

Información General

Valor: 30 % de la calificación final del curso
Modalidad: Grupos de 2-3 estudiantes
Lenguaje: C (estándar C11 o C99)

1 DESCRIPCIÓN DEL PROYECTO

En este proyecto, los estudiantes diseñarán e implementarán un sistema inteligente de scheduling para optimizar la logística de una empresa de distribución. El sistema recibirá como entrada datos sobre entregas pendientes (ubicaciones, ventanas de tiempo, prioridades) y recursos disponibles (vehículos, capacidades, horarios), y deberá generar como salida un plan de asignación óptimo que minimice costos y maximice el servicio al cliente. El desafío central consiste en desarrollar algoritmos eficientes que asignen entregas a una flota de vehículos heterogénea, considerando múltiples restricciones operativas. Los estudiantes deberán analizar, implementar y comparar diferentes estrategias voraces (greedy), evaluando su eficacia en términos de tiempo computacional, calidad de solución y adaptabilidad a escenarios dinámicos típicos de la distribución urbana moderna.

2 OBJETIVOS DE APRENDIZAJE

- Comprender y aplicar algoritmos voraces (Greedy) en problemas de scheduling
- Desarrollar habilidades para modelar problemas de optimización con múltiples restricciones
- Implementar diferentes heurísticas y comparar su efectividad
- Analizar la complejidad temporal y espacial de las soluciones implementadas
- Evaluar trade-offs entre calidad de la solución y tiempo de ejecución
- Aplicar estructuras de datos eficientes para problemas de asignación de recursos

3 REQUISITOS TÉCNICOS

3.1 MODELO DEL PROBLEMA

Implementar las siguientes estructuras:

1. Tareas (Entregas)

- ID único
- Ubicación de origen y destino (coordenadas)
- Ventana de tiempo para entrega (inicio y fin)
- Duración estimada del servicio

- Prioridad (1-5)
- Tipo de vehículo requerido
- Volumen y peso de la carga

2. Recursos (Conductores/Vehículos)

- ID único
- Tipo de vehículo
- Capacidad (volumen y peso)
- Horario de disponibilidad
- Ubicación inicial
- Especialización (si aplica)

3. Sistema de Asignación

- Interfaz para cargar datos desde archivos
- Implementación de múltiples estrategias de scheduling
- Mecanismos para evaluar y comparar soluciones
- Visualización de resultados

3.2 ALGORITMOS DE SCHEDULING

Implementar al menos tres de las siguientes estrategias greedy:

1. Earliest Deadline First (EDF)

- Priorizar entregas con fecha límite más temprana
- Implementación con cola de prioridad

2. Shortest Processing Time (SPT)

- Priorizar entregas con menor tiempo de servicio
- Análisis de variantes (ponderadas y no ponderadas)

3. Priority-Based Scheduling

- Priorizar por importancia del cliente/pedido
- Implementar función de prioridad personalizable

4. Nearest Neighbor

- Asignar entregas basadas en proximidad geográfica
- Implementar cálculo eficiente de distancias

5. Minimum Slack Time

- Priorizar tareas con menor margen temporal
- Análisis de robustez ante retrasos

3.3 MEJORAS Y OPTIMIZACIONES

Implementar al menos dos de las siguientes mejoras:

1. Detección de Conflictos

- Identificar y resolver conflictos de tiempo/recursos
- Implementar estrategias de resolución automática

2. Balanceo de Carga

- Distribuir equitativamente el trabajo entre conductores
- Minimizar tiempos muertos y maximizar utilización

3. Refinamiento Local

- Mejorar solución inicial mediante intercambios locales
- Implementar algoritmo 2-opt o similar

4. Restricciones Adicionales

- Manejar descansos obligatorios para conductores
- Considerar zonas de tráfico o restricciones horarias

4 APLICACIÓN PRÁCTICA: SISTEMA DE OPTIMIZACIÓN LOGÍSTICA

Desarrollar un sistema completo con las siguientes funcionalidades:

1. Gestión de Datos

- Cargar/guardar datos de tareas y recursos desde archivos
- Soporte para diferentes formatos (CSV, JSON)
- Validación de datos de entrada

2. Planificación de Rutas

- Generar asignaciones óptimas según la estrategia seleccionada
- Considerar todas las restricciones del problema
- Detectar y manejar casos sin solución factible

3. Análisis de Escenarios

- Simular diferentes condiciones operativas
- Evaluar respuesta a retrasos o cancelaciones
- Comparar rendimiento de distintas estrategias

4. Métricas y Evaluación

- Calcular métricas clave:
 - Número de entregas completadas
 - Tiempo total de espera

- Distancia total recorrida
- Satisfacción del cliente (basada en prioridades)
- Utilización de recursos
- Tiempo de ejecución del algoritmo

5. Visualización de Resultados

- Mostrar cronograma de asignaciones
- Generar informes comparativos
- Exportar resultados para análisis posterior

5 CONJUNTO DE DATOS DE EJEMPLO

El sistema deberá ser probado con el siguiente conjunto de datos de ejemplo:

```
# entregas.csv
id,origen_x,origen_y,destino_x,destino_y,inicio,fin,duracion,prioridad,tipo_vehiculo,volumen,peso
E001,10.5,20.3,15.2,25.4,09:00,12:00,30,3,1,2.5,10
E002,12.1,18.7,15.2,25.4,08:30,11:30,25,4,2,1.8,8
E003,9.8,22.5,30.1,10.5,10:00,14:00,45,2,1,3.2,15
E004,5.6,15.3,12.1,18.7,08:00,10:30,20,5,3,1.0,5
...

# vehiculos.csv
id,tipo,capacidad_volumen,capacidad_peso,hora_inicio,hora_fin,pos_x,pos_y,especialidad
V001,1,10,50,08:00,18:00,0,0,0
V002,2,8,40,09:00,17:00,5,5,1
V003,1,10,50,07:00,15:00,2,3,0
V004,3,5,25,10:00,20:00,8,2,2
...
```

5.1 DESCRIPCIÓN DE LOS CAMPOS

5.1.1. Archivo de Entregas (entregas.csv)

- **id:** Identificador único de la entrega (formato: E seguido de tres dígitos)
- **origen_x, origen_y:** Coordenadas cartesianas del punto de recogida (en km desde un punto de referencia)
- **destino_x, destino_y:** Coordenadas cartesianas del punto de entrega (en km desde un punto de referencia)
- **inicio:** Hora de inicio de la ventana de tiempo para realizar la entrega (formato 24h)
- **fin:** Hora límite para completar la entrega (formato 24h)
- **duracion:** Tiempo estimado para completar el servicio de entrega (en minutos)
- **prioridad:** Nivel de importancia de la entrega (escala 1-5, donde 5 es la máxima prioridad)
- **tipo_vehiculo:** Tipo de vehículo requerido para la entrega (1=pequeño, 2=mediano, 3=grande)

- **volumen:** Volumen de la mercancía a transportar (en metros cúbicos)
- **peso:** Peso de la mercancía a transportar (en kilogramos)

5.1.2. Archivo de Vehículos (vehiculos.csv)

- **id:** Identificador único del vehículo (formato: V seguido de tres dígitos)
- **tipo:** Categoría del vehículo (1=pequeño, 2=mediano, 3=grande)
- **capacidad_volumen:** Volumen máximo que puede transportar el vehículo (en metros cúbicos)
- **capacidad_peso:** Peso máximo que puede transportar el vehículo (en kilogramos)
- **hora_inicio:** Hora en que el conductor comienza su jornada laboral (formato 24h)
- **hora_fin:** Hora en que el conductor termina su jornada laboral (formato 24h)
- **pos_x, pos_y:** Coordenadas cartesianas de la ubicación inicial del vehículo (en km desde un punto de referencia)
- **especialidad:** Habilidad específica del conductor (0=ninguna, 1=manejo de materiales frágiles, 2=manejo de materiales peligrosos, 3=entregas expresas)

5.1.3. Consideraciones Adicionales

- La distancia entre dos puntos debe calcularse utilizando la fórmula euclidiana.
- El tiempo de viaje entre dos puntos se puede estimar considerando una velocidad promedio de 50 km/h.
- Para realizar una entrega, un vehículo debe:
 - Ser del tipo requerido o superior
 - Tener capacidad suficiente (volumen y peso)
 - Estar disponible durante la ventana de tiempo
 - Tener la especialidad requerida (si aplica)
- El formato de tiempo es HH:MM en sistema de 24 horas.

