F1 Sustainable Logistics:

Optimización de Calendario mediante Algoritmos Genéticos

Autor: Jaime Arias

Rol: Supply Chain Analyst & Python Developer

Tecnologías: Python, Pandas, Folium, Plotly, Genetic Algorithms.

1. Definición del Problema (Business Case)

El Reto

La Fórmula 1 es un "circo logístico" global que mueve miles de toneladas de equipamiento y personal a través de 24 sedes en 5 continentes en solo 9 meses. El problema central es la ineficiencia de las rutas heredadas, que a menudo implican saltos transatlánticos redundantes ("rutas espagueti"), aumentando drásticamente los costos operativos y la huella de carbono.

El Objetivo

Desarrollar un modelo matemático capaz de proponer un calendario alternativo que:

1. **Minimice la distancia total de viaje** (ahorro de combustible y costos).
2. **Respete restricciones climáticas estrictas** (evitar monzones en Asia o nieve en Canadá).
3. **Cumpla con reglas comerciales innegociables** (Inicio en Australia, Fin en Abu Dhabi).

Este no es un simple problema de "Viajante de Comercio" (TSP), es un **TSP con Ventanas de Tiempo (TSPTW) y Restricciones de Anclaje**, lo que eleva exponencialmente su complejidad.

2. Justificación Algorítmica: ¿Por qué Algoritmos Genéticos?

Complejidad Computacional

El problema implica ordenar 22 carreras (excluyendo inicio/fin fijos). El número de combinaciones posibles es factorial: $22! \approx 1.12 \times 10^{21}$.

- **Fuerza Bruta:** Tardaría millones de años en computar todas las opciones.
- **Programación Lineal (MIP):** Posible, pero rígida para manejar reglas de negocio no lineales (como las "Zonas de Riesgo").

La Solución Evolutiva

Elegimos un **Algoritmo Genético (GA)** porque es una heurística de búsqueda estocástica ideal para espacios de solución gigantescos y rugosos.

- **Mecanismo:** Imita la selección natural. Las "rutas" son individuos. Las mejores rutas sobreviven y se cruzan; las peores mueren.
- **Ventaja:** No garantiza el óptimo matemático absoluto (que es incalculable en tiempo razonable), pero garantiza una solución **"Suficientemente Buena" (Good Enough)** y ejecutable en minutos.

3. Anatomía del Código (Code Walkthrough)

El sistema sigue una arquitectura modular Orientada a Objetos (OOP) para garantizar escalabilidad.

A. Gestión de Datos (class F1LogisticsData)

- **Ubicación:** Celda 2.
- **Función:** Carga los CSVs (f1_circuits.csv, race_constraints.csv) y pre-calcula la **Matriz de Distancias Geodésicas**.
- **Lógica Clave:**
    - Método get_fixed_start_end_indices(): Identifica dinámicamente qué ciudades son las "anclas" (Australia/Abu Dhabi) leyendo el archivo de configuración, desacoplando la regla de negocio del código duro.

B. Motor de Optimización (class SustainableOptimizer)

- **Ubicación:** Celda 3.
- **Función:** El cerebro de la IA. Contiene el bucle evolutivo.
- **Componentes Críticos:**
    - **Función Fitness (fitness):** Evalúa qué tan buena es una ruta combinando Costo ($) + Penalizaciones.
    - **Evaluador de Restricciones (_check_date_constraints):** Implementa la lógica de semáforo (ver sección 5).
    - **Mecanismo de Paciencia (run):** Implementa *Early Stopping*. Si la IA no mejora en 150 generaciones, se detiene para ahorrar tiempo de cómputo.

C. Visualización y Reportes (class LogisticsVisualizer)

- **Ubicación:** Celda 4.
- **Función:** Genera los entregables visuales.
- **Tecnología:** Usa Folium para mapas geoespaciales interactivos (DualMap) y Plotly para el Dashboard Ejecutivo HTML interactivo.

4. Diccionario de Variables y Hiperparámetros

Para entender cómo "piensa" el modelo, definimos sus parámetros clave configurados en la **Celda 5 (Ejecución)**.

