I IBERIAN CONFERENCE ON MCDM/A (IMCDM/A)
Universidade de Coimbra, 2025

Enhanced BPSO for Binary Optimization: A Robust Metaheuristic for Complex Problems (RRPS-PAT)

BPSO Mejorado para Optimización Binaria: Una Metaheurística Robusta para Problemas Complejos (RRPS-PAT)

Alfonso Mateos Caballero; Gabriel A. Peña Delfín; Antonio Jiménez-Martín

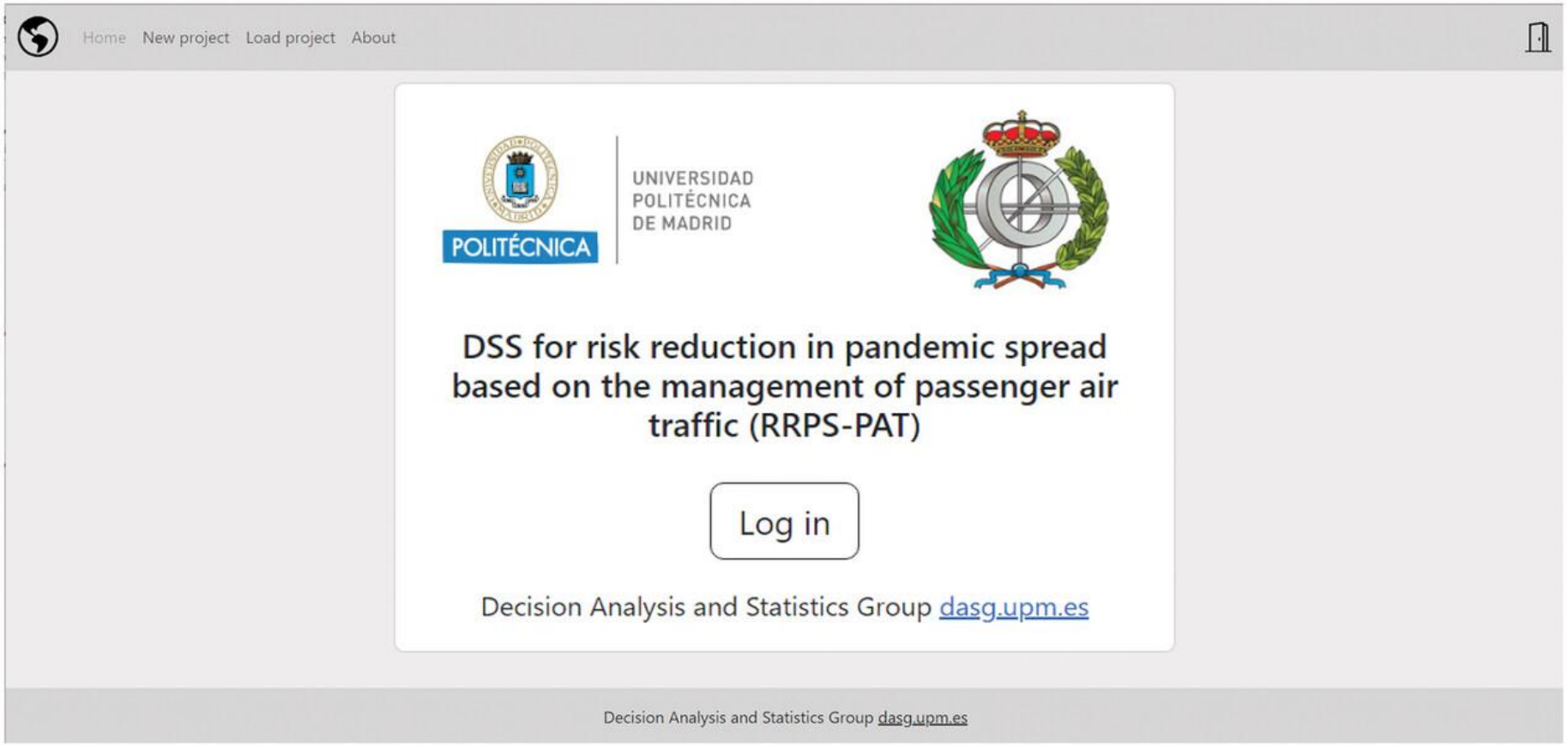
Grupo de Análisis de Decisiones y Estadística

