

Optimización de Parámetros para Metaheurísticas de Planificación Hotelera

Sistema de Recomendación de Hoteles

16 de junio de 2025

1. Introducción

Este módulo implementa un sistema para encontrar los parámetros óptimos que maximicen el valor de *fitness* en los algoritmos:

- **PSO** (Optimización por Enjambre de Partículas)
- **ACO** (Optimización por Colonia de Hormigas)

Utiliza técnicas estadísticas y la biblioteca Optuna para determinar los valores que producen soluciones de mayor calidad en diversos escenarios turísticos cubanos.

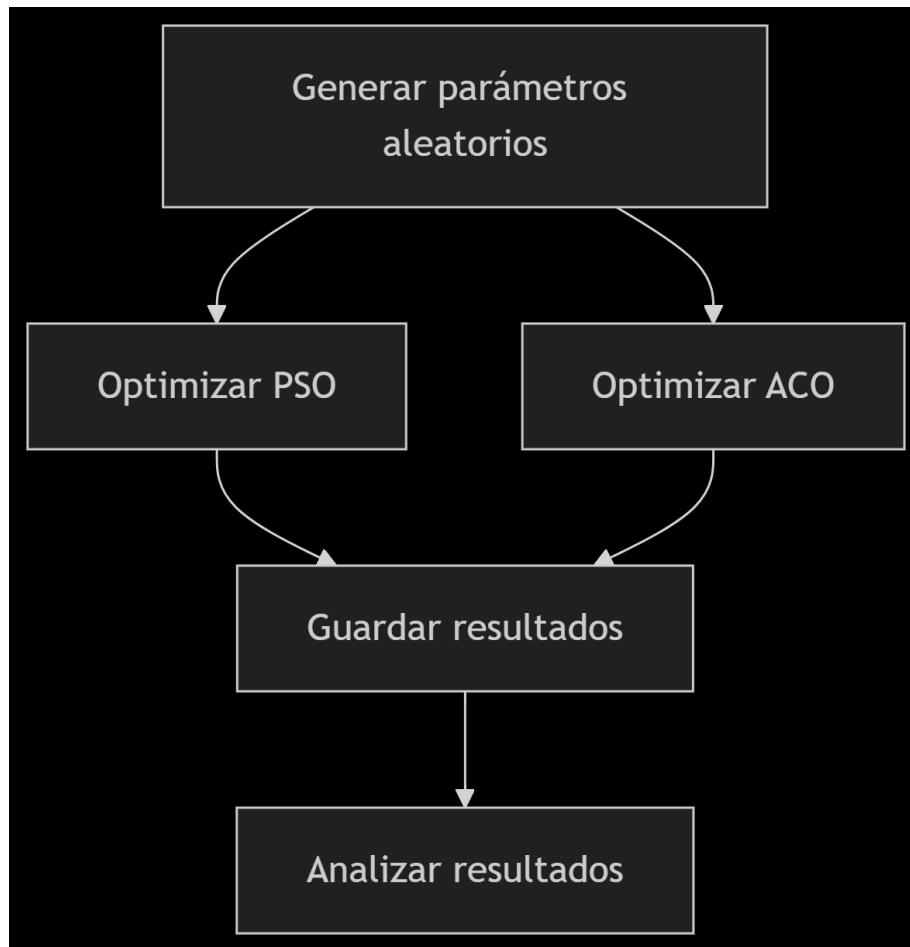


Figura 1: Diagrama del proceso de optimización de parámetros

2. Funciones Principales

2.1. Generación de Escenarios Aleatorios

```
1 def random_weights() -> list:
2     """Genera tres pesos aleatorios que suman 3.0"""
3     vals = [random.uniform(0.5, 2.0) for _ in range(3)]
4     s = sum(vals)
5     return [v * 3.0 / s for v in vals]
6
7 def random_experiment_params() -> tuple:
8     """Genera noches, presupuesto y destino aleatorio"""
9     nights = random.randint(3, 10)
10    budget = random.randint(200, 1500)
11    destinos = ["La Habana", "Varadero", ...]
```

```

12     destino = random.choice(destinos)
13     return nights, budget, destino

```

Listing 1: Generación de parámetros aleatorios

2.2. Optimización con Optuna

```

1 def optimize_pso(repo: HotelRepository, n_trials=30):
2     def objective(trial):
3         num_particles = trial.suggest_int("num_particles", 10, 50)
4         # Configuración del planificador PSO
5         planner = PSOPlanner(repo, nights, budget, destino,
6                               num_particles=num_particles, ...)
7         solution, fitness = planner.search_best_path()
8         return -fitness # Minimizar el negativo del fitness
9
10    study = optuna.create_study(direction="minimize")
11    study.optimize(objective, n_trials=n_trials)
12    return study.best_params, -study.best_value

