Computación Bioinspirada

Dr. Edward Hinojosa Cárdenas
ehinojosa@unsa.edu.pe

PSO para problemas de Múltiples Objetivos

- M. Torres y B. Barán, “Optimización de Enjambre de Partículas para Problemas de Muchos Objetivos”, , 2015 XLI Latin American Computing Conference (CLEI), Perú, 2015.

- J. Moore y R. Chapman, “Application of particle swarm to multiobjective optimization” Department of Computer Science and Software Engineering, Auburn University, 1999.

PSO para problemas de Múltiples Objetivos

- En los problema de multiobjetivos (MOPs – Multi-Objective Problems), existen por lo general conflictos entre los diverso objetivos considerados, lo que usualmente impide obtener una única solución óptima y en cambio se encuentra un conjunto de soluciones de compromiso llamadas conjunto Pareto óptimo.

- En este contexto, se dice ue una solución domina a otra si no es peor en ningún objetivo, y es estrictamente mejor en al menos uno de los objetivos

PSO repaso

- La metaheurística de Optimización por Enjambre de partículas (PSO - Particle Swarm Optimization), desarrollada por Kennedy y Eberhart en 1995.

- Se inspira en el comportamiento social de individuos como peces, aves y abejas, que realizan exploración de una zona en grupo, detectando alimentos o guiando al conjunto (cardumen, parvada, enjambre) de manera colectiva a posiciones ventajosas.

PSO repaso

- La metaheurística consiste básicamente en un algoritmo iterativo basado en una población de individuos llamada “enjambre”, en la que cada individuo denominado “partícula”, se dice que sobrevuela el espacio de decisión en busca de soluciones óptimas.

- En un espacio de búsqueda n-dimensional, cada partícula *i* del enjambre conoce su posición actual xi = $[x_{i1}, x_{i2}, \cdots, x_{in}]$, su velocidad $v_i = [v_{i1}, v_{i2}, \cdots, v_{in}]$ (con la cual ha llegado a dicha posición) y la mejor posición $p_i = [p_{i1}, p_{i2}, \cdots, p_{in}]$ en la que ha estado, denominada “mejor posición personal”.

PSO repaso

- Además, todas las partículas conocen la mejor posición de entre todas las mejores posiciones personales en el enjambre $g = [g_1, g_2, \cdots, g_n]$, a la cual se denomina "mejor posición global".

- En cada iteración t del algoritmo, cada componente j de la posición y la velocidad de cada partícula i del enjambre se actualiza conforme a las siguientes ecuaciones:

$$v_{ij}^{t+1} = \omega \times v_{ij}^{t} + C_1 \times rand() \times (p_{ij}^{t} - x_{ij}^{t}) + C_2 \times rand() \times (g_{ij}^{t} - x_{ij}^{t})$$

$$x_{ij}^{t+1} = x_{ij}^{t} + v_{ij}^{t+1}$$

PSO para problemas de Múltiples Objetivos

- Adaptar el algoritmo PSO para realizar una optimización multiobjetivo requiere un nuevo cálculo de velocidad.

- Dicho cálculo debe tener en cuenta que la mejor posición global debe ser necesariamente una solución no dominada de compromiso encontrada por el enjambre.

- De manera similar, la mejor solución local debe ser una solución de compromiso encontrada por la partícula.

PSO para problemas de Múltiples Objetivos

- Moore y Chapman, proponen un método en el cual cada partícula almacena en un repositorio personal todas las soluciones no dominadas que ha encontrado la partícula, así como un repositorio para todas las soluciones no dominadas encontradas por el enjambre.

- Para el cálculo de la nueva velocidad, cada partícula toma como mejor posición personal a cualquier miembro del repositorio de mejores posiciones personales, y como mejor posición global, a cualquier miembro del repositorio de mejores posiciones globales.

