Relatório do Sprint C de ALGAV

Grupo 38

Catarina Magalhães (1201192)

Nuno Alves (1201427)

Miguel Póvoas (1201716)

Guilherme Melo (1211008)

Miguel Alecrim (1200621)

Índice

[1. Breve introdução do trabalho proposto 3](#_Toc155128377)

[2. Algoritmo Genético 4](#_Toc155128378)

[2.1. Melhor(es) indivíduos passam para a geração seguinte e método de seleção não elitista 4](#_Toc155128379)

[2.2. Evitar que a sequência de cruzamentos seja sempre entre elementos em sequências já definidas 2 a 2 6](#_Toc155128380)

[2.3. Condições de término 7](#_Toc155128381)

[3. Complexidade 7](#_Toc155128382)

[4. Estado de Arte 8](#_Toc155128383)

[4.1. Machine Learning 9](#_Toc155128384)

[4.2. Deep Learning 9](#_Toc155128385)

[4.3. Processamento de Linguagem Natural (PLN) 9](#_Toc155128386)

[4.4. Pros e Contra 12](#_Toc155128387)

[4.4.1. Vantagens 12](#_Toc155128388)

[4.4.2. Desvantagem 12](#_Toc155128389)

[5. Bibliografia 12](#_Toc155128390)

# Breve introdução do trabalho proposto

O presente relatório irá dar continuidade ao trabalho anteriormente realizado na Unidade Curricular (UC) de Algoritmia Avançada.

Após a realização da otimização proposta no Sprint passado (Sprint B) é descrito que o robot poderá realizar vários tipos de tarefas dentro dos edifícios onde opera. Estas tarefas puderam ser vigilância, desinfeção e entrega de um item e são caracterizadas por terem um ponto inicial e um ponto final.

Nas tarefas do tipo vigilância ou desinfeção os pontos iniciais e finais de cada tarefa estão sempre no mesmo corredor interno de um piso de um edifício e o caminho mais curto entre os pontos de partida e chegada são suficientes já que os dispositivos de vigilância e desinfeção são rotativos, tratando todo o ambiente próximo à volta do robot.

Nas tarefas do tipo entrega o ponto inicial e final poderão ser de edifícios diferentes.

Assim, neste Sprint (Sprint C), é-o pedido para gerar, a partir do Algoritmo Genético (AG) facultado pelos docentes da UC, um plano de atendimento das tarefas indicadas que indique uma sequência de execução das tarefas e escolha a tarefa que menor tempo durar. Neste mesmo ponto, iremos realizar um estudo de complexidade que permite concluir até que dimensão de número de tarefas se pode usar essa abordagem.

Para complementar o trabalho é feito o Estado de Arte sobre AI. Nesta parte do relatório tratamos de dar a conhecer um pouco do que é Inteligência Artificial. Descrevemos o que é a IA, damos um exemplo da utilização de uma ferramenta desse tipo, especificamente ChatGPT, e concluímos com os prós e contras das ferramentas desse tipo.

# Algoritmo Genético

O Algoritmo Genético (AG) fornecido pelos docentes, no PowerPoint “Apoio ao Sprint C do Trabalho Prático de ALGAV – Planeamento da Trajetória de Robots dentro de edifícios conectados – 6(Algoritmo Genético), apresenta um exemplo pouco otimizado do Algoritmo Genético necessário para criar a sequência de execução das tarefas.

Assim, a parte relacionada com o AG terá como foco a sua melhoria. Para tal, iremos aplicar as correções necessárias para os aspetos citados no PowerPoint já falado.

As correções sugeridas são as seguintes:

* Garantir que, pelo menos, o melhor indivíduo entre a população corrente e a nova população gerada passa para a população seguinte,
* Evitar que a sequência de cruzamentos se dê sempre entre a 1º e 2º elementos da população, depois entre a 3º e a 4º, 5º e 6º.
* Aplicar um método de seleção que não seja puramente elitista, dando alguma probabilidade de um individuo com pior avaliação possa passar à geração seguinte, mesmo não estando entre os DP (dimensão da população) mais bem avaliados da população anterior e dos seus descendentes
* A condição de término poder ser diferente (para além de ser por nº de gerações podemos considerar outras possibilidades tais como tempo, estabilização da solução, etc).