6 ENTREGABLES

1. **Código fuente:** Archivos .c y .h bien organizados y comentados
2. **Makefile:** Para compilar el proyecto
3. **Conjuntos de Datos:** Archivos con datos de prueba
4. **Informe técnico:** Documento PDF (8-10 páginas) que incluya:
 - Descripción de la implementación
 - Análisis teórico de la complejidad de los algoritmos

- Resultados experimentales con comparativas
- Discusión de resultados y conclusiones

5. **Presentación:** Exposición de 10-15 minutos del trabajo realizado

7 EVALUACIÓN

La evaluación del proyecto se realizará considerando tres componentes principales:

1. **Evaluación del profesor (70 %):** Basada en la rúbrica detallada a continuación.
2. **Evaluación del informe técnico (20 %):** Se dará especial importancia a la calidad, profundidad y rigurosidad del informe técnico.
3. **Evaluación entre pares (10 %):** Cada miembro del equipo evaluará la contribución de sus compañeros según criterios de participación, responsabilidad y calidad del trabajo.

7.1 EVALUACIÓN ENTRE PARES

La evaluación entre pares se realizará mediante un formulario confidencial donde cada estudiante evaluará a los demás miembros de su equipo en los siguientes aspectos:

- Asistencia y participación en reuniones de trabajo
- Cumplimiento de tareas asignadas
- Calidad de las contribuciones
- Colaboración y disposición para ayudar
- Resolución de problemas

Cada aspecto será calificado en una escala de 1 a 7. Esta evaluación permitirá ajustar la nota individual dentro del trabajo grupal, premiando el compromiso y la calidad de las aportaciones individuales.

8 CONSEJOS Y RECURSOS

- Comience por implementar una representación adecuada del problema antes de desarrollar los algoritmos.
- Desarrolle y pruebe cada algoritmo de forma incremental.
- Utilice conjuntos de datos pequeños para validar la correctitud antes de escalar a conjuntos más grandes.
- Para la implementación de colas de prioridad, considere usar heaps binarios.
- Recursos recomendados:
 - Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2009). Introduction to Algorithms (3rd ed.). MIT Press. (Capítulos sobre algoritmos greedy)
 - Brucker, P. (2007). Scheduling Algorithms (5th ed.). Springer.
 - Pinedo, M. L. (2022). Scheduling: Theory, Algorithms, and Systems (6th ed.). Springer.

9 ESCALA DE EVALUACIÓN

Las notas parciales (Nota del Profesor y Nota del Informe Técnico) se calcularán según la siguiente escala, con interpolación lineal entre los puntos de referencia:

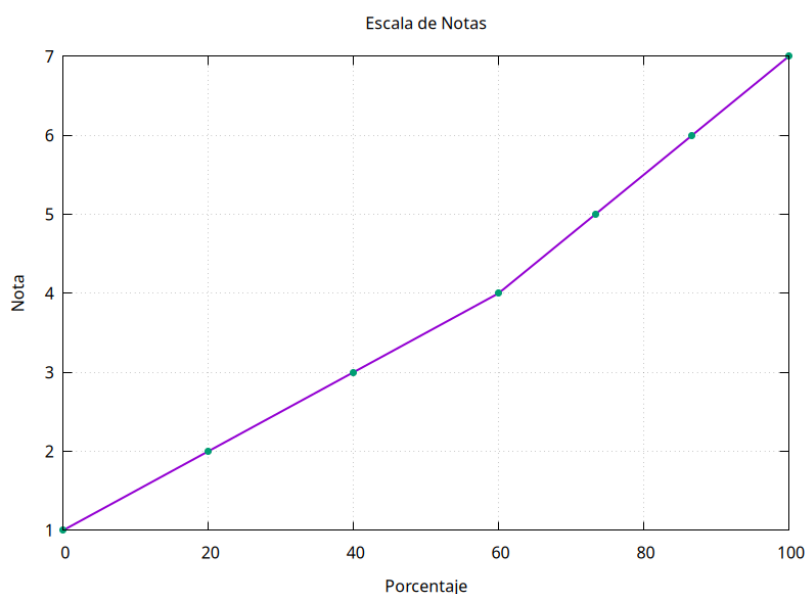


Figura 1: Escala de notas con interpolación lineal

9.1 CÁLCULO DE LA NOTA FINAL

La nota final del estudiante se calculará mediante la siguiente fórmula:

$$\text{Nota Final} = \text{Nota del Profesor} + \text{Nota del Informe Técnico} + \text{Promedio de Notas de Pares}$$

Donde:

- **Nota del Profesor (70 %):** Se obtiene aplicando la escala de evaluación mostrada en la figura al puntaje obtenido en la rúbrica.
- **Nota del Informe Técnico (20 %):** Se obtiene aplicando la misma escala de evaluación al informe técnico entregado.
- **Promedio de Notas de Pares (10 %):** Es el promedio de las evaluaciones realizadas por los compañeros de equipo.

Todas las notas parciales se escalan de 1.0 a 7.0 según la figura anterior antes de aplicar la ponderación final.

10 RÚBRICA DE EVALUACIÓN

Criterio	Excelente (90-100 %)	Bueno (75-89 %)	Satisfactorio (60-74 %)	Insuficiente (0-59 %)
Modelado (20 %)	Modelo completo y sofisticado que captura todas las complejidades del dominio.	Modelo adecuado que representa las principales características del problema.	Modelo básico funcional pero con simplificaciones significativas.	Modelo incompleto o con errores conceptuales graves.
Impl. de Algoritmos (30 %)	Impl. optimizada de múltiples estrategias greedy con análisis profundo.	Impl. correcta de al menos tres estrategias con buen análisis.	Impl. funcional de las estrategias básicas requeridas.	Impl. incompleta o con errores significativos.
Mejoras y Optim. (20 %)	Impl. innovadora de múltiples mejoras con análisis de impacto detallado.	Impl. efectiva de al menos dos mejoras con análisis adecuado.	Impl. básica de las mejoras requeridas.	Mejoras no implementadas o con funcionalidad limitada.
Sistema Completo (20 %)	Sistema robusto e intuitivo que implementa todas las funcionalidades con alta calidad.	Sistema completo que implementa la mayoría de las funcionalidades requeridas.	Sistema funcional pero con algunas limitaciones o carencias.	Sistema incompleto o con funcionalidad muy limitada.
Presentación (10 %)	Exposición clara, completa y bien estructurada. Dominio total del tema.	Buena exposición con estructura adecuada. Buen dominio del tema.	Exposición básica pero suficiente. Conocimiento general del tema.	Exposición desorganizada, incompleta o con errores graves.

Cuadro 1: Rúbrica de evaluación del proyecto

11 PREGUNTAS FRECUENTES

1. ¿Es necesario implementar todos los algoritmos mencionados?

No, se requiere implementar al menos tres de las estrategias de scheduling mencionadas y al menos dos de las mejoras propuestas.

2. **¿Cómo debemos manejar casos donde no hay solución factible?**

Su implementación debe detectar estos casos y proporcionar información sobre las restricciones que no se pueden satisfacer.

3. **¿Es obligatorio usar el formato de datos proporcionado?**

El formato proporcionado es una referencia. Pueden modificarlo si lo justifican adecuadamente en su documentación.

4. **¿Se puede usar alguna biblioteca externa para estructuras de datos?**

Se permite usar bibliotecas estándar de C para estructuras básicas, pero las implementaciones de los algoritmos de scheduling deben ser desarrolladas por ustedes.