

# Planeamento Multi-Objetivo de Rotas Multimodais no Grande Porto

Guilherme Pinto  
Universidade do Minho  
Escola de Engenharia  
Braga, Portugal  
PG60225@alunos.uminho.pt

Luís Silva  
Universidade do Minho  
Escola de Engenharia  
Braga, Portugal  
PG60390@alunos.uminho.pt

João Azevedo  
Universidade do Minho  
Escola de Engenharia  
Braga, Portugal  
PG61693@alunos.uminho.pt

Pedro Reis  
Universidade do Minho  
Escola de Engenharia  
Braga, Portugal  
PG59908@alunos.uminho.pt

**Abstract**—Este trabalho apresenta um sistema de planeamento de rotas multimodais e dependentes do tempo para o Grande Porto, integrando deslocações a pé, metro e autocarro. A rede de mobilidade é modelada como um grafo multimodal construído a partir de dados OpenStreetMap (rede pedonal) e GTFS (STCP e Metro do Porto), permitindo a ligação entre paragens e a contabilização de tempos de espera com base em horários. O problema é formulado como otimização multiobjetivo, minimizando simultaneamente o tempo total de viagem e as emissões de CO<sub>2</sub>. Para gerar alternativas de elevada qualidade e diversidade, utiliza-se uma inicialização inteligente baseada em múltiplas execuções de A\* com diferentes perfis de preferência, seguida de otimização evolutiva com NSGA-III, recorrendo a operadores topológicos de cruzamento e mutação adaptados a caminhos em grafos. Os resultados demonstram a capacidade do sistema em produzir um conjunto de rotas não dominadas que evidenciam claramente os trade-offs entre rapidez e sustentabilidade, suportando decisões informadas do utilizador em cenários urbanos reais.

**Index Terms**—Planeamento de rotas, Grafos multimodais, Otimização multiobjetivo, NSGA-III, A\* dependente do tempo, GTFS, OpenStreetMap, Transporte público, Emissões de CO<sub>2</sub>

## I. INTRODUÇÃO

A mobilidade urbana eficiente numa área metropolitana densa, como o *Grande Porto* (Área Metropolitana do Porto), exige uma integração fluida entre diferentes modos de transporte. A escolha do trajeto ideal entre origem e destino deixou de ser um problema unidimensional centrado apenas na duração: hoje envolve um *trade-off* entre rapidez e impacto ambiental. Assim, além do tempo total de viagem, torna-se relevante considerar métricas como as emissões de CO<sub>2</sub> associadas ao transporte.

Este projeto propõe e implementa um sistema de planeamento de rotas *multimodais e dependentes do tempo*, modelando a rede de mobilidade como um grafo integrado que combina caminhada, metro e autocarro. O utilizador define um ponto de partida, um ponto de chegada e uma hora, bem como preferências e restrições. Em resposta, o sistema devolve um conjunto de rotas alternativas com diferentes compromissos

entre critérios, permitindo uma decisão informada, nomeadamente em pontos e estações multimodais.

O problema é tratado como *multiobjetivo*, considerando objetivos frequentemente contraditórios:

- **Tempo:** minimizar a duração total da viagem, incluindo tempos de espera;
- **Sustentabilidade:** minimizar as emissões de CO<sub>2</sub>.

As principais contribuições deste trabalho são:

- Processamento de dados reais dos operadores (STCP e Metro do Porto) em formato GTFS para a construção da rede de transportes;
- Construção de um grafo de rede pedonal, permitindo não só a ligação entre paragens, mas também a quantificação da componente pedonal como métrica adicional para o utilizador;
- Integração multimodal num grafo único que combina a rede pedonal com a rede de metro e autocarro;
- Modelo de custos dependentes do tempo, integrando os horários das paragens e estações;
- Abordagem de otimização multi-objetivo, suportada por uma inicialização baseada em heurísticas de domínio;
- Sistema de apoio à decisão que gera múltiplas alternativas de rota, evidenciando os *trade-offs* entre tempo e sustentabilidade.

O restante documento está organizado da seguinte forma: a Secção II apresenta a metodologia de projeto adotada; a Secção III descreve a aquisição e tratamento dos dados e a construção do grafo multimodal; a Secção IV detalha os algoritmos de procura analisados e a adaptação para custos dependentes do tempo; a Secção V introduz a formulação multiobjetivo e a geração de alternativas; a Secção VI explica os algoritmos de otimização; a Secção VII reúne pontos fortes e limitações do projeto e, por fim, a Secção VIII apresenta os resultados, a discussão, o trabalho futuro e as conclusões.