Iremos analisar cada ponto e mostrar a respetiva resolução para a correção do mesmo.

## Melhor(es) indivíduos passam para a geração seguinte e método de seleção não elitista

Neste ponto do relatório iremos nos focar em dois dos pontos das correções sugeridas. O Algoritmo Genético fornecido pelos docentes substitui a toda a população corrente pelos seus descendentes; assim, não há garantia que o melhor individuo da anterior geração passe para a geração seguinte.

Caso seja pretendido que o melhor, ou alguns dos melhores indivíduos passem para a próxima geração basta, então, encontrar numa lista que junte os elementos da população anterior com os seus descendentes obtidos através dos predicados cruzamento e mutação.

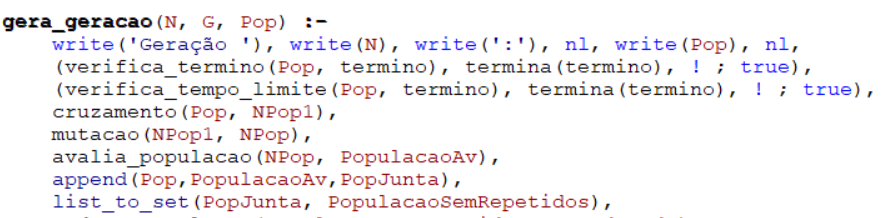
Se for necessária a seleção dos indivíduos restantes da lista, podemos fazê-lo através de um predicado de escolha que não seja puramente elitista.

Este predicado que faça a seleção dos indivíduos que faltam da lista é feita através da junção dos N indivíduos da população atual com os seus descendentes obtidos pela mutação e cruzamento já referido.

Ordenamos de acordo com a avaliação de cada um e escolhemos os P primeiros.

Retiramos esses elementos da lista e criamos uma lista com os restantes T-P indivíduos associando a cada um o produto da avaliação por um número aleatório gerado entre 0 e 1 e depois ordenamos os indivíduos dessa nova lista em ordem crescente de acordo com esse produto. Passamos os N-P primeiros elementos dessa lista para a geração seguinte.

Iremos agora explicar com o código para a solução.



Na figura 1, como se pode ver, nós utilizamos dois predicados do SWI-Prolog: o predicado append/3 e o list\_to\_set/2. O primeiro dos predicados vai juntar os indivíduos de ambas as listas e o segundo vai remover os repetidos presentes.

Após se obter a lista com os valores sem repetidos, iremos ordená-los de acordo com a sua avaliação utilizando o predicado ordena\_populacao/2 presente no algoritmo original e, a seguir, selecionamos os P melhores indivíduos da lista com o predicado melhores/2.

Uma imagem com texto, captura de ecrã, Tipo de letra

Descrição gerada automaticamente

Uma imagem com texto, Tipo de letra, captura de ecrã, file

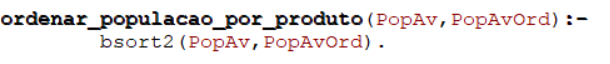
Descrição gerada automaticamente

Obtidos os melhores indivíduos, utilizaremos o predicado retira\_melhores/3 para passá-los para a geração seguinte, obtendo assim uma nova lista contendo T-P indivíduos onde iremos associar cada um deles o produto da avaliação por um número aleatório gerado entre 0 e 1 através do predicado liga\_produto\_avaliacao/2 e, por consequência, ordenamos essa lista por ordem crescente de acordo com o produto com o predicado ordenar\_populacao\_por\_produto/2.



Uma imagem com texto, Tipo de letra, captura de ecrã

Descrição gerada automaticamente



Por fim, os N-P primeiros elementos dessa lista são passados para a geração seguinte com o predicado melhores\_sobra/4; sendo que, para isso, primeiro temos de remover o valor do produto associado e ficar a avaliação. Para tal, usamos o predicado remover\_produtos/2.

A partir destas duas listas podemos juntar ambas numa única lista, ordená-la através dos predicados do SWI-Prolog e passar para a geração seguinte.

