

Relatório do Sprint D de Algoritmia Avançada

**Turma: 3DD Grupo 2**

**Membros do grupo:**

1181255 Nuno Teixeira

1181498 Rafael Barbarroxa

1181500 Vítor Crista

1181529 Diogo Sousa

**Docente/Orientador:**

António Silva (ASS)

**Unidade Curricular:**

Algoritmia Avançada



Índice:

[1. Introdução 3](#_Toc62414695)

[2. Representação do conhecimento do domínio 4](#_Toc62414696)

[3. Operadores de cruzamento e mutação implementados 5](#_Toc62414697)

[4. Chamada do Algoritmo Genético 6](#_Toc62414698)

[5. Chamada do Algoritmo A-Star 6](#_Toc62414699)

[6. Deteção automática de hard constraints 7](#_Toc62414700)

[7. Correção automática de hard constraints 8](#_Toc62414701)

[8. Estado da Arte - Robótica Inteligente 9](#_Toc62414702)

[9. Conclusão 11](#_Toc62414703)

[Referências 12](#_Toc62414704)

Índice de Figuras:

[Figura 1- Criar tuples 5](#_Toc62414174)

[Figura 2- Atribuir drivers a vehicle duty 5](#_Toc62414175)

[Figura 3- Chamada do algoritmo genético 6](#_Toc62414176)

[Figura 4- Chamada do algoritmo aStar 7](#_Toc62414177)

[Figura 5- Deteção de hard constraints 7](#_Toc62414178)

[Figura 6- Tarefas associadas à I.A. 9](#_Toc62414179)

# Introdução

O Sprint D de ALGAV divide-se em duas partes, a afetação de todos os motoristas a todos os autocarros e correspondente deteção e correção de hard constraints no escalonamento dos motoristas nos serviços de viatura, assim como, um estado de arte do uso de robótica inteligente aplicada ao problema do transporte público terrestre.

Contrariamente aos Sprints anteriores, não foi fornecido nenhum algoritmo base, ou seja, as escolhas realizadas são apenas algumas das possibilidades para resolução do problema apresentado.

Apesar disso, foi seguido um processo regular tendo em conta as instruções dadas, onde se começou por atribuir motoristas aos serviços de viatura, serviços esses que foram alvo de análise pelo algoritmo genético do Sprint C. Após distribuídos os motoristas, é necessário detetar não só as hard constraints que estão presentes na lista final, como verificar se os motoristas que estão a afetados a vários serviços e viatura têm tempo para se mover de um ponto de rendição até outro, utilizando o algoritmo A-Star do Sprint B. Finalmente, quaisquer problemas encontrados serão automaticamente resolvidos, ou caso o mesmo não seja possível, deve-se avisar o utilizador para que sejam resolvidos manualmente.

No que diz respeito à componente teórica, foi criado um pequeno relatório introdutório às tecnologias provenientes da robótica inteligente no âmbito dos transportes públicos. Isto foi possível através da análise de vários estudos científicos sobre a aplicação de robôs, desde sensores a veículos autónomos, com o objetivo de facilitar e agilizar autonomamente processos na área.

# Representação do conhecimento do domínio

O domínio para este Sprint de ALGAV engloba a totalidade dos factos presentes nos Sprints B e C, tendo em conta que estes são necessários para a realização dos algoritmos A-Star e Genético, respetivamente.

* **Nó**: ponto da rede de transportes.

Representados pelo facto **no/6**, correspondente a nó (nome da paragem, abreviatura da paragem, flag ponto de rendição, flag estação de recolha, longitude, latitude).

Durante a realização do projeto a navegação entre nós era feita através da abreviatura da paragem, sendo essa utilizada para a representação de nós em outros factos. As flags de ponto de rendição e estação de recolha são essenciais, porque são estas que permitem a criação de ligações.

* **Linha**: Corresponde a um conjunto de percursos que são trajetos coincidentes com as sequências de nós. Contrariamente aos nós, as linhas não são equivalentes nos domínios de LAPR5 e ALGAV, consistindo numa junção de linhas e percursos.