Hiperparámetros del Algoritmo

Parámetro	Valor	Explicación Técnica
pop_size	200	Número de calendarios alternativos que compiten simultáneamente en cada ciclo. Más población = mayor diversidad inicial.
generations	1000	Número máximo de ciclos evolutivos. Es el límite de tiempo de "aprendizaje".
mutation_rate	0.25 (25%)	Probabilidad de que una ruta sufra un cambio aleatorio. Un valor alto (25%) es vital para evitar "Óptimos Locales" (quedarse atascado en una solución mediocre).
PATIENCE_LIMIT	150	(Ubicado en Celda 3). Si tras 150 generaciones no hay mejora global, el algoritmo asume convergencia y termina. Eficiencia pura.

Métricas de Negocio (KPIs)

- **Costo Logístico ($):** Combustible Estimado + Costos Operativos ($250/km overhead).
- **Lead Time (Días):** Se asume una velocidad efectiva de logística de **60 km/h** (promedio aire/tierra) + **48 horas** de montaje/desmontaje por carrera.
- **Huella CO2:** Factor de emisión de 3.16 kg CO2 por Litro de Jet Fuel consumido.

5. Lógica de Restricciones: El Modelo de "Semáforo de Riesgo"

A diferencia de los modelos académicos que fallan ante la menor desviación, este modelo implementa una lógica de negocio realista llamada **Tiered Risk Model** (Modelo de Riesgo Escalonado).

Ubicación en Código: Método _check_date_constraints (Celda 3).

1. 🟢 **ZONA VERDE (Ideal):**
    - **Condición:** La carrera ocurre dentro de su ventana climática oficial o con una desviación ≤ 7 días.
    - **Penalización:** $0.
    - *Significado:* Operación estándar.
2. 🟡 **ZONA AMARILLA (Gestión de Riesgo):**
    - **Condición:** Desviación entre 8 y 45 días.
    - **Penalización: $2.500.000 USD** (Costo Soft).
    - *Significado:* La fecha no es ideal (ej: calor en Vegas), pero es físicamente posible. El algoritmo "paga" una multa virtual que simula el costo de mitigación (aire acondicionado, logística express). Si el ahorro en aviones supera los $2.5M, la IA tomará esta opción. **Esto es inteligencia de negocios.**
3. 🔴 **ZONA ROJA (Imposible):**
    - **Condición:** Desviación > 45 días.
    - **Penalización: $100.000.000 USD** (Costo Nuclear).
    - *Significado:* Restricción dura (Hard Constraint). Ej: Nieve en Montreal. El costo se vuelve infinito para forzar al algoritmo a descartar esta ruta.

6. Guía de Outputs (Entregables)

Al ejecutar el script en modo **Robusto ('R')**, se generan dos archivos clave en la carpeta outputs/:

A. Dashboard Ejecutivo (dashboard_ejecutivo.html)

Un panel de control interactivo diseñado para stakeholders.

- **Tarjetas KPI:** Comparativa directa (Actual vs. Optimizado) de dinero, CO2 y tiempo.
- **Gráficos Plotly:** Barras separadas para evitar distorsión de escala y curva de convergencia para validar la estabilidad del algoritmo.
- **Tabla Detallada:** El calendario final propuesto carrera por carrera.

B. Mapa Comparativo Dual (mapa_comparativo_dual.html)

Una visualización geoespacial lado a lado.

- **Izquierda (Rojo):** La ruta actual, mostrando los cruces caóticos.
- **Derecha (Verde):** La ruta optimizada, mostrando una "serpiente" fluida que recorre regiones contiguas.
- **Interactividad:** Al hacer zoom en uno, el otro se mueve sincronizado.

7. Impacto Industrial y Evolución Futura

Aplicación en la Industria Real

Este código no es exclusivo de la F1. El núcleo (SustainableOptimizer) es agnóstico y aplicable a:

- **Giras de Artistas/Eventos:** Optimización de fechas para estadios mundiales.
- **Logística Naviera:** Rutas de buques cargueros con ventanas de entrega estrictas.
- **Retail:** Rutas de distribución regional.

Siguientes Pasos (Roadmap)

1. **Transporte Multimodal:** Diferenciar tramos que se pueden hacer en tren/camión (Europa) vs. Avión (Intercontinental) para afinar el cálculo de CO2.
2. **Restricciones de Tripulación:** Añadir reglas de descanso obligatorio para pilotos y mecánicos.
3. **API Integration:** Conectar el script a una API de clima en tiempo real para validar las ventanas dinámicamente.

Este proyecto demuestra cómo la Ciencia de Datos y la Investigación de Operaciones pueden transformar decisiones estratégicas de alto nivel, convirtiendo restricciones complejas en oportunidades de ahorro millonario.