## II. METODOLOGIA DE PROJETO

Este capítulo descreve a abordagem metodológica adotada para o planeamento e desenvolvimento do sistema. O projeto foi estruturado em quatro fases, maioritariamente sequenciais mas com iteração entre etapas, visando transformar um problema complexo de mobilidade urbana numa solução de software funcional e validada.

### A. Fase 1: Análise e Definição do Problema

A primeira etapa consistiu na análise dos requisitos e das restrições do problema de mobilidade no *Grande Porto*. O foco foi caracterizar a natureza **multiobjetivo** das deslocações urbanas, reconhecendo que a minimização do tempo nem sempre é o único interesse do utilizador.

Foram definidos dois eixos de otimização fundamentais:

- Tempo;
- Sustentabilidade (emissões de  $CO_2$ );

Nesta fase foi ainda efetuada a análise de fontes técnicas e documentação relevante, bem como a seleção do ecossistema tecnológico, garantindo viabilidade de implementação e reproduzibilidade.

### B. Fase 2: Estratégia de Dados e Modelação

Definidos os objetivos, a metodologia concentrou-se na estratégia de processamento e fusão da informação. Em conformidade com os requisitos do projeto, a modelação do ambiente baseou-se nos conjuntos de dados oficiais disponibilizados (*open data*), garantindo a fidelidade ao cenário:

- Utilização da infraestrutura viária para modelar as deslocações pedonais;
- Integração dos dados fornecidos em formato GTFS (*General Transit Feed Specification*) para representar a oferta, paragens e horários da STCP e Metro do Porto.

A decisão metodológica central desta fase foi unificar estas fontes num grafo multimodal, transformando dados brutos numa estrutura navegável e extensível. A concretização técnica desta modelação e o tratamento das fontes fornecidas são detalhados na Secção III.

### C. Fase 3: Desenvolvimento Algorítmico

Com o modelo de dados definido, a terceira fase dedicou-se à conceção da lógica de resolução, dividindo o desafio em dois problemas complementares:

- 1) **Motor de routing**: adaptação de algoritmos de procura para processarem um grafo de grande dimensão com custos dependentes do tempo;
- 2) **Otimização multiobjetivo**: definição de uma estratégia para gerar múltiplas alternativas, permitindo oferecer ao utilizador escolhas com diferentes compromissos entre tempo e sustentabilidade.

Os algoritmos e a sua implementação são aprofundados na Secção IV.

### D. Fase 4: Prototipagem e Validação

A última fase metodológica consistiu na materialização do sistema num protótipo funcional, unindo o motor de cálculo a uma camada de interação com o utilizador. O desenvolvimento da interface gráfica focou-se na usabilidade e na visualização geográfica, permitindo:

- A seleção intuitiva de origem e destino através de cliques no mapa ou pesquisa textual;
- A renderização visual das rotas calculadas sobre o mapa da cidade, distinguindo os segmentos pedonais dos segmentos de transporte público.
- Inspeção das métricas associadas a cada alternativa.

Por fim, a validação foi conduzida através de simulações e casos de teste no Grande Porto, verificando:

- **Connectividade multimodal**: a eficácia na alternância entre caminhada, metro e autocarro;
- **Coerência temporal**: a correta contabilização dos tempos de espera reais nas paragens;
- **Sensibilidade aos critérios**: a confirmação de que os perfis “rápido” e “sustentável” produzem trajetos efetivamente distintos.

Esta etapa permitiu testar a robustez da solução e confirmar o cumprimento dos objetivos de intermodalidade e apoio à decisão.

## III. MODELAÇÃO DA REDE E CONSTRUÇÃO DO GRAFO MULTIMODAL

O sistema de planeamento de rotas depende diretamente da fidelidade da representação digital da rede de transportes.

Neste projeto, a Área Metropolitana do Porto é modelada através de um grafo multimodal  $G = (V, E)$ , construído a partir da fusão de dados geográficos da rede viária e dados operacionais de transportes públicos.

A implementação segue uma estratégia em camadas: construção dos grafos unimodais (pedonal, autocarro e metro) e subsequente integração num multimodal utilizando conectores espaciais.

### A. Aquisição e tratamento de dados

O processo inicia-se com a ingestão de duas fontes de dados primárias, sujeitas a análise e pré-processamento específico.

