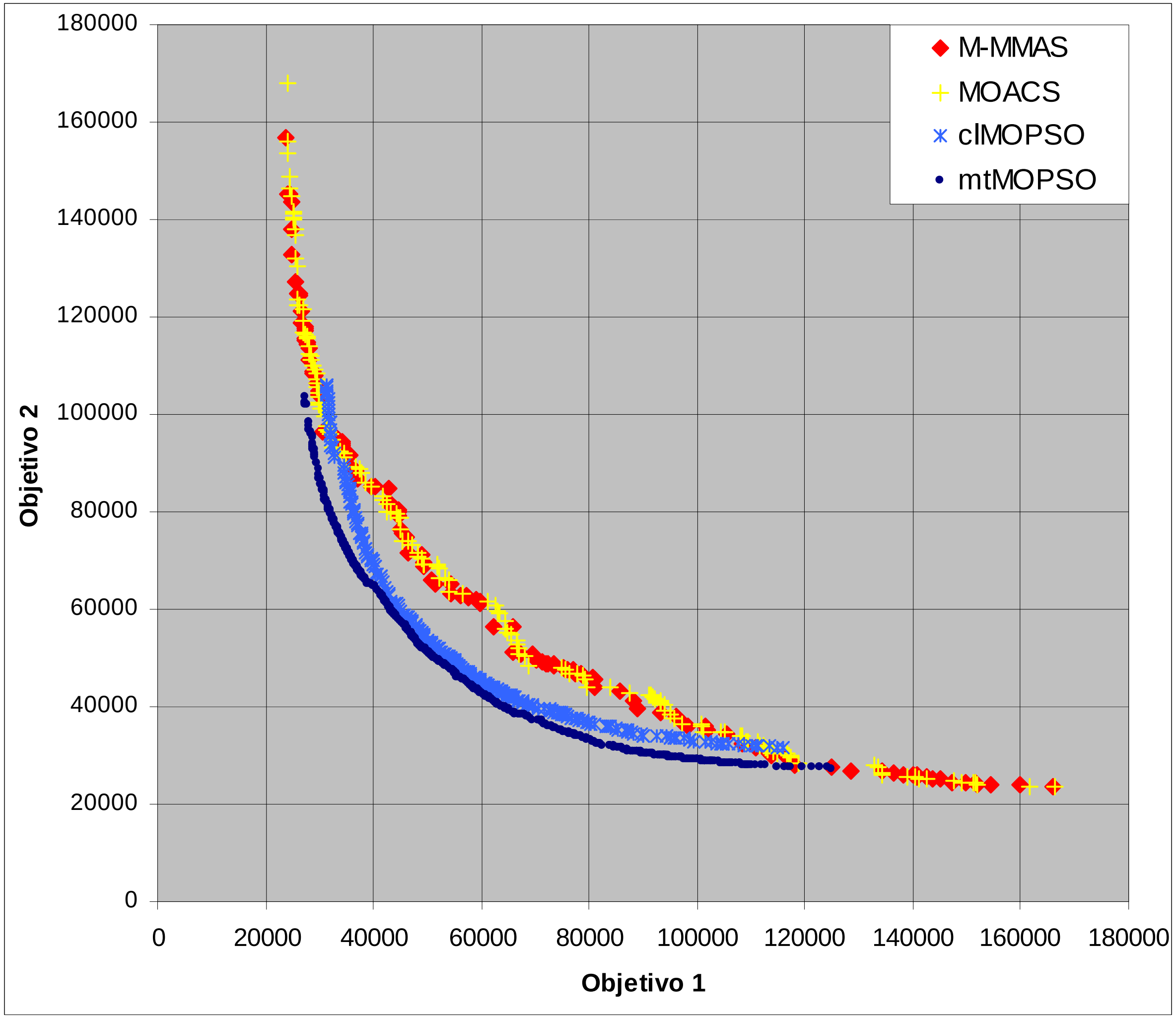
M_1^*		M_2^*		M_3^*		*Error*	
mtMOPSO	102,37	mtMOPSO	633,72	MOACS	535,51	mtMOPSO	21,07%
heMOPSO	644,41	mcMOPSO	289,13	M-MMAS	524,75	fsMOPSO	33,98%
clMOPSO	2.037,02	clMOPSO	230,89	fsMOPSO	442,49	heMOPSO	77,15%
fsMOPSO	2.044,95	fsMOPSO	230,11	mtMOPSO	417,19	M-MMAS	93,80%
mcMOPSO	2.797,82	heMOPSO	226,62	clMOPSO	399,24	MOACS	94,16%
PACO	4.810,19	MOACS	124,58	mcMOPSO	334,14	clMOPSO	100,00%
M-MMAS	6.643,43	M-MMAS	104,52	heMOPSO	281,57	mcMOPSO	100,00%
MOACS	6.700,11	PACO	24,48	PACO	212,31	PACO	100,00%