Uma imagem com texto, Tipo de letra, captura de ecrã, file

Descrição gerada automaticamente

Uma imagem com texto, Tipo de letra, captura de ecrã, file

Descrição gerada automaticamente

## Evitar que a sequência de cruzamentos seja sempre entre elementos em sequências já definidas 2 a 2

O algoritmo fornecido tenta cruzar sempre o 1º com o 2º, o 3º com o 4º e por aí a diante. Uma forma simples de colmatar esse problema é utilizar uma permutação aleatória entre os elementos da lista de indivíduos da população, antes de utilizar o que já está implementado. Para tal ser feito, irá ser usada o predicado do SWI Prolog random\_permutation/2.

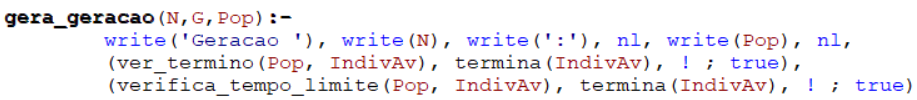
Uma imagem com texto, captura de ecrã, Tipo de letra

Descrição gerada automaticamente

## Condições de término

Para a condição de termino, o algoritmo genético fornecido é executado para o número de gerações que o utilizador introduziu como input. Para além dessa condição de termino, adicionamos também mais uma condição em que o algoritmo corre até atingir um determinado individuo, isto é, até atingir um valor igual ao estabelecido pelo utilizador. Além disso, caso não seja encontrado porque o valor indicado pelo utilizador é inferior á solução ideal, implementamos também uma condição de termino que estabelece um tempo limite de execução para evitar ciclos infinitos.

Para realizar o que foi acima descrito, é aplicado, no predicado gera\_geracao/3, os predicados ver\_termino/3 e o verifica\_tempo\_limite/2. No primeiro predicado é verificado se há algum individuo com uma avaliação igual ou inferior ao valor dado como input e no segundo predicado verifica se o tempo atual de execução excede o limite imposto. Esta condição de tempo limite é implementada através do predicado do SWI-Prolog get\_time/1. O predicado get\_time/1 obtêm o tempo em que foi inicializada a execução atual do algoritmo e depois verifica-se se o valor é maior ao valor introduzido para o limite.



Uma imagem com texto, captura de ecrã, Tipo de letra

Descrição gerada automaticamente

# Complexidade

Para realizar o estudo da Complexidade criamos um predicado que cria tarefas aleatórias. O número de tarefas aleatórias é igual ao input que o utilizador escreve.

Começamos, numa primeira fase do estudo de complexidade, por criar apenas quatro tarefas. Estas quatro tarefas, aleatórias, serviram depois para gerar a melhor sequencia entre elas para se compreender qual é o melhor caminho que o robô pode usar.

Vimos que, nestas quatro tarefas geradas aleatoriamente, os algoritmos não têm dificuldade em aplicar a melhor sequencia.

Portanto, agora, iremos testar para cinco tarefas.

Aqui, mais uma vez, não houve problema relativamente a execução do algoritmo. Nota-se, como é normal, um maior tempo de execução comparativamente ao exemplo com quatro tarefas. Este tempo de execução irá aumentar gradualmente conforme o crescimento do número de tarefas.

Ao fim de se tentar, para cada valor de tarefas, começa-se a notar, a partir das dez tarefas que o SWI-Prolog não consegue fazer a operação corretamente.   
 O erro descrito, na consola do SWI-Prolog é o seguinte:

Uma imagem com texto, Tipo de letra, captura de ecrã

Descrição gerada automaticamente

# Estado de Arte

A Inteligência Artificial (IA) é a capacidade de uma máquina de reproduzir tarefas semelhantes às humanas.

A IA permite que os sistemas técnicos percebam o ambiente que lhes rodeia, lidem com resolução de problemas, agindo no sentido de alcançar um objetivo específico. Para tal, o computador irá receber dados, irá processá-los e irá responder com a sua solução.

Os sistemas de IA são capazes de adaptar o seu comportamento, de forma autónoma, através de uma análise dos efeitos das ações anteriores.

O funcionamento da Inteligência Artificial se baseia na combinação de grandes volumes de dados digitais