#### a) Rede viária pedonal:

A infraestrutura física para deslocações a pé é obtida a partir do *OpenStreetMap* (OSM), utilizando a biblioteca OSMNX. A área de interesse é delimitada por uma *bounding box* (*Grande Porto*) e o grafo resultante é simplificado para reduzir redundâncias topológicas, mantendo apenas nós estruturais (interseções e fins de via).

Na implementação, a rede pedonal é convertida para um conjunto de segmentos onde cada aresta  $(u, v)$  retém:

- **Distância ( $d$ )**: Comprimento real do segmento em metros (*Haversine*);
- **Tempo base ( $t$ )**: Calculado assumindo uma velocidade média de caminhada ( $v = 1,38$  m/s);

- **Modo:** Identificado explicitamente como "walk".

#### b) Rede de transportes públicos:

A componente de transportes (STCP e Metro do Porto) é processada a partir dos dados GTFS (*General Transit Feed Specification*) oficiais fornecidos. O processamento destes ficheiros estáticos permite reconstruir a oferta de transporte através das seguintes tabelas críticas:

- `stops.txt`: Localização e identificação dos nós de paragem;
- `stop_times.txt`: Horários detalhados de chegada e partida, fundamentais para a modelação temporal;
- `shapes.txt`: Geometria detalhada do percurso, permitindo calcular a distância real percorrida pelo veículo entre paragens;
- `calendar.txt` e `calendar_dates.txt`: Definição dos serviços ativos (dias úteis, fins de semana, feriados).

**Normalização Temporal:** Todos os horários são convertidos para segundos a partir da meia-noite (00:00:00), facilitando as operações aritméticas no cálculo de custos de viagem e tempos de espera.

#### B. Construção de grafos unimodais

##### a) Grafo pedonal ( $G_{walk}$ ):

O grafo pedonal é instanciado utilizando a biblioteca `caNetworkX`, onde os nós representam intersecções ou fins de via e as arestas representam as ruas navegáveis.

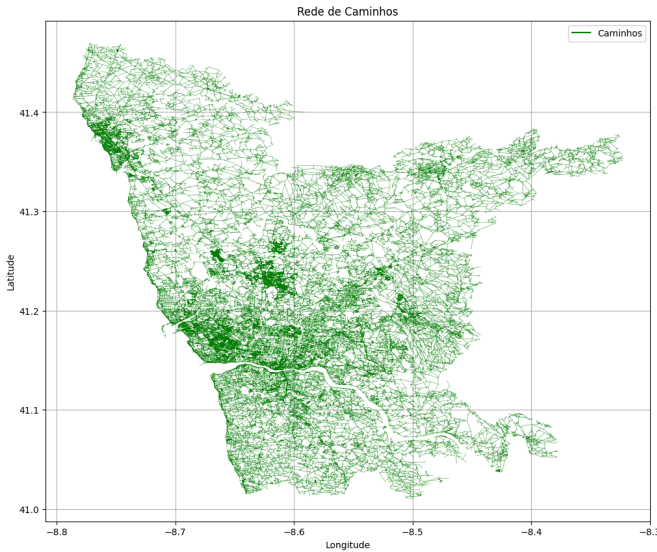


Fig. 1. Grafo da rede viária pedonal extraída do *OpenStreetMap*

##### b) Grafos de transportes ( $G_{bus}$ e $G_{transit}$ ):

Para os modos motorizados, a construção do grafo foca-se na topologia da rede, evitando a explosão de arestas que ocorreria se cada viagem fosse uma aresta individual.

A estrutura adotada coloca a informação temporal nos nós e a conectividade nas arestas:

- 1) **Nós:** Cada nó armazena as coordenadas e os horários da respetiva estação ou paragem. Con-

tém assim a lista de todas as passagens de transportes públicos por aquela paragem (extraído de `stop_times.txt`), incluindo `trip_id`, `arrival_time` e `departure_time`.

- 2) **Arestas:** Criam-se arestas direcionadas entre paragens consecutivas de uma rota. Estas arestas representam a existência de uma ligação física, mas não uma viagem específica num horário específico. A aresta guarda atributos estáticos como a `distance`, o modo ("bus" ou "transit") e o tempo de viagem.

**Cálculo de Distâncias e Emissões:** A distância entre duas paragens não é apenas a linha reta que liga as paragens ou estações. O algoritmo calcula a distância, pelo formato das ruas definida em `shapes.txt` usando lógica de *Nearest Shape Point* e soma a distância ao longo do trajeto real. As emissões são derivadas desta distância multiplicada pelos fatores de emissão específicos de cada modo (Autocarro: 109,9 g/km, Metro: 40 g/km).