Assim sendo, **linha/5** tem como parâmetros linha (nome da linha, número do percurso, lista das abreviaturas das paragens, tempo minutos, distância metros).

* **Horário:** definição horária de um conjunto de percursos.

Representado pelo facto **horario/6**, correspondente a horario (identificador de percurso, identificador de viagem, lista do tempo de passagem a cada nó do percurso).

* **Bloco de trabalho/Work Block**: conjunto de viagens feitas, sem interrupções, pela mesma viatura, e com ou sem interrupções, pelo mesmo tripulante.

Os blocos de trabalho podem agregar uma ou mais viagens e estão conectados aos horários, já que, o tempo de partida do bloco corresponde ao primeiro elemento da lista de passagem do horário menos 120s, e o tempo de término ao último (das viagens respetivas).

Assim sendo, o bloco de trabalho é representado em PROLOG por workblock (identificador de bloco de trabalho, lista de identificadores de viagens, tempo de partida, tempo de término).

* **Serviço de viatura/Vehicle Duty:** corresponde ao período de trabalho diário de uma viatura.

Representado por vehicleduty (identificador de serviço de viatura, lista de identificadores de bloco de trabalho), o serviço de viatura vai ser o meio com o qual se faz a ligação entre os tripulantes que o vão executar e os blocos de trabalho.

* **Lista de motoristas para o serviço de viatura:** não tem representação no domínio de LAPR5, mas é necessário para a resolução do problema, permitindo saber quantos workblocks vão ser executados por um determinado condutor para um serviço de viatura.

Corresponde ao facto **lista\_motoristas\_n\_workblocks** (identificador de serviço de viatura, lista de condutores e respetivos número de workblocks a executar).

* **Motorista/Driver:** criado no âmbito de averiguar quando é que o motorista entra e sai do trabalho, driver (identificador de driver, data de início do trabalho, data de fim do trabalho).

# Operadores de cruzamento e mutação implementados

Inicialmente, antes da chamada do algoritmo genético é necessário criar condições para que o mesmo funcione. Para tal, têm de ser criados os factos **lista\_motoristas\_n\_workblocks** que estão em falta, associando drivers aos vehicle duties guardados.

Para auxiliar o processo, primeiramente, para cada vehicle duty foram gerados factos **rangevd** (Identificador do vehicle duty, Tempo de início do vehicle duty, Tempo de fim do vehicle duty), que é utilizado para calcular a carga, isto é, a soma de todas as horas dos vehicle duty.

Após isso, é necessário calcular a soma dos tempos de trabalho de todos os drivers e verificar se esse valor é superior (1.2x superior) à carga dos vehicle duties, de maneira a verificar se o serviço pode ser realizado pelos drivers guardados no sistema. Isso foi facilitado através da criação dos factos **horáriosmotorista** (Identificador do driver, Tempo de início de trabalho, Tempo de fim de trabalho, Blocos de tempo dos drivers).

Seguidamente, com base nos factos criados, são gerados **tuples** com a disponibilidade de todos os drivers contendo a informação tuples (Tempo de início do bloco, Tempo de fim do bloco, Tempo em segundo de trabalho, Identificador do driver).

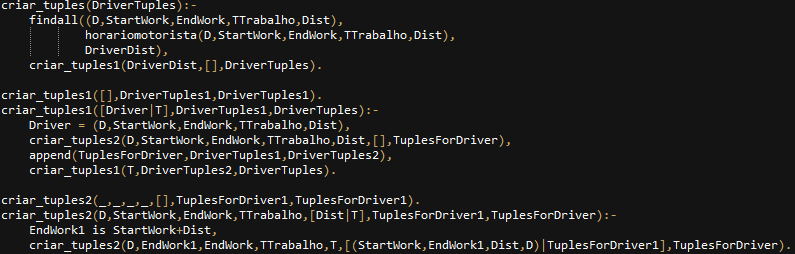


Figura - Criar tuples

Finalmente, é possível atribuir os drivers ao vehicle duties, colocando os drivers que começam a trabalhar mais cedo nos vehicle duties que se iniciam mais cedo, e, alternadamente, os vehicle duties que terminam mais tarde vão buscar os drivers que terminam mais tarde. Ordenando a lista de tuples de forma crescente pelo 1º atributo ou de forma decrescente pelo 2º, respetivamente, até todos os vehicle duties estarem preenchidos.