- Los MOPSOs presentan mejores evaluaciones en lo que refiere a calidad, distribución y error.

- Los MOACOs presentan mejor evaluación en lo que refiere a la extensión del frente Pareto generado.

- Los MOPSOs generan dos a tres veces más soluciones que los MOACOs.

Resultados experimentales



Conclusiones

- Los MOPSOs y MOACOs presentan resultados complementarios
- Los MOPSOs poseen mejor evaluación que los MOACOs considerando 3 de las 4 métricas utilizadas.
- Los MOPSOs presentan más soluciones que los MOACOs.
- **Resolución de problemas combinatorios a través del PSO.**
- **Resolución de problemas multiobjetivos mediante MOPSOs.**

Trabajos futuros:

- Mejorar la extensión del frente hallado por los MOPSOs.
- Aplicar los MOPSOs a otros problemas combinatorios como VRP, QAP, enrutamiento en redes de computadores, etc.
- Aplicar los MOPSOs y MOACOs en un "*Team Algorithm*"
- Añadir otros MOPSOs al estudio.

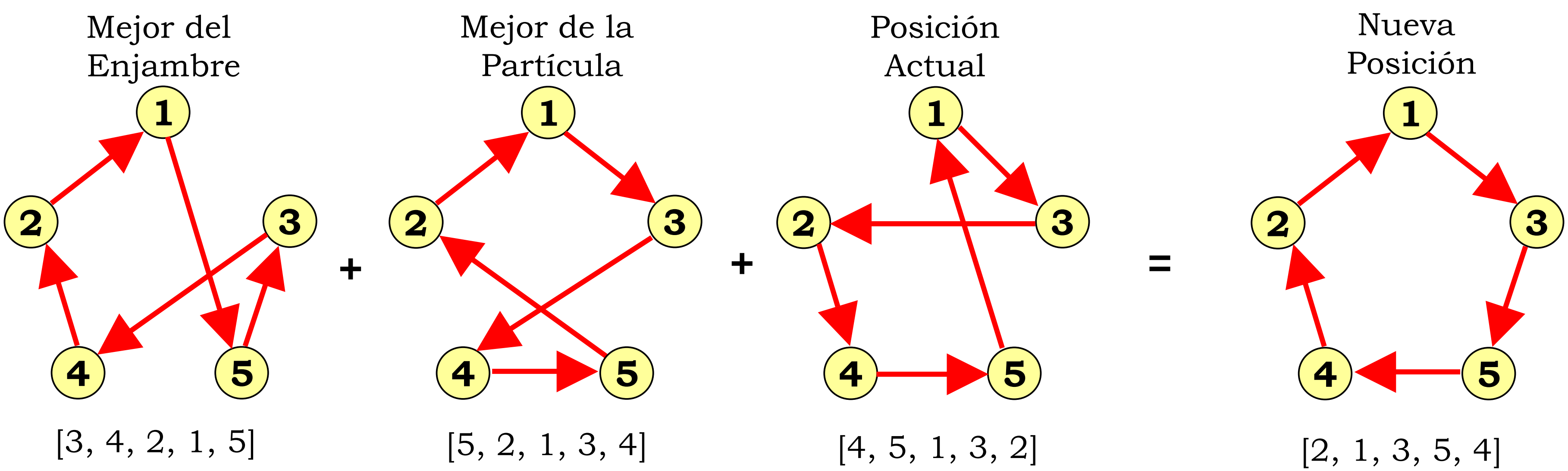
Aplicación de Optimización de Enjambre de Partículas al Problema del Cajero Viajante bi-objetivo