**Cálculo do Tempo de Viagem:** O tempo de deslocação efetivo entre as duas paragens, definido pela diferença dos horários de chegada e partida dos nós respetivos.

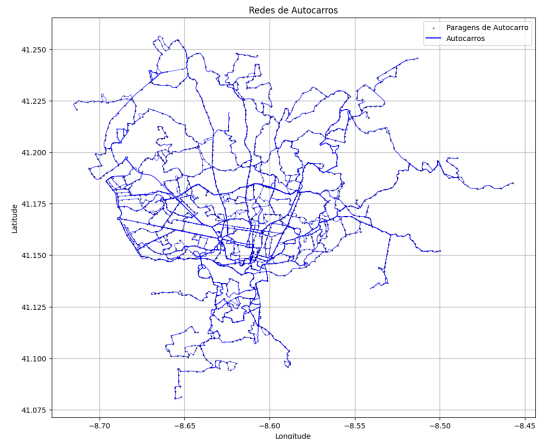


Fig. 2. Grafo da rede de autocarros

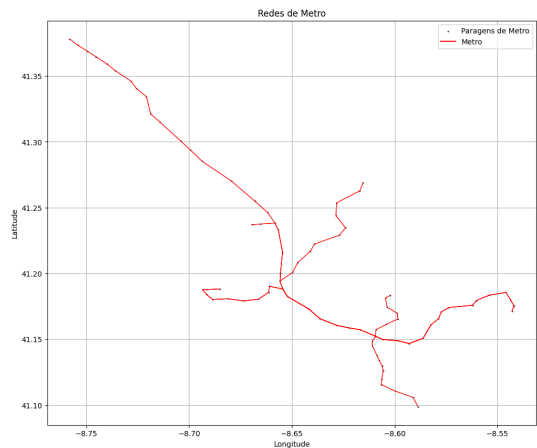


Fig. 3. Grafo da rede de metro

### C. Integração multimodal

As três camadas ( $G_{\{walk\}}$ ,  $G_{\{bus\}}$ ,  $G_{\{transit\}}$ ) são fundidas num único objeto `nx.MultiGraph`. A integridade espacial deste grafo depende da conexão entre as paragens de transporte e a rede viária, uma vez que estas possuem coordenadas distintas.

A solução implementada utiliza uma *K-Dimensional Tree* (`scipy.spatial.cKDTree`) para eficiência computacional:

- 1) Constrói-se a árvore com todos os nós da rede pedonal;
- 2) Para cada paragem de transporte, consulta-se o vizinho mais próximo na árvore (*Nearest Neighbor*);
- 3) É criada uma aresta bidirecional entre a paragem e o nó viário, com peso igual à distância real a pé.

Esta abordagem assegura que o algoritmo de procura pode transitar fluidamente entre caminhar na rua e entrar numa estação ou paragem.

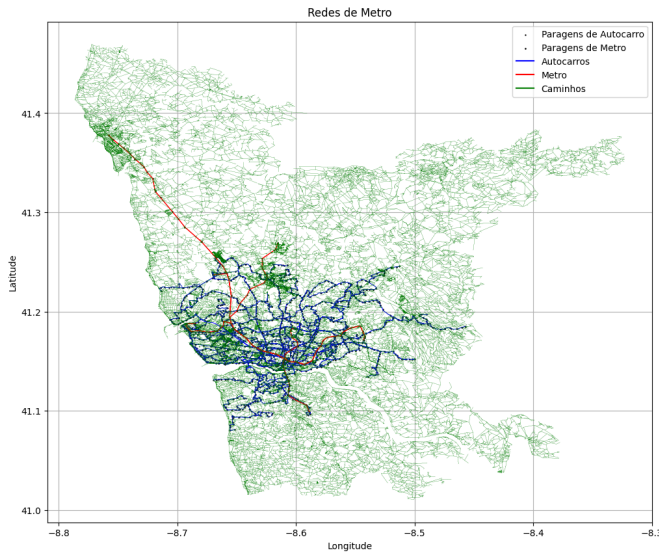


Fig. 4. Sobreposição das redes após integração multimodal.

