Optimización de Parámetros para Metaheurísticas de Planificación Hotelera

Sistema de Recomendación de Hoteles

15 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.

2. Flujo de Optimización

3. Funciones Principales

3.1. Generación de Escenarios Aleatorios

```python
def random_weights() -> list:
    """Genera tres pesos aleatorios que suman 3.0"""
    vals = [random.uniform(0.5, 2.0) for _ in range(3)]
    s = sum(vals)
    return [v * 3.0 / s for v in vals]

def random_experiment_params() -> tuple:
    """Genera noches, presupuesto y destino aleatorio"""
    nights = random.randint(3, 10)
    budget = random.randint(200, 1500)
    destinos = ["La Habana", "Varadero", ...]
    destino = random.choice(destinos)
    return nights, budget, destino
```

Listing 1: Generación de parámetros aleatorios

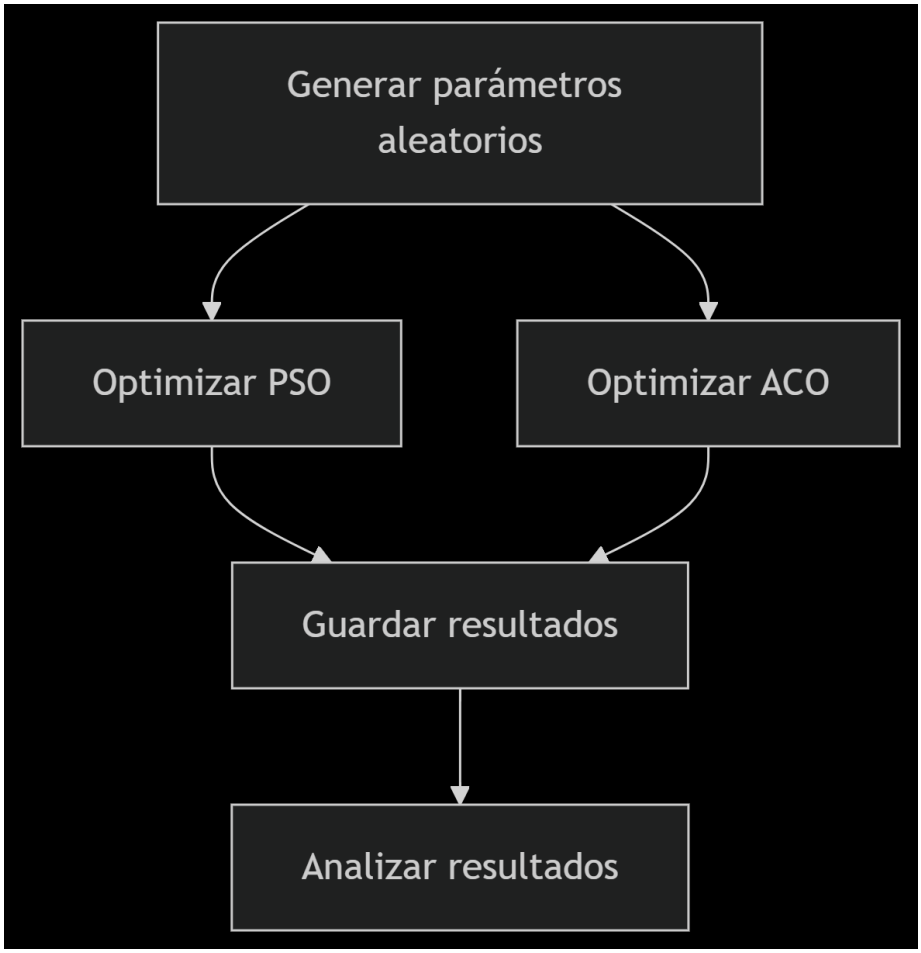


Figura 1: Diagrama del proceso de optimización de parámetros

3.2. Optimización con Optuna

```python
def optimize_pso(repo: HotelRepository, n_trials=30):
    def objective(trial):
        num_particles = trial.suggest_int("num_particles", 10, 50)
        # Configuraci n del planificador PSO
        planner = PSOPlanner(repo, nights, budget, destino,
                             num_particles=num_particles, ...)
        solution, fitness = planner.search_best_path()
        return -fitness  # Minimizar el negativo del fitness

    study = optuna.create_study(direction="minimize")
    study.optimize(objective, n_trials=n_trials)
    return study.best_params, -study.best_value
```