Universidad Politécnica de Madrid

Índice

1. Sistema de Ayuda a la Decisión: RRPS-PAT
2. Particle Swarm Optimization (PSO)
3. Enhanced Binary Particle Swarm Optimization (EBPSO)
4. Comparación
5. Conclusiones

Enhanced BPSO for Binary Optimization: A Robust Metaheuristic for Complex Problems (RRPS-PAT)



Peña, G.A., Mateos, A., Jiménez-Martín, A. and Sanchis, R.G. (2025), A decision support system for risk reduction in pandemic spread based on the management of passenger air traffic. Intl. Trans. in Op. Res., 32: 1893-1917. https://doi.org/10.1111/itor.13576

I IBERIAN CONFERENCE ON MCDM/A (IMCDM/A)
Universidade de Coimbra, 2025

GRUPO ESPAÑOL DE DECISIÓN MULTICRITERIO

UNIVERSIDADE DE COIMBRA

INESCC Instituto de Engenharia de Sistemas e Computadores de Coimbra

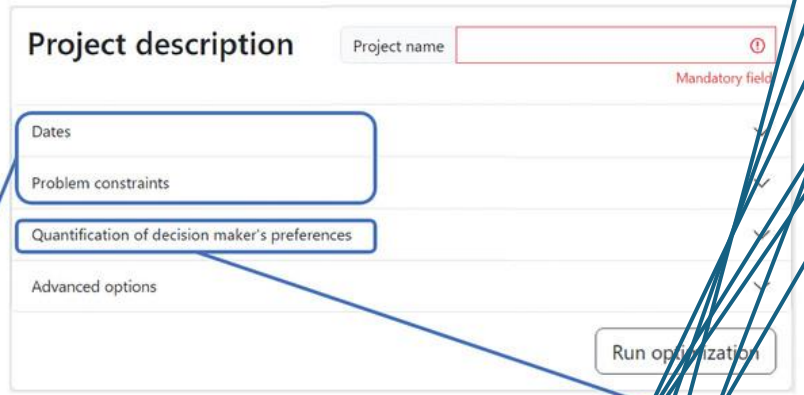
DECISION ANALYSIS AND STATISTICS GROUP

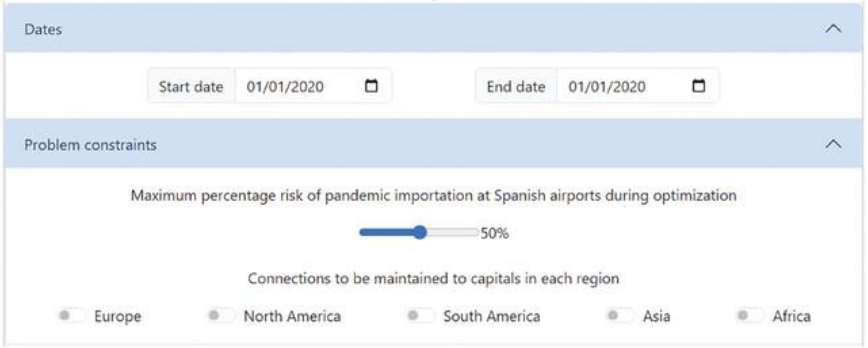
Enhanced BPSO for Binary Optimization: A Robust Metaheuristic for Complex Problems (RRPS-PAT)

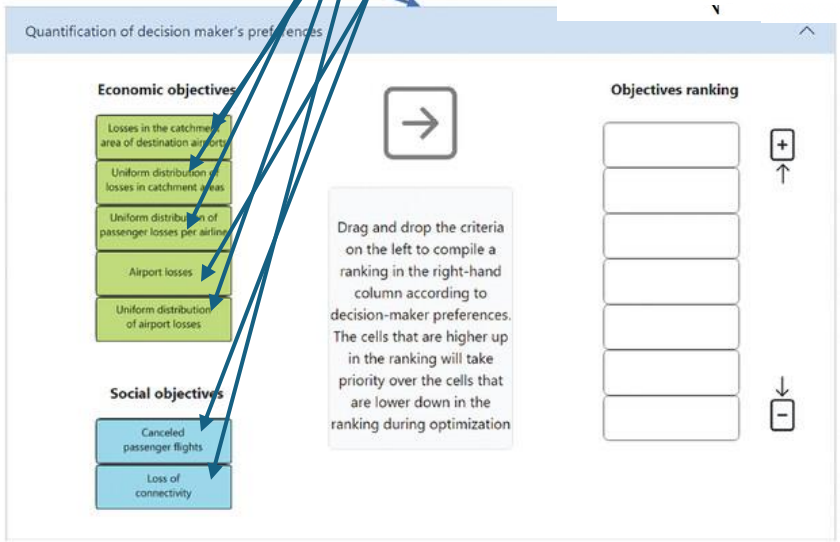
Minimizar la pérdida de conectividad de la