### D. Validação topológica e refinamento da conectividade

Durante a fase de verificação da integridade do grafo multimodal, foi realizada uma análise de componentes conexos para garantir a navegabilidade em toda a área de estudo. Nesta etapa, detetou-se uma descontinuidade crítica na rede pedonal ( $G_{walk}$ ): a inexistência de ligações efetivas concelhos.

Para solucionar esta fragmentação, ao invés de extrair a topologia de cada concelho individualmente, definiu-se a área de estudo através de um polígono de delimitação geográfica (zona do *Grande Porto*).

Esta intervenção garantiu a unificação topológica do território, permitindo o cálculo correto de rotas multimodais que envolvam a travessia do rio ou travessia entre concelhos.

## IV. ALGORITMOS DE PROCURA

A resolução do problema de mobilidade no *Grande Porto* resume-se, fundamentalmente, à descoberta de um caminho

de menor custo num grafo ponderado  $G = (V, E)$ . Nesta secção descrevem-se dois algoritmos clássicos (Dijkstra e  $A^*$ ), a comparação entre eles e a adaptação desenvolvida neste projeto para lidar com a componente temporal do transporte público.

### A. Problema de caminho mais rápido

Dado um grafo ponderado com pesos não-negativos,  $w(u, v) \geq 0$ , pretende-se encontrar um caminho  $P$  de uma origem  $s$  a um destino  $t$  que minimize o custo total:

$$\text{cost}(P) = \sum_{(u,v) \in P} w(u, v) \quad (1)$$

A particularidade do caso é que, em transporte público, é necessário considerar o tempo de espera nas paragens.

### B. Algoritmo de Dijkstra

O algoritmo de *Dijkstra* é a abordagem clássica para obter o caminho de menor custo a partir de um nó origem para todos os outros nós, assumindo pesos não-negativos.

#### a) Ideia:

O algoritmo mantém, para cada nó  $n$ , o melhor custo conhecido  $g(n)$  desde a origem. Em cada iteração, escolhe-se o nó não visitado com menor  $g(n)$  e relaxam-se as suas arestas. A expansão ocorre de forma radial, explorando uniformemente em todas as direções até encontrar o destino.

#### b) Propriedades:

- **Correção:** Garante o ótimo se  $w(u, v) \geq 0$ .
- **Ineficiência:** Em redes geográficas, expande muitos nós irrelevantes antes de chegar ao alvo.
- **Complexidade:**  $O((|V| + |E|) \log |V|)$  com fila de prioridade binária.

### C. Algoritmo $A^*$

O algoritmo  $A^*$  é uma generalização do Dijkstra que utiliza informação heurística para guiar a procura na direção do destino, reduzindo o espaço de pesquisa.

#### a) Função de avaliação:

$$f(n) = g(n) + h(n) \quad (2)$$

Onde:

- $g(n)$ : custo real acumulado desde a origem até  $n$ ;
- $h(n)$ : estimativa do custo restante de  $n$  até ao destino;
- $f(n)$ : estimativa do custo total da solução que passa por  $n$ .

#### b) Heurística: Admissibilidade:

Para garantir uma solução ótima, a heurística deve ser admissível, ou seja, nunca pode sobrestimar o custo real:

$$h(n) \leq h^*(n) \quad (3)$$

Se  $h(n)$  for admissível, o  $A^*$  garante encontrar o caminho mais curto, tal como o *Dijkstra*, mas visitando menos nós.

#### D. Comparação: Dijkstra vs. $A^*$

A escolha recaiu sobre o  $A^*$  devido à necessidade de eficiência em tempo real.

TABLE I  
COMPARAÇÃO COMPARATIVA ENTRE OS ALGORITMOS DE PROCURA DIJKSTRA E  $A^*$ .

Critério	Dijkstra	$A^*$ (A-Star)
Tipo de Procura	Não informada	Informada (Heurística)
Função de Avaliação	$f(n) = g(n)$	$f(n) = g(n) + h(n)$
Garantia de Ótimo	Sim (pesos $\geq 0$ )	Sim (se $h$ admissível)
Eficiência Espacial	Explora todas as direções	Focada no objetivo
Desempenho	Lento em redes grandes	Rápido e direto

#### E. Adaptação com informação temporal do $A^*$

A implementação do algoritmo  $A^*$  neste projeto distingue-se pela gestão distribuída da informação temporal. Ao contrário de grafos puramente estáticos, onde o peso reside exclusivamente na aresta, a nossa abordagem utiliza uma arquitetura híbrida onde o custo total de transição é uma soma de componentes da aresta (tempo de viagem) e do nó (tempo de espera).