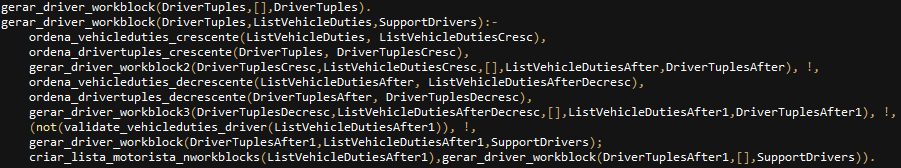


Figura - Atribuir drivers a vehicle duty

# Chamada do Algoritmo Genético

Agora que todos os vehicle duties estão preenchidos com drivers e que todos os factos **lista\_motoristas\_n\_workblocks** foram criados, o algoritmo genético desenvolvido no sprint passado pode finalmente ser chamado. Assim sendo, é necessário preencher os factos que anteriormente seriam preenchidos pelo utilizador, nomeadamente o número de gerações (100), número de elementos da população (10), probabilidade de cruzamento (0.75), probabilidade de mutação (0.05) e vehicle duty em análise, que contrariamente aos outros que são constantes, é atualizado sempre que a lista de vehicle duties é iterada e um novo está a ser escalonado.

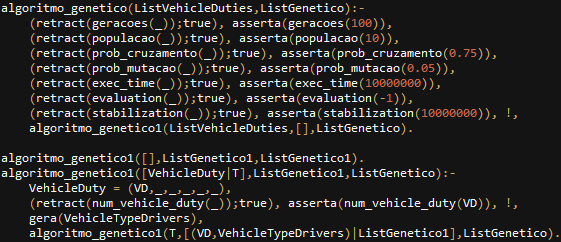


Figura 3- Chamada do algoritmo genético

# Chamada do Algoritmo A-Star

Após obtidos os resultados do escalonamento de drivers para todos os vehicle duties, os driver duties resultantes são propícios a ter problemas, nomeadamente o mesmo driver estar a realizar dois blocos de trabalho ao mesmo tempo, ou, caso o driver trabalhe em vários vehicle duties, pode não ter tempo para viajar de um ponto de rendição até outro. Para detetar esse tipo de problemas, para que depois possam ser corrigidos, é utilizado o algoritmo A-Star do Sprint B.

Com isso em mente, primeiramente é necessário verificar quais dos drivers trabalham em vários vehicle duties e, após isso, quais os workblocks onde estão a trabalhar obtendo uma lista contendo (Identificador de driver, Lista (Tempo de início do workblock, Tempo de fim do workblock, Identificador de workblock, Identificador de vehicle duty respetivo)).

Através desta lista podemos verificar se existe alguma incongruência com os workblocks, ordenando-a de forma crescente do tempo de início e comparando a pares. Sempre que se verifica uma mudança de vehicle duty, onde não há sobreposição de horários é necessário confirmar se o driver tem tempo de chegar do ponto de rendição onde está até ao ponto de rendição onde vai prestar serviço.

É aqui, que tem de ser usado o algoritmo A-Star para calcular o tempo de um ponto ao outro, mas para tal é primeiro necessário saber quais são esses pontos, sendo facilmente encontrados, percorrendo os factos existentes no sistema. Assim sendo, através dos identificadores de workblocks a que temos acesso podemos chegar à lista de identificadores de trips, através dos quais podemos conhecer os identificadores de paths através dos factos **horários** e finalmente através dos paths sabe-se as linhas e consequentemente os seus nós. Conhecendo agora os nós basta chamar o algoritmo do Sprint B e verificar se o tempo resultado é inferior ao tempo de início do workblock.