Gracias por su atención

Joaquín Quinto Lima Molinari
joaquinlima@gmail.com

Adaptación de Secrest (2001)

- Se reciben las posiciones y los valores para K1, K2 y K3 → 80, 10, 10
- Se generan N - 2 números aleatorios entre 0 y 100 → 60, 83, 95
- Se elige como vértice inicial al ultimo vértice de la posición actual
- **60** $\leq$ K1 → Selecc. sgte. vértice del mejor del enjambre
- K1 < **83** $\leq$ K1 + K2 → Selecc. sgte. vértice del mejor de la partícula
- K1 + K2 < **95** $\leq$ K1 + K2 + K3 → Selecc. sgte. vértice de la posicion actual



Multiobjective Particles Swarm Optimization – MOPSO

Moore y Chapman (1999)

- MOPSO en el cual cada partícula mantiene una lista de todas las soluciones no dominadas en las que se ha encontrado.

- Cada partícula toma como mejor personal a cualquier elemento de su lista.

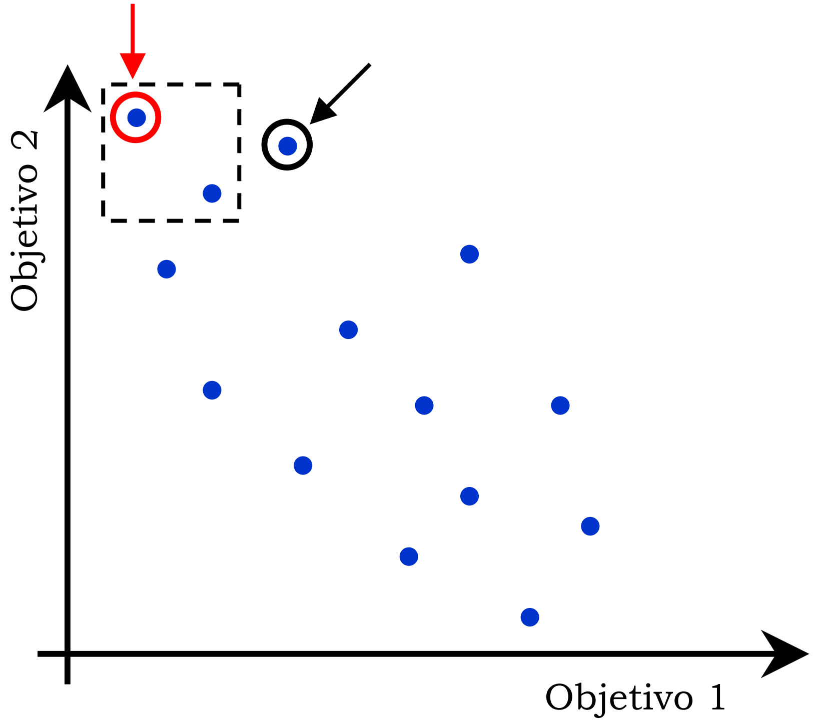
- Cada partícula elige como mejor global a cualquier elemento de su lista no dominado por las soluciones de las listas de las partículas de su grupo.

Multiobjective Particles Swarm Optimization – MOPSO

Hu y Eberhart (2002)

- MOPSO en el cual para cada miembro del enjambre se elige como mejor global a una de las m partículas más cercanas en valor considerando un sólo objetivo fijo.