O custo de atravessar uma ligação  $(u, v)$  não é um valor fixo, mas sim uma função do instante  $t_{chegada}$  de chegada ao nó  $u$ . O peso total  $w(u, v, t_{chegada}, t_{partida})$  é calculado em tempo de execução durante a expansão do grafo:

$$w(u, v, t_{chegada}, t_{partida}) = T_{viagem} + T_{espera} \quad (4)$$

onde :

- $t_{chegada}$  corresponde ao instante de tempo que o indivíduo chegou ao nó;
- $t_{partida}$  corresponde ao instante de tempo que o indivíduo apanhou o meio de transporte no nó;
- $T_{viagem}$  corresponde ao tempo de viagem que o indivíduo demorou a chegar ao nó da paragem;
- $T_{espera}$  corresponde ao tempo de espera ( $t_{partida} - t_{chegada}$ ).

#### V. INICIALIZAÇÃO INTELIGENTE (SEEDING COM $A^*$ PARA NSGA-III)

Nesta fase, em vez de inicializar a população de forma aleatória (o que geraria muitas rotas inválidas ou desconexas), foi adotada uma inicialização inteligente baseada em heurísticas de domínio: executar o motor de procura ( $A^*$  dependente do tempo) várias vezes com diferentes perfis de preferência, e usar as rotas obtidas como população inicial do algoritmo evolutivo.

A intuição é simples: ao variar a importância relativa entre tempo e emissões, obtêm-se rotas com características distintas (por exemplo, mais rápidas ou mais sustentáveis). Assim, a

população inicial já começa diversa e de elevada qualidade, reduzindo o esforço das primeiras gerações e acelerando a obtenção de soluções úteis.

#### A. Perfis de Preferência (Pesos)

Para gerar sementes distintas, criam-se combinações de pesos para os dois critérios: Tempo e Emissões ( $\text{CO}_2$ ). Em termos práticos, foram gerados perfis ao longo do intervalo de preferências, desde “priorizar tempo” até “priorizar sustentabilidade”.

Em vez de depender de uma única execução, o sistema executa o  $A^*$  múltiplas vezes, cada vez com um perfil diferente, de modo a produzir rotas alternativas que sirvam como bons pontos de partida.

#### B. Custo Usado no Seeding

Cada execução do  $A^*$  resolve um problema de caminho mais curto usando um custo escalarizado que combina:

tempo por aresta, incluindo deslocação e (quando aplicável) tempo de espera;

emissões por aresta, dependentes do modo de transporte (caminhada assume emissão nula).

Desta forma, as sementes já refletem compromissos realistas entre rapidez e impacto ambiental, em vez de depender de rotas aleatórias.

#### C. Compatibilidade Temporal

A inicialização tem em conta a hora e o dia de partida:

A hora é convertida para um formato interno (por exemplo, segundos desde a meia-noite);

Os serviços GTFS ativos nesse dia são obtidos usando calendar e calendar\_dates, filtrando viagens disponíveis;

Durante a procura, uma transição de transporte público só é considerada válida se existir uma partida compatível com o instante corrente.

Isto evita semear a população com rotas inviáveis, como tentar usar metro/autocarro fora do período de serviço.

#### D. $A^*$ Dependente do Tempo para Geração de Seeds

Para cada perfil de preferência, executa-se uma procura do tipo  $A^*$  com custos dependentes do tempo. O estado mantido inclui:

custo acumulado do caminho;

hora corrente no nó (atualizada com tempos de deslocação e espera);

caminho parcial (lista de nós), permitindo reconstruir a rota final.

A heurística mantém-se admissível (não sobrestima), baseando-se num limite inferior do tempo mínimo até ao destino (por exemplo, distância em linha reta dividida por uma velocidade máxima). Para o termo ambiental, pode ser usada uma aproximação otimista (assumindo emissão mínima), garantindo que o seeding não elimina soluções potencialmente boas.

Para controlar o custo computacional, pode ser imposto um limite de exploração (por exemplo, número máximo de nós visitados) em cada execução de  $A^*$ .

### E. Tratamento de Falhas e Fallback

A inicialização é conservadora e robusta:

Se uma execução falhar para certo perfil, reutiliza-se a última rota válida encontrada;

Se não existir conectividade entre origem e destino, o sistema devolve pelo menos uma rota fallback (por exemplo, uma solução baseada em conectividade pedonal/geométrica), garantindo que o algoritmo evolutivo recebe uma população com o tamanho necessário;

Se o número de caminhos válidos for inferior ao tamanho pretendido, replica-se uma solução válida para preencher a amostra.