PSO para problemas de Múltiples Objetivos

```
input : ω, C1, C2, N
for i = 1 to N do
    Inicializar la i-ésima partícula en una velocidad y
    posición aleatoria
    evaluar(i)
    Actualizar el repositorio global (G)
    Actualizar el repositorio personal de la i-ésima
    partícula (Pi)
while not condicion_de_parada() do
    for i = 1 to N do
        Seleccionar mejor posición personal (elemento
        aleatorio en Pi)
        Seleccionar mejor posición global (elemento
        aleatorio en G)
        Calcular velocidad de la i-ésima partícula
        Calcular la nueva posición de la i-ésima
        partícula
    for i = 1 to N do
        evaluar(i)
        Actualizar el repositorio global (G)
        Actualizar el repositorio personal de la i-ésima
        partícula (Pi)
Retornar soluciones_no_dominadas(G)
```

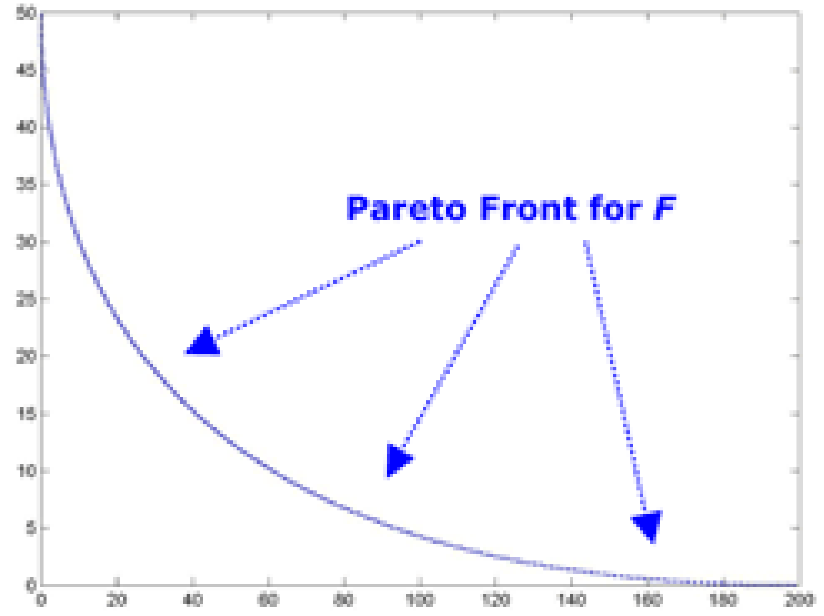
MOPSO Moore y Chapman

Laboratorio – MOPSO

- Aplicar el Algoritmo MOPSO al siguiente problema. Probar diferentes parámetros:

$$
\begin{aligned}
\text{Minimize} \quad F &= (f_1(x,y), f_2(x,y)) \\
f_1(x,y) &= 4x^2 + 4y^2 \\
f_2(x,y) &= (x-5)^2 + (y-5)^2
\end{aligned}
$$

$$
\begin{aligned}
0 \leq \quad x \quad \leq 5 \\
0 \leq \quad y \quad \leq 3
\end{aligned}
$$



Laboratorio – MOPSO - TSP

- Aplicar el Algoritmo MOPSO al siguiente problema TSP. Se puede combinar los métodos vistos en clase o uno que se encuentre en la literatura. Probar diferentes parámetros:

Laboratorio – MOPSO - TSP

- La siguiente tabla muestra el las distancias entre las ciudades:

	A	B	C	D	E	F	G	H	I	J
A	0	12	3	23	1	5	23	56	12	11
B	12	0	9	18	3	41	45	5	41	27
C	3	9	0	89	56	21	12	48	14	29
D	23	18	89	0	87	46	75	17	50	42
E	1	3	56	87	0	55	22	86	14	33
F	5	41	21	46	55	0	21	76	54	81
G	23	45	12	75	22	21	0	11	57	48
H	56	5	48	17	86	76	11	0	63	24
I	12	41	14	50	14	54	57	63	0	9
J	11	27	29	42	33	81	48	24	9	0

Laboratorio – MOPSO - TSP

- La siguiente tabla muestra el costo de viaje entre las ciudades:

	A	B	C	D	E	F	G	H	I	J
A	0	22	47	15	63	21	23	16	11	9
B	22	0	18	62	41	52	13	11	26	43
C	47	18	0	32	57	44	62	20	8	36
D	15	62	32	0	62	45	75	63	14	12
E	63	41	57	62	0	9	99	42	56	23
F	21	52	44	45	9	0	77	58	22	14
G	23	13	62	75	99	77	0	30	25	60
H	16	11	20	63	42	58	30	0	66	85
I	11	26	8	14	56	22	25	66	0	54
J	9	43	36	12	23	14	60	85	54	0

GRACIAS

Dr. Edward Hinojosa Cárdenas
ehinojosa@unsa.edu.pe