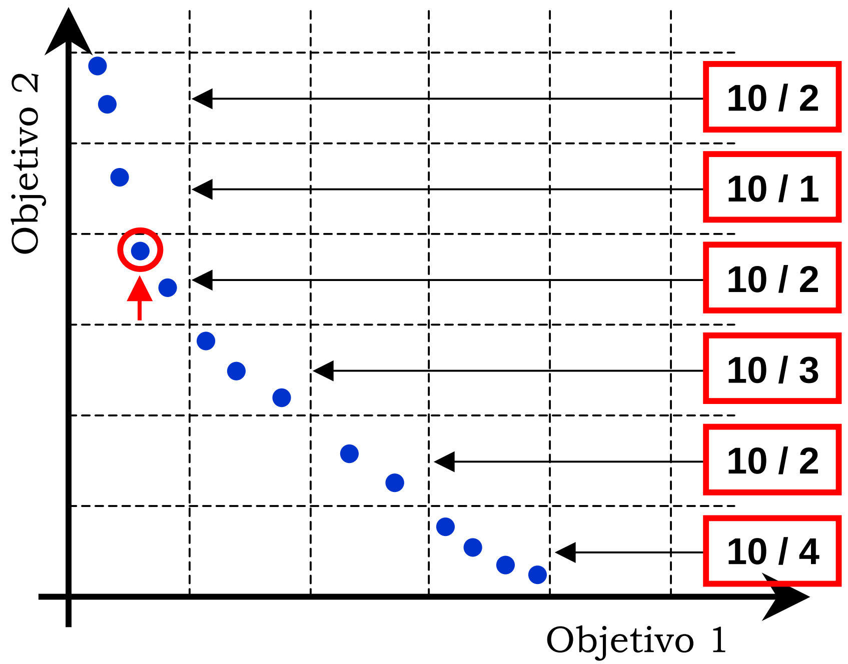
- Cada partícula mantiene como mejor solución personal a la ultima posición no dominada que ha encontrado.



Multiobjective Particles Swarm Optimization – MOPSO

Coello y Lechuga (2002)

- MOPSO que utiliza un repositorio global que representa el espacio objetivo por regiones denominadas *hipercubos.*
- Cada *hipercubo* recibe una calificación igual al resultado de dividir 10 entre el número de soluciones no dominadas almacenadas en él.
- El mejor global se selecciona eligiendo un *hipercubo* mediante el método de la ruleta y luego se elige al azar a cualquier elemento dentro de este.
- Cada partícula mantiene como mejor solución personal a la última solución no dominada que ha encontrado.



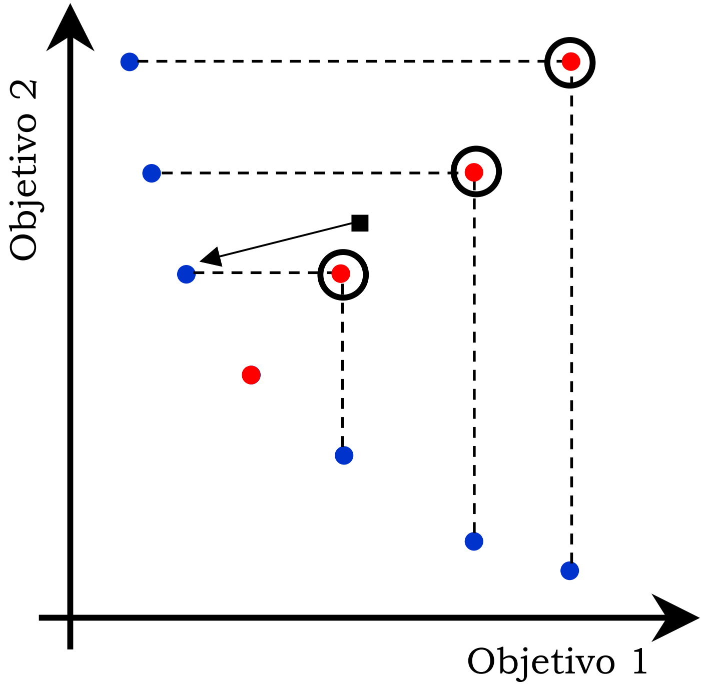
Multiobjective Particles Swarm Optimization – MOPSO

Fieldsend y Singh (2002)

- MOPSO que utiliza una estructura de datos llamada *dominated tree*, que se construye a partir del repositorio global soluciones no dominadas.

- El mejor global para una partícula se elige del repositorio global haciendo uso del *dominated tree.*

- Cada partícula mantiene una lista de las soluciones no dominadas que ha encontrado y elige como mejor personal a cualquier elemento de esta lista



Multiobjective Particles Swarm Optimization – MOPSO

Mostaghim y Teich (2003)

- MOPSO que hace uso de una función llamada *sigma* y un repositorio global de soluciones no dominadas.
- La función *sigma* asocia a cada vector solución un vector que define una línea gradiente hacia el origen de coordenadas en el espacio objetivo.
- Cada partícula elige como mejor global al elemento del repositorio para el cual la distancia euclidiana entre sus vectores σ sea la menor.
- Todas las partículas mantienen como mejor personal a la última posición no dominada que han encontrado.