### F. Integração com o Algoritmo Evolutivo

A população inicial é devolvida como uma lista de caminhos e usada diretamente como amostra inicial (sampling) no NSGA-III. Esta estratégia fornece ao algoritmo:

soluções iniciais viáveis;

diversidade controlada por diferentes perfis de preferência;

menor dependência de mutações aleatórias nas primeiras gerações.

Em suma, a inicialização inteligente transforma o problema de gerar rotas aleatórias num processo guiado por heurísticas, acelerando a obtenção de soluções de qualidade desde o arranque.

## VI. OPERADORES DE CROSSOVER E MUTAÇÃO

Ao aplicar algoritmos evolutivos (NSGA-III em conjunto com  $A^*$ ) ao problema de encaminhamento em grafos, cada solução é representada por um caminho, isto é, uma sequência ordenada de nós. Operadores genéticos tradicionais (como os usados em vetores) não são diretamente aplicáveis, pois podem quebrar a conectividade e produzir rotas inválidas. Nesta secção descrevem-se operadores topológicos concebidos para preservar a viabilidade das rotas e introduzir diversidade.

### A. Representação

Cada indivíduo corresponde a uma rota desde a origem até ao destino, representada como uma sequência de nós (paragens/estações/pontos pedonais) em que cada par consecutivo está ligado no grafo. Esta representação garante que a rota é válida e percorrível.

### B. Operador de Crossover (Baseado em Interseção)

O crossover combina dois pais, procurando preservar subestruturas úteis (por exemplo, um bom acesso pedonal ou um segmento eficiente de transporte).

Foi adotado um crossover por interseção:

Identificam-se nós comuns internos entre as duas rotas (excluindo origem e destino);

Se existirem nós comuns, escolhe-se aleatoriamente um nó de cruzamento;

Constrói-se o descendente juntando o prefixo de uma rota até ao nó de cruzamento com o sufixo da outra rota a partir desse nó até ao destino.

Robustez: Se não existirem nós comuns, o crossover é ignorado (mantém-se uma solução válida) para evitar produ-

zir trajetos desconexos. Caso a concatenação gere ciclos ou repetição de nós, aplica-se um pós-processamento simples de remoção de ciclos locais.

### C. Operador de Mutação (Desvio Local)

A mutação introduz diversidade na população e reduz o risco de convergência prematura.

Foi implementada uma mutação por desvio local (regeneração parcial):

Selecionam-se dois pontos da rota (dois nós) que definem um subsegmento interno;

Remove-se esse subsegmento;

Recalcula-se um subcaminho alternativo entre esses dois nós através de uma procura local no grafo, de forma a obter um trajeto diferente do original;

Substitui-se o subsegmento removido pelo novo subcaminho.

Se não for possível encontrar um subcaminho viável, a mutação é ignorada e o indivíduo original é preservado, garantindo a estabilidade do processo evolutivo.

## VII. ALGORITMO DE OTIMIZAÇÃO (NSGA-III)

Dada a complexidade do espaço de procura e a necessidade de gerar uma Frente de Pareto bem distribuída, foi utilizado o algoritmo **NSGA-III (Non-Dominated Sorting Genetic Algorithm III)**, implementado através da biblioteca `pymoo`.

Ao contrário de abordagens baseadas em decomposição (como o MOEA/D), o NSGA-III utiliza o conceito de dominância de Pareto em conjunto com um conjunto de direções de referência que orientam a diversidade das soluções ao longo da frente. Desta forma, é possível obter um conjunto equilibrado de rotas que representam diferentes compromissos entre tempo total de viagem e emissões de  $\text{CO}_2$ .

### A. Representação das Soluções

Cada indivíduo da população representa uma rota válida entre um ponto de origem e um destino, composta por uma sequência de nós (paragens, estações ou segmentos pedonais) e respetivos modos de transporte.

A avaliação de cada solução considera dois critérios principais:

- **Tempo total de viagem:** Incluindo caminhada, espera e deslocação nos transportes;
- **Emissões de  $\text{CO}_2$ :** Com base no modo de transporte utilizado em cada segmento.