Minimizar la desviación estándar del porcentaje de pérdidas del aeropuerto de destino debido a

$$\min Z_7 = \sqrt{\sum_{i=1}^{I}\left(\frac{\sum_{d=1}^{D}\sum_{j=1}^{J} c_i\left(1-x_{ij}^{d}\right)}{\sum_{i=1}^{I} c_i} - \frac{\sum_{i=1}^{I}\frac{\sum_{d=1}^{D}\sum_{j=1}^{J} c_i\left(1-x_{ij}^{d}\right)}{\sum_{i=1}^{I} c_i}}{\sum_{d=1}^{D}\sum_{i=1}^{I} e_i^d}\right)^2}$$







Enhanced BPSO for Binary Optimization: A Robust Metaheuristic for Complex Problems (RRPS-PAT)

Dates

Problem constraints

Quantification of decision maker's preferences

Advanced options

Metaheuristic parameters

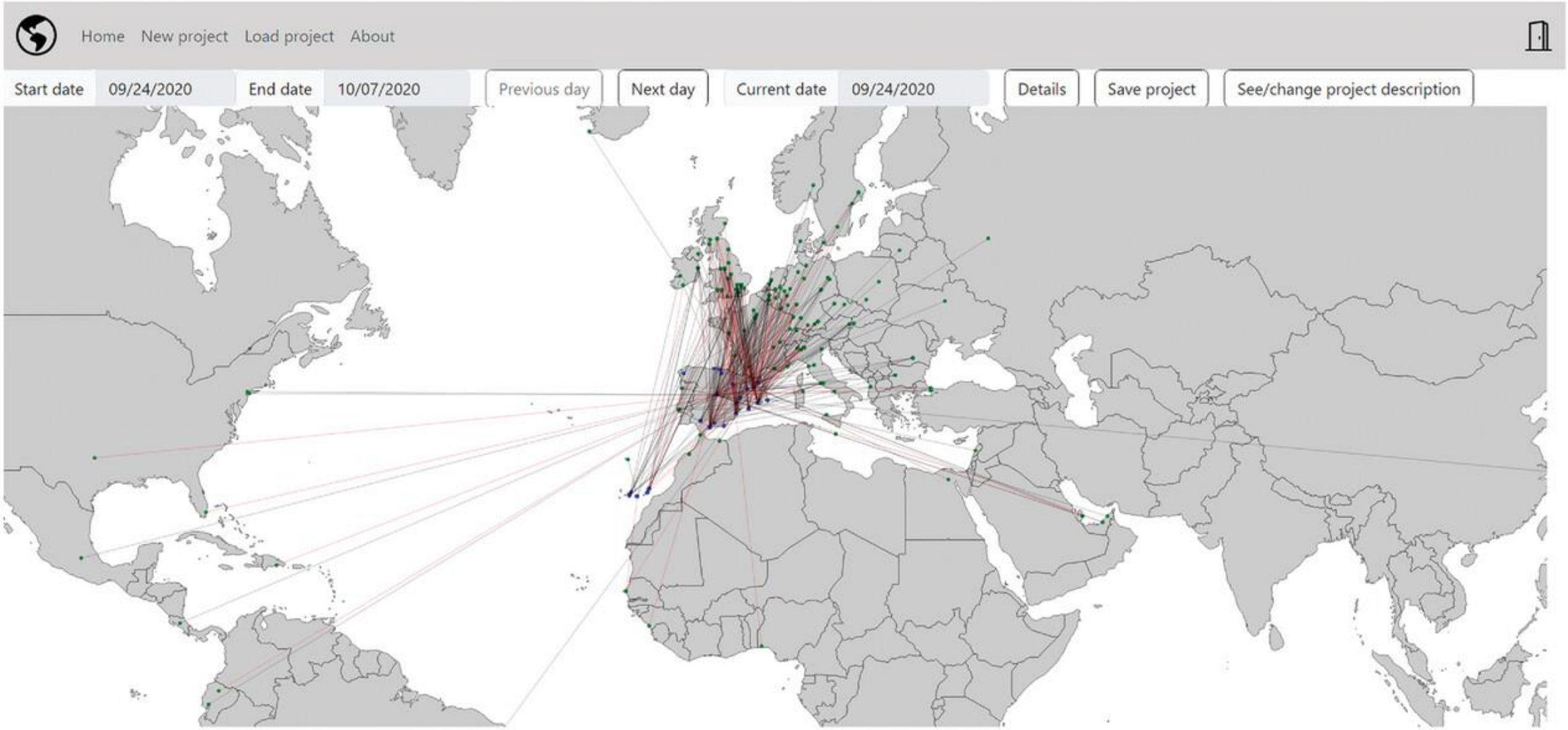
Number of particles (N)	5000	Defines the number of particles to be used in the BPSOM
Inertia weight (W)	0.9	Defines how fast the particles move through the search space
Cognitive component (C1)	1.5	Defines the weight of the personal bests over the global best
Social component (C2)	1.5	Defines the weight of the global best over the personal bests