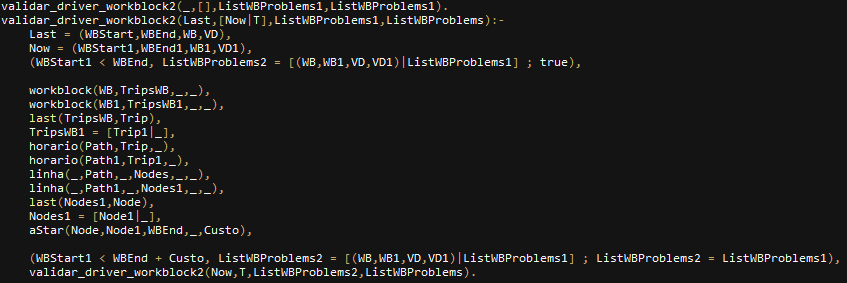


Figura 4- Chamada do algoritmo aStar

# Deteção automática de hard constraints

Para além, dos problemas anteriormente detetados, os drivers duties resultantes do escalonamento são propícios a ter outras hard constraints que não conseguiram ser resolvidas no algoritmo genético.

No âmbito de detetar esses problemas, foi desenvolvido um algoritmo que, um a um, avalia os drivers duties de maneira a identificar quais os problemas e quais os drivers que geram o problema. Isso é possível com um algoritmo idêntico ao avalia utilizado no algoritmo genético que diferencia os vários tipos de hard constraints, nomeadamente hora de jantar e almoço, o driver conduzir mais de 4 horas seguidas e não ter descanso de 1 hora após conduzir um período de 4 horas.

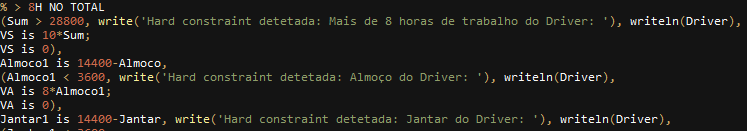


Figura 5- Deteção de hard constraints

Sempre que encontrado um problema é guardado o identificador de driver, o tempo de início e fim associado ao mesmo, assim como o identificador de vehicle duty, são guardados para que posteriormente possam ser resolvidos.

# Correção automática de hard constraints

Finalmente, agora que as hard constraints foram identificadas, é possível resolvê-las utilizando os drivers de suporte, isto é, no momento de associação de drivers aos vehicle duties, nem todos os drivers são alocados, ou seja, existe um grupo de drivers que não estão associados a nenhum bloco de trabalho e podem ser utilizados para resolver as hard constraints.

Assim sendo, a lista que guarda os problemas detetados é iterada caso a caso, em os drivers problemáticos são substituídos por drivers de suporte nos driver duties, de forma genérica resolvendo os problemas.

No caso dos problemas criados por drivers que não têm tempo de chegar de um ponto a outro resultantes do uso do algoritmo A-Star, a solução é idêntica, iterando-se a lista e substituindo os drivers por outros que não estão a trabalhar. Estes problemas são geralmente raros devido à maneira que os drivers são associados a vehicle duties.

Caso a resolução de algum problema não seja possível, o utilizador é avisado de maneira a ser resolvido manualmente.

# Estado da Arte - Robótica Inteligente

A **Inteligência Artificial (IA)** é a combinação de algoritmos, projetados para criar máquinas que tenham as mesmas capacidades que o ser humano. Por outras palavras são sistemas que realizam ações que, se efetuadas por humanos, seriam consideradas inteligentes.