### B. Ciclo Evolutivo

Em cada geração, o NSGA-III executa as seguintes etapas principais:

a) **1. Seleção e Reprodução:** São selecionados dois indivíduos da população com base na classificação por dominância e diversidade. De seguida, aplicam-se os operadores:

- **Cruzamento (PathCrossover):** Combina partes de duas rotas compatíveis, gerando um novo percurso híbrido;
- **Mutação (PathMutation):** Altera um segmento da rota, substituindo parte do trajeto por um percurso alternativo.

b) 2. *Avaliação*: Cada novo indivíduo (rota) é avaliado de acordo com os dois objetivos: tempo e CO<sub>2</sub>.

c) 3. *Seleção de Sobreviventes*: O conjunto combinado de pais e descendentes é classificado por dominância. O NSGA-III utiliza as direções de referência para manter a diversidade, assegurando que a Frente de Pareto permaneça bem distribuída.

O processo repete-se até atingir o número máximo de gerações, produzindo um conjunto final de rotas não dominadas que representam alternativas viáveis para o utilizador.

### C. Parâmetros da Otimização

A configuração utilizada nos testes foi a seguinte:

- **Tamanho da população**: Igual ao número de direções de referência ( $\approx 100$  indivíduos);
- **Número de gerações**: 50;
- **Probabilidade de cruzamento**: 0.8;
- **Probabilidade de mutação**: 0.3.

Esta configuração revelou um bom equilíbrio entre diversidade de soluções e tempo de execução, garantindo que o conjunto final apresente opções rápidas e sustentáveis.

## VIII. PONTOS FORTES E LIMITAÇÕES

A implementação do sistema de planeamento multimodal para o Grande Porto permitiu identificar diversas mais-valias na abordagem adotada, bem como constrangimentos inerentes à complexidade do problema.

### A. Pontos Fortes

- **Multimodalidade e Conectividade**: A fusão de dados heterogêneos resultou num grafo unificado robusto. A intervenção manual na topologia resolveu falhas críticas de dados abertos, garantindo a navegabilidade pedonal entre Porto e Gaia.
- **Apoio à Decisão (Trade-offs)**: A utilização do algoritmo NSGA III permitiu gerar não apenas uma solução, mas um conjunto de alternativas. Isto fortalece o utilizador, permitindo-lhe visualizar claramente o “custo” ambiental de ganhar alguns minutos, promovendo escolhas mais conscientes e sustentáveis.
- **Inicialização Inteligente**: A estratégia de utilizar o  $A^*$  dinâmico para gerar a população inicial demonstrou-se eficaz na redução do tempo de convergência do algoritmo evolutivo, evitando o desperdício computacional em gerações iniciais de baixa qualidade.

### B. Limitações

- **Natureza Estática dos Dados (Ausência de Real-Time)**: O sistema baseia-se em horários planeados (GTFS estático). Não considera atrasos em tempo real, supressão de veículos ou congestionamento de trânsito, fatores que afetam significativamente a fiabilidade dos autocarros em ambiente urbano.
- **Limitação do Tempo**: Uma limitação fundamental do sistema reside no facto de o algoritmo de procura não explorar a componente temporal presente nos dados modelados. Embora a arquitetura do grafo integre a

informação topológica das redes de transporte público (GTFS), o algoritmo de otimização não realiza a consulta dinâmica aos horários de partida durante a expansão dos nós.

- **Ausência de Altimetria**: O modelo atual assume um plano 2D. No entanto, a orografia do Porto é acidentada. A falta de dados de elevação no cálculo do custo pedonal ignora o esforço físico extra de subir ruas íngremes, o que pode levar a sugestões de rotas “curtas” mas fisicamente exaustivas.

## IX. CONCLUSÃO

O presente projeto abordou o desafio da mobilidade urbana no Grande Porto através do desenvolvimento de um sistema de planeamento de rotas multimodal e multiobjetivo. A aplicação de técnicas de Computação Inteligente permitiu transformar dados brutos de transporte (GTFS) e geografia (OSM) em informação útil para o cidadão.

A implementação dos algoritmos de procura  $A^*$  e de otimização evolutiva NSGA III cumpriu os objetivos propostos. Foi possível demonstrar que é viável calcular rotas eficientes que equilibram o tempo de viagem com a sustentabilidade ambiental. A comparação entre algoritmos evidenciou que, embora o *Dijkstra* garanta a otimização em grafos simples, a heurística do  $A^*$  e o mecanismo de seleção ambiental do NSGA-III baseado em direções de referência são essenciais para lidar com a complexidade combinatória e temporal de uma rede metropolitana.