Save project description

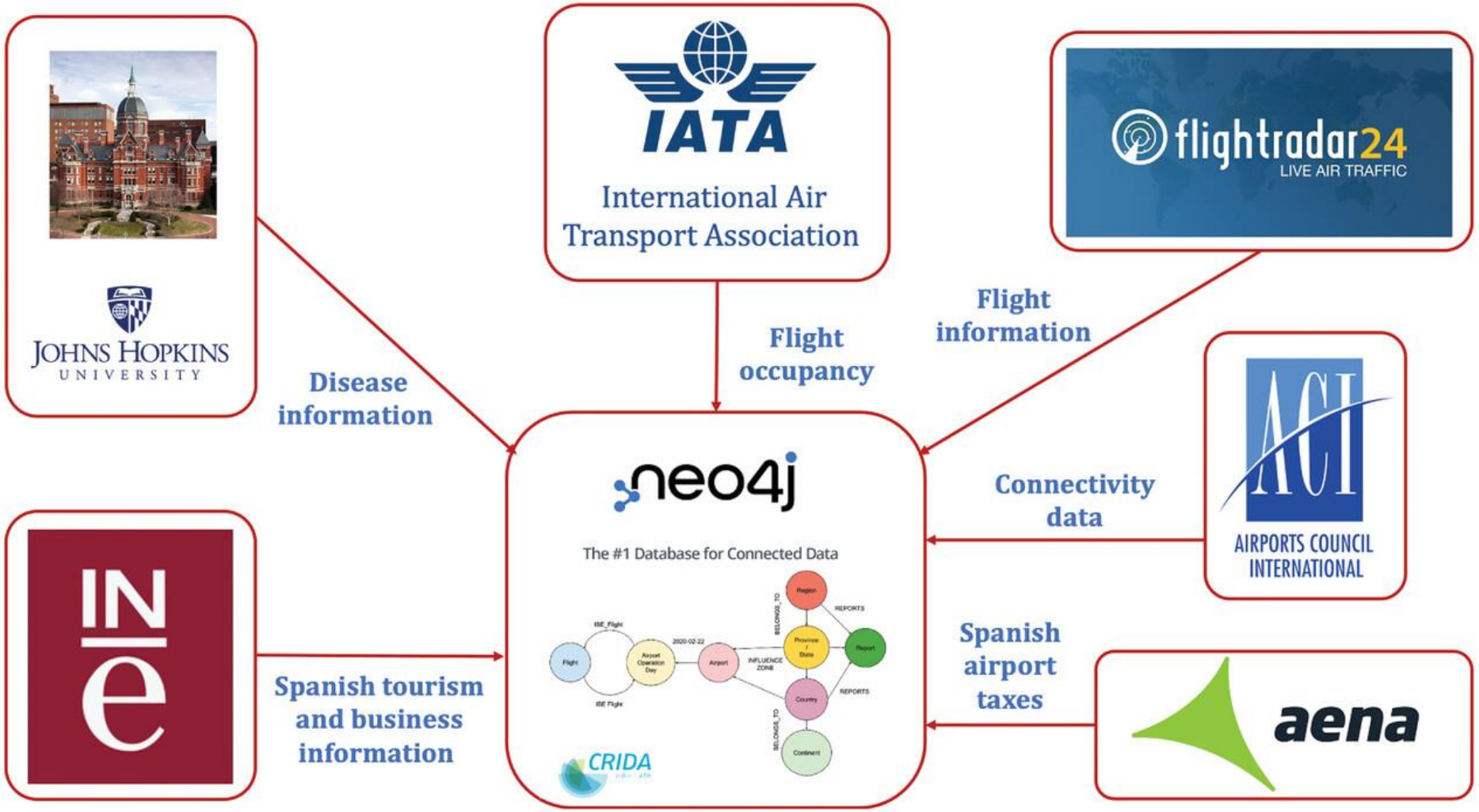
Run optimization

I IBERIAN CONFERENCE ON MCDM/A (IMCDM/A)
Universidade de Coimbra, 2025

Enhanced BPSO for Binary Optimization: A Robust Metaheuristic for Complex Problems (RRPS-PAT)