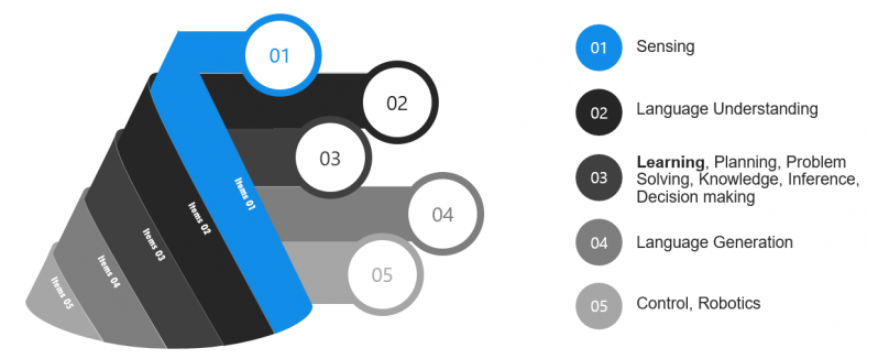
Ela pode estar dividida em várias tarefas:

Figura 6- Tarefas associadas à I.A.

Dentro destas tarefas em cima descritas a que vamos abordar é a relacionada à robótica, mais concretamente a **robótica inteligente**, que é um dos domínios da IA.

A **Robótica Inteligente** é a área que conjuga a Robótica com a IA, permitindo atribuir aos Robots capacidades inteligentes, entre as quais que eles interpretem o meio envolvente a partir da informação de sensores, efetuem raciocínios elaborados e atuem em mundos não estruturados, aprendendo com a interação entre esses ambientes.

Os Robots Inteligentes combinam comportamentos deliberativos com comportamentos reativos, de modo a poderem atingir os seus objetivos e conseguirem lidar com situações não planeadas.

No âmbito da gestão e planeamento de transportes públicos irão ser abordados vários exemplos de como a Robótica Inteligente pode ser deveras benéfica para o nosso quotidiano.

**SENSOR DE POLUIÇÃO:**

Este sensor que é implantado, de modo a se poder monitorizar a poluição do ar, é usado num sistema de transporte público, nomeadamente um autocarro. Com isto, é formulado o Bus Sensor Deployment Problem (BSDP), para selecionar quais as rotas de autocarro em que os sensores devem ser implantados, utilizando-se o Chemical Reaction Optimization (CRO) para o resolver.

CRO é uma meta-heurística projetada para resolver uma ampla gama de problemas de otimização, utilizando dados do mundo real desde as rotas de autocarro, tendo-se mostrado bastante eficiente. Assim evita-se a disposição de diversos sensores ao longo do veículo, fazendo a sua minimização e ainda gerando uma cobertura satisfatória.

**VEÍCULOS AUTÓNOMOS:**

A implementação de veículos autónomos, nomeadamente autocarros, oferece inúmeras vantagens entre as quais uma viagem mais segura, mais barata, mais confortável, mais sustentável, e, portanto, iria-se poder reduzir substancialmente os custos generalizados da mesma. Além disso, caso todas estas suposições se viessem efetivamente a concretizar, isto não só iria revolucionar o transporte, como também mudar drasticamente a forma urbana.

Ao reduzir substancialmente o custo generalizado das viagens, isso pode induzir a uma maior procura das mesmas, levando assim ao impulsionamento de uma nova onda de suburbanização e expansão urbana.

**ROBOT ASSISTENTE:**

Isto trata-se de um robô assistente que vai permitir auxiliar as pessoas à entrada das estações de metro ou de comboio, caso necessitem do acesso a alguma informação relevante. No entanto, para isto resultar, é necessário que haja uma interação humano-robô socialmente aceitável, por isso o reconhecimento humano, desde a sua posição à direção do corpo em torno do robô são de extrema importância.

Para isso é então usado uma câmara RGB-D, que permite detetar a postura de uma pessoa e um conjunto de lasers inseridos à altura da canela para fazer uma estimativa da sua direção corporal.

**SENSOR INFORMATIVO:**

Este sensor que é colocado à entrada das estações de metro ou comboio, permite fazer um reconhecimento de rosto inteligente, bem como um registo de temperatura, sendo uma solução versátil e descomplicada, principalmente para a situação pandémica atual em que nos encontramos. Contém uma câmara de infravermelhos binocular altamente eficiente, que permite um reconhecimento em pouquíssimo tempo.