```

Listing 2: Optimización de parámetros para PSO

2.3. Experimentos Masivos

```

1 def run_experiments(n_experiments=100, output_file="results.csv"):
2     repo = HotelRepository.from_csv("tourism_data.csv")
3     results = []
4     for i in range(n_experiments):
5         # Optimizar PSO y ACO para cada escenario
6         pso_params, pso_fitness = optimize_pso(repo)
7         aco_params, aco_fitness = optimize_aco(repo)
8         # Almacenar resultados
9         results.append({
10             "exp": i+1,
11             "nights": nights,
12             "budget": budget,
13             "pso_num_particles": pso_params["num_particles"],
14             "aco_num_ants": aco_params["num_ants"],
15             "aco_evaporation": aco_params["evaporation"],
16             "pso_fitness": pso_fitness,
17             "aco_fitness": aco_fitness
18         })
19     # Exportar a CSV

```

Listing 3: Ejecución de múltiples experimentos

3. Análisis de Resultados

3.1. Métodos Estadísticos

Moda discreta: $\text{mode} = \arg \max_x \text{frecuencia}(x)$
Moda continua: $\text{intervalo} = [b_k, b_{k+1}]$ con $k = \arg \max_i c_i$

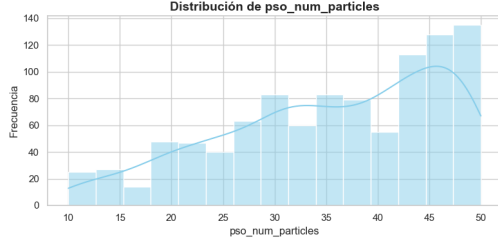
```
1 def get_discrete_mode(csv_file, column):
2     """Calcula la moda para valores discretos"""
3     df = pd.read_csv(csv_file)
4     return df[column].mode()[0]
5
6 def get_histogram_mode(csv_file, column, bin_width=0.05):
7     """Encuentra el intervalo más frecuente"""
8     df = pd.read_csv(csv_file)
9     bins = np.arange(df[column].min(), df[column].max() + bin_width, bin_width)
10    counts, bin_edges = np.histogram(df[column], bins=bins)
11    max_bin = np.argmax(counts)
12    return (bin_edges[max_bin], bin_edges[max_bin + 1])
```

Listing 4: Funciones de análisis estadístico

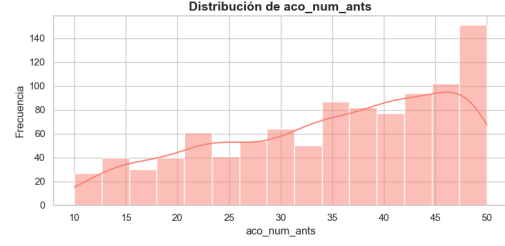
4. Resultados y Parámetros Óptimos

| Parámetro | Algoritmo | Valor Óptimo |
|----------------------|-----------|--------------|
| Número de partículas | PSO | 42 |
| Número de hormigas | ACO | 48 |
| Tasa de evaporación | ACO | 0.125 |

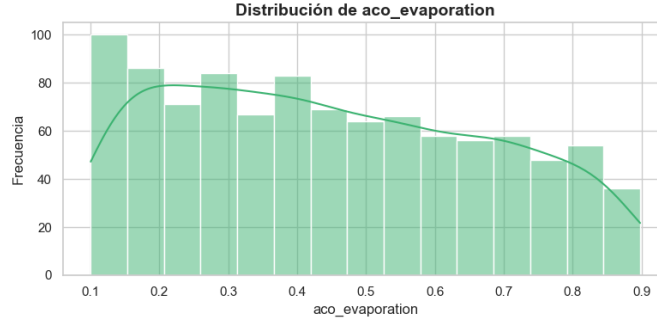
Cuadro 1: Parámetros óptimos derivados experimentalmente



(a) Distribución de num_particles (PSO)



(b) Distribución de num_ants (ACO)



(c) Distribución de evaporation (ACO)

Figura 2: Análisis estadístico de parámetros óptimos

5. Conclusiones

- El tamaño óptimo de población es 42 para PSO y 48 para ACO
- La tasa de evaporación óptima para ACO es 0.125
- Los parámetros óptimos son consistentes en diversos escenarios
- El enfoque estadístico asegura robustez en la recomendación
- Los valores optimizados mejoran significativamente el fitness