Enhanced BPSO for Binary Optimization: A Robust Metaheuristic for Complex Problems (RRPS-PAT)



Enhanced BPSO for Binary Optimization: A Robust Metaheuristic for Complex Problems (RRPS-PAT)

Compromise solution information

Selected date 09/30/2020 | Selected airport | Apply filter | Reset filter | See project description

Connections	AAL - AGP	AMS - AGP	AMS - ALC	AMS - BCN	AMS - BIO	AMS - IBZ	AMS - LPA	AMS - MAD	AMS - PMI	AMS - TFS	ARN - AGP	ARN - ALC	ARN - PMI	ATH - MAD	AUH - BCN	BAQ - MAD	BEG - VIT	BGY - BCN
Cancelled connections	0	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	1	0

0 = Closed
1 = Opened

310 connections

Economic objectives

Objectives	Losses in the catchment area of destination airports	Uniform distribution of losses in catchment areas	Uniform distribution of passenger losses per airline	Airport losses	Uniform distribution of airport losses
Optimal values	24.43%	0.139	0.271	24.94%	0.276

Social objectives

Objectives	Canceled passenger flights	Loss of connectivity
Optimal values	24.18%	23.76%

Percentage reduction ranges in catchment areas, airlines and airports

Influence areas	Airlines	Airports
(5.01, 47.81)%	(4.41, 48.02)%	(8.01, 68.11)%

Particle Swarm Optimization (PSO)

1. Inicialización:

1. Crea un enjambre de partículas con posiciones y velocidades aleatorias.
2. Evalúa la función de aptitud de cada partícula.
3. Establece *Pbest* para cada partícula y *Gbest* para el enjambre.

2. Iteración:

1. Para cada partícula:
    1. Actualiza la velocidad usando la ecuación de velocidad.
    2. Actualiza la posición usando la ecuación de posición.
    3. Evalúa la nueva posición con la función de aptitud.
    4. Actualiza *Pbest* si la nueva posición es mejor.
    5. Actualiza *Gbest* si alguna partícula encuentra una solución mejor que la actual.
2. Repite hasta cumplir un criterio de parada (número máximo de iteraciones o convergencia).

3. Resultado:

1. Devuelve *Gbest* como la solución óptima.

Particle Swarm Optimization (PSO)

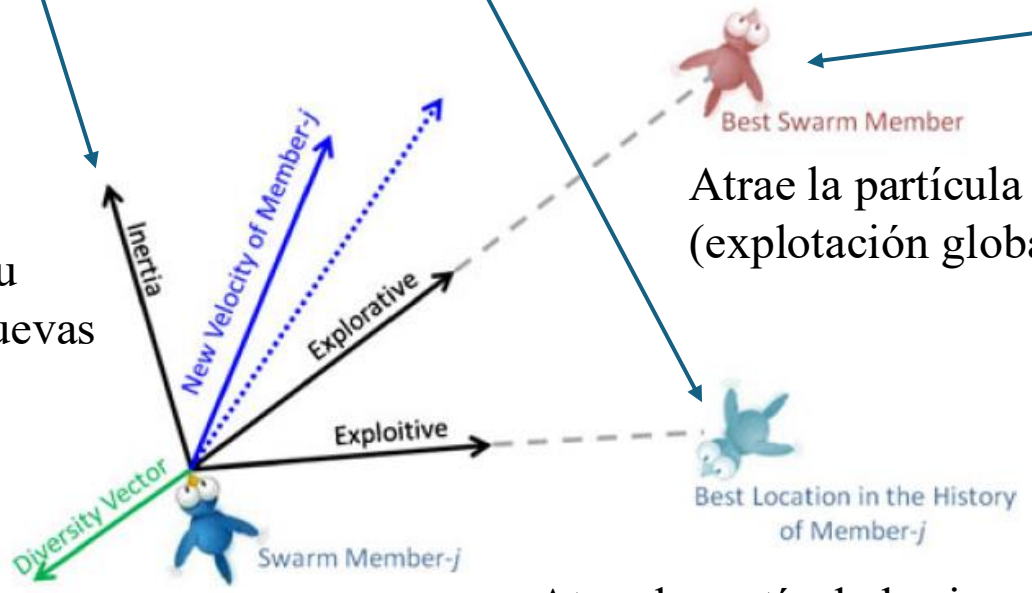
Solución en el espacio de búsqueda por la partícula *i*

$$x_{id}(t+1) = x_{id}(t) + V_{id}(t+1)$$

Inercia — Componente cognitiva — Componente social

$$V_{id}(t+1) = V_{id}(t)\cdot w + C_1 \times R_1 \times (Pbest_{id}(t) - x_{id}) + C_2 \times R_2 \times (Gbest_d(t) - x_{id})$$

Permite a las partículas mantener su dirección actual y explorar áreas nuevas



Atrae la partícula hacia el Gbest (explotación global)

Atrae la partícula hacia su *Pbest* (explotación local)

Binary particle swarm optimization (BPSO)

J. Kennedy, R. C. Eberhart, A discrete binary version of the particle swarm algorithm, in: Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics. Computational Cybernetics and Simulation, Vol. 5, IEEE, 1997, pp. 4104–4108.

La probabilidad de que un bit en la dimensión *d* de la partícula *i* cambie a 0 o 1, respectivamente

$$d^0_{id,1}(t+1) = \begin{cases} -C_1 \times R_1, & \text{if } Pbest_{id}(t) = 1 \\ C_1 \times R_1, & \text{if } Pbest_{id}(t) = 0 \end{cases} \qquad d^0_{id,2}(t+1) = \begin{cases} -C_2 \times R_2, & \text{if } Gbest_d(t) = 1 \\ C_2 \times R_2, & \text{if } Gbest_d(t) = 0 \end{cases}$$

$$V^0_{id}(t+1) = w \times V^0_{id}(t) + d^0_{id,1}(t+1) + d^0_{id,2}(t+1)$$

$$V^1_{id}(t+1) = w \times V^1_{id}(t) + d^1_{id,1}(t+1) + d^1_{id,2}(t+1)$$

$$d^1_{id,1}(t+1) = \begin{cases} C_1 \times R_1, & \text{if } Pbest_{id}(t) = 1 \\ -C_1 \times R_1, & \text{if } Pbest_{id}(t) = 0 \end{cases} \qquad d^1_{id,2}(t+1) = \begin{cases} C_2 \times R_2, & \text{if } Gbest_d(t) = 1 \\ -C_2 \times R_2, & \text{if } Gbest_d(t) = 0 \end{cases}$$

$$V_{id}^{0}(t+1) = w \times V_{id}^{0}(t) + d_{id,1}^{0}(t+1) + d_{id,2}^{0}(t+1)$$

$$V_{id}^{1}(t+1) = w \times V_{id}^{1}(t) + d_{id,1}^{1}(t+1) + d_{id,2}^{1}(t+1)$$

$$V_{id}^{c}(t+1) = \begin{cases} V_{id}^{0}(t+1), & \text{if } x_{id}(t) = 1 \\ V_{id}^{1}(t+1), & \text{if } x_{id}(t) = 0 \end{cases}$$

$$x_{id}(t+1) = \begin{cases} \bar{x}_{id}(t), & \text{if } rand \leq \mathrm{T}(\mathrm{x}_{id}(t)) \\ x_{id}(t), & \text{if } rand > \mathrm{T}(\mathrm{x}_{id}(t)) \end{cases}$$

Parámetros dinámicos

Peso de inercia variable en el tiempo

Y. Shi, R. C. Eberhart, Empirical study of particle swarm optimization, in: Proceedings of the 1999 Congress on Evolutionary Computation- CEC99 (Cat. No. 99TH8406), Vol. 3, IEEE, 1999, pp. 1945–1950.

$$w = (w_{max} - w_{min})\frac{T-t}{T} + w_{min};$$

Coeficientes de aceleración

A. Ratnaweera, S. K. Halgamuge, H. C. Watson, Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients, IEEE Transactions on Evolutionary Computation 8 (3) (2004) 240–255.