Para além disso, o seu monitor também permite fornecer informações ao utilizador, desde horários de funcionamento, sendo por isso ideal para transportes públicos devido às suas capacidades de interação intuitivas e rápidas.

# Conclusão

A realização do Sprint D testou na totalidade os conhecimentos de domínio dos sprints anteriores, assim como um completo domínio dos algoritmos genético e A-Star que foram parte integral para resolução do problema.

Em relação à associação dos motoristas aos vehicle duties, apesar de não estritamente necessário o uso de factos dinâmicos como contentores da informação, foi usado para realização dos cálculos e distribuições, sendo que foi um facilitador que permitiu a realização dos processos de uma maneira muito mais simples e com menos complexidade.

A deteção de hard constraints é parte integral do problema e foi executada utilizando todas as validações estipuladas apesar de algumas como por exemplo o driver ter mais de 8h de trabalho não serem possíveis por causa da maneira que estes são distribuídos pelos vehicle duties. No caso de os drivers trabalharem em vários vehicle duties permitiu o uso do A-Star não pelo caminho gerado, mas pelo tempo usado para o percorrer.

Detetados os problemas, a correção dos mesmos foi facilitada graças aos drivers de suporte terem sido anteriormente guardados no momento da associação, permitindo com simples trocas de driver a correção de hard constraints nos driver duties. Apesar disso, algumas correções de maior magnitude não são possíveis ser resolvidas, como por exemplo, drivers estarem a trabalhar fora do seu horário de trabalho que implicaria mudanças drásticas no driver duty. Estes tipo de problemas podem ser evitados correndo o algoritmo genético durante mais tempo, ou através de heurísticas no início do mesmo que ordenem os drivers por horário, algo que o grupo não conseguir implementar no sprint passado.

Em relação à componente teórica, a robótica inteligente tem sido uma área de grande desenvolvimento e crescimento económico nas suas várias vertentes. Os robôs inteligentes têm demonstrado serem ferramentas de elevada importância nesta era moderna, sem os quais muito dificilmente poderíamos igualar a qualidade de vida que hoje possuímos.

A autonomia tem sido um fator preponderante no desenvolvimento de robôs dotados de uma maior inteligência. Permitindo assim que estes consigam executar funções com a mesma destreza que nos, seres humanos, assumimos como inato. Algo que pode corroborar isso mesmo são alguns dos exemplos inumerados em cima, de como a robótica inteligente associada ao contexto dos transportes públicos em específico pode trazer diversos benefícios para o nosso dia a dia, facilitando as nossas vidas em diversos aspetos, sendo por isso mesmo um tema fulcral e que deveria ser cada vez mais desenvolvido e implementado no nosso mundo.

**Distribuição de tarefas**:

Diogo Sousa: Estado de Arte.  
Nuno Teixeira: Associação dos motoristas aos vehicle duties.  
Rafael Barbarroxa: Chamada adequada do algoritmo de cálculo de mudança de motoristas entre um ponto de rendição/recolha e outro.  
Vítor Crista: Deteção automática de hard constraints no driver duties gerados.

# Referências

*Yu J, Li V,Lam A, 2015, 'Sensor Deployment for Air Pollution Monitoring Using Public Transportation System',*

Ramos, C. (s.d.). *6-Domínios da IA 6.2-Robótica Inteligente.*

*Meyer, Jonas; Becker, Henrik; Bösch, Patrick M.; Axhausen, Kay W, 2017, 'Autonomous vehicles The next jump in accessibilities? ETH Library Autonomous Vehicles: The next Jump in Accessibilities?'*

*Yorozu A, Takahashi M, 2020, 'Estimation of body direction based on gait for service robot applications'*

*Miguel Guevara, 2020, CCG, “Disponível em: https://www.ccg.pt/como-trazer-a-inteligencia-artificial-para-a-sua-empresa/”*

*Ana Pereira, 2020, Robotica, “Disponível em:* *https://www.robotica.pt/produtos-e-tecnologias/pc-com-painel-inteligente-da-holitech-na-rutronik/"*