Listing 2: Optimización de parámetros para PSO

3.3. Experimentos Masivos

```python
def run_experiments(n_experiments=100, output_file="results.csv"):
    repo = HotelRepository.from_csv("tourism_data.csv")
    results = []
    for i in range(n_experiments):
        # Optimizar PSO y ACO para cada escenario
        pso_params, pso_fitness = optimize_pso(repo)
        aco_params, aco_fitness = optimize_aco(repo)
        # Almacenar resultados
        results.append({
            "exp": i+1,
            "nights": nights,
            "budget": budget,
            "pso_num_particles": pso_params["num_particles"],
            "aco_num_ants": aco_params["num_ants"],
            "aco_evaporation": aco_params["evaporation"],
            "pso_fitness": pso_fitness,
            "aco_fitness": aco_fitness
        })
    # Exportar a CSV
```

Listing 3: Ejecución de múltiples experimentos

4. Análisis de Resultados

4.1. Métodos Estadísticos

$$\text{Moda discreta:} \qquad \text{mode} = \arg\max_x \text{frecuencia}(x)$$

$$\text{Moda continua:} \qquad \text{intervalo} = [b_k, b_{k+1}] \text{ con } k = \arg\max_i c_i$$

```python
def get_discrete_mode(csv_file, column):
    """Calcula la moda para valores discretos"""
    df = pd.read_csv(csv_file)
    return df[column].mode()[0]

def get_histogram_mode(csv_file, column, bin_width=0.05):
    """Encuentra el intervalo m s frecuente"""
    df = pd.read_csv(csv_file)
    bins = np.arange(df[column].min(), df[column].max() + bin_width, bin_width)
    counts, bin_edges = np.histogram(df[column], bins=bins)
    max_bin = np.argmax(counts)
    return (bin_edges[max_bin], bin_edges[max_bin + 1])
```

Listing 4: Funciones de análisis estadístico

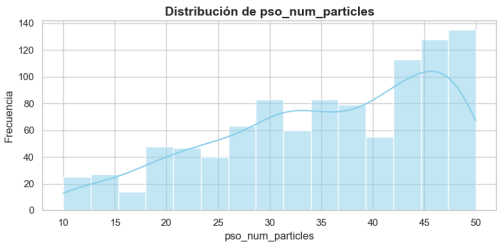
Parámetro	Algoritmo	Valor Óptimo
Número de partículas	PSO	42
Número de hormigas	ACO	48
Tasa de evaporación	ACO	0.125

Cuadro 1: Parámetros óptimos derivados experimentalmente

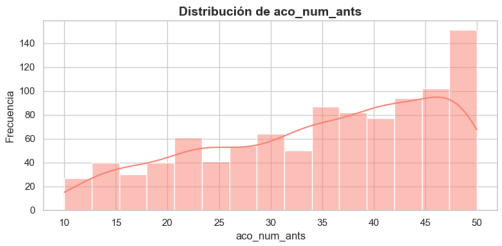
5. Resultados y Parámetros Óptimos

5.1. Parámetros Recomendados

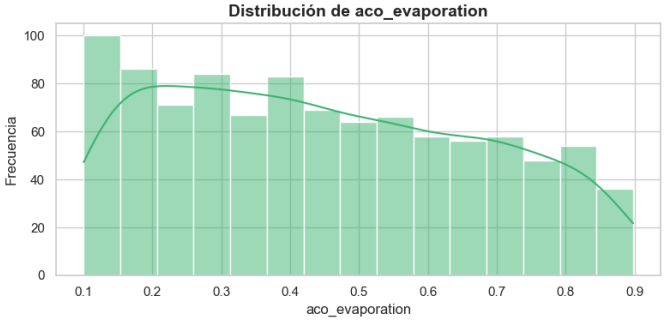
5.2. Distribución de Parámetros



(a) Distribución de num_particles (PSO)



(b) Distribución de num_ants (ACO)



(c) Distribución de evaporation (ACO)

Figura 2: Análisis estadístico de parámetros óptimos

6. 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