$$C_1 = (C_{1fin} - C_{1ini})\frac{t}{T} + C_{1ini};$$

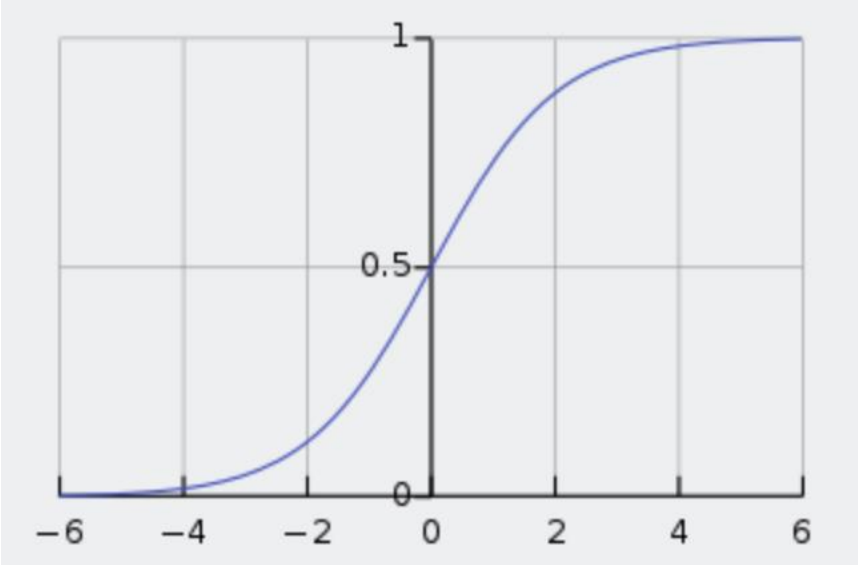
$$C_2 = (C_{2fin} - C_{2ini})\frac{t}{T} + C_{2ini};$$

Enhanced Binary Particle Swarm Optimization (EBPSO)

$$T(x_{id}(t)) = \frac{1}{1 + e^{-\alpha \times V_{id}(t)}}$$

H es el número de partículas cuya distancia de Hamming respecto a *Gbest* es menor que *hd*.

$$\alpha = \alpha_{min} + (\alpha_{max} - \alpha_{min}) \times (1 - \frac{H}{N})$$



- Si $H = N$ entonces la función de transferencia presenta su gradiente mínimo
- Si $H = 0$ entonces el gradiente alcanza su magnitud máxima. *hd* es un parámetro introducido por el usuario que especifica cuán sensible es la función de transferencia a la distancia de las partículas con respecto a Gbest.
- Un valor alto de α hace que las partículas se desplacen hacia su principal atractor, ya que la función de transferencia incrementa la probabilidad de que ocurra $rand \leq T(x_{id}(t))$ frente a $rand > T(x_{id}(t))$

Método de reparación/mejora

Algorithm 2 Modified repair/improvement method

```
for dimension d ∈ [1 ... D] do
    R_d ← f(x)_d / c(x)_d
end for
for particle i ∈ [1 ... N] do
    diff ← MaxCons − c(x_i(t))
    if diff < 0 then
        while diff < 0 do
            if rand ≥ SRate then  a ← Dimension with lowest R_d
            else  a ← Random dimension
            end if
            x_ia ← 0
            diff ← MaxCons − c(x_i(t))
        end while
    end if
    if diff > 0 then
        endloop ← false
        while endloop == false do
            for dimension d ∈ [1 ... D] do
                if MaxCons − c(x_i(t)_{d=1}) ≥ 0 then Candidates_d ← d
                end if
            end for
            if length(Candidates) > 0 then
                if rand ≥ SRate then
                    b ← Dimension with largest R_d in Candidates
                else
                    b ← Random dimension in Candidates
                end if
                x_ib ← 1
                diff ← MaxCons − c(x_i(t))
            else  endloop = true
            end if
        end while
    end if
    EvaluateFitness(x_i(t))
end for
```

El paso inicial consiste en calcular el cociente *Rd*, que es el resultado de dividir el valor de la función objetivo entre los valores de las restricciones para cada elemento dimensional

La metodología de reparación implica la eliminación sistemática de elementos del conjunto solución hasta que se alcanza la factibilidad.

La metodología de mejora consiste en la adición incremental de elementos a la solución desde una lista precomputada de candidatos

Configuración de parámetros para EBPSO

- 10 partículas;
- valores inicial y final de C_1 y C_2 se establecieron en 1.0 y 0.0, respectivamente;
- *wmax* es 0.9 y *wmin* es 0.0;
- α varía entre 1.0 y 5.0;
- $hd = 1$, y
- la probabilidad *SRate* es 0.5

Enhanced BPSO for Binary Optimization: A Robust Metaheuristic for Complex Problems (RRPS-PAT)

Table 7: Best achieved fitness values and execution times for BPSO and EBPSO

		fitness (BPSO)	fitness (EBPSO)	time (BPSO)	time (EBPSO)
Scenario 1	Mean	0.233	**0.220**	5h 21min	**40min**
	Std. Dev.	6.17E-03	**7.97E-4**	**4.07E-05**	5.18E-5
Scenario 1a	Mean	0.242	**0.222**	5h 22min	**39min**
	Std. Dev.	9.80E-03	**4.70E-4**	8.55E-05	**6.96E-5**
Scenario 2	Mean	0.245	**0.224**	5h 24min	**39min**
	Std. Dev.	6.44E-03	**5.36E-4**	8.64E-05	**7.58E-5**
Scenario 3	Mean	0.256	**0.230**	5h 15min	**38min**
	Std. Dev.	9.70E-03	**2.82E-4**	6.84E-05	**3.07E-5**
Scenario 4	Mean	0.258	**0.234**	5h 21min	**38min**
	Std. Dev.	3.39E-03	**8.45E-4**	6.37E-05	**3.03E-5**
Scenario 5	Mean	0.374	**0.366**	5h 27min	**1h 2min**
	Std. Dev.	9.22E-03	**6.24E-4**	**5.14E-05**	7.03E-5
Scenario 5a	Mean	0.390	**0.385**	5h 24min	**1h 5min**
	Std. Dev.	8.98E-03	**8.83E-4**	**3.95E-05**	6.57E-5
Scenario 6	Mean	0.378	**0.364**	5h 21min	**1h 3min**
	Std. Dev.	9.70E-03	**3.94E-4**	**4.57E-05**	4.85E-5
Scenario 7	Mean	0.369	**0.361**	5h 20min	**1h 7min**
	Std. Dev.	4.46E-03	**1.32E-4**	**2.96E-05**	5.64E-5
Scenario 8	Mean	0.354	**0.342**	5h 16min	**59min**
	Std. Dev.	6.59E-03	**3.36E-4**	5.29E-05	**2.41E-5**
Scenario 9	Mean	0.546	**0.533**	5h 24min	**1h 20min**
	Std. Dev.	8.75E-03	**7.77E-4**	**2.81E-05**	6.30E-5
Scenario 9a	Mean	0.574	**0.564**	5h 24min	**1h 21min**
	Std. Dev.	8.93E-03	**6.62E-4**	**5.77E-05**	6.03E-5
Scenario 10	Mean	0.532	**0.524**	5h 28min	**1h 17min**
	Std. Dev.	7.57E-03	**2.24E-4**	5.43E-05	**4.90E-05**
Scenario 11	Mean	0.530	**0.520**	5h 15min	**1h 19min**
	Std. Dev.	6.14E-03	**3.02E-4**	**2.28E-05**	2.93E-5
Scenario 12	Mean	0.491	**0.479**	5h 23min	**1h 21min**
	Std. Dev.	3.73E-03	**7.80E-4**	**2.97E-05**	3.56E-5

Conclusiones

1. Se introduce una versión mejorada del BPSO (EBPSO), que equilibra la exploración y la explotación mediante **parámetros variables en el tiempo, nueva función de transferencia** y un **método de reparación/mejora**.

2. La **función de transferencia** permite que las partículas se dispersen o sigan a su principal atractor, variando según la fase de la optimización, mientras que **los parámetros variables en el tiempo** se encargan de modificar gradualmente el atractor de cada partícula, haciendo que pasen de un comportamiento más individual a uno más social, equilibrando de este modo la exploración y la explotación. Además, se ha incorporado un **método de reparación/mejora** para tratar soluciones infactibles de manera robusta.

3. **EBPSO supera claramente a BPSO** en la reducción del riesgo de importación pandémica mediante la gestión del tráfico aéreo de pasajeros. En los 12 escenarios analizados, EBPSO supera a BPSO tanto en la calidad de la solución óptima obtenida como en el tiempo necesario para alcanzarla, reduciendo los tiempos de más **de 5 horas a aproximadamente una hora**.

4. EBPSO ha sido evaluado en cinco **análisis comparativos frente a diversas metaheurísticas binarias** de última generación, para el problema de la mochila (*knapsack)*. Los resultados muestran que EBPSO las supera o, en el peor de los casos, iguala el rendimiento de los métodos más eficaces en prácticamente todos los casos.

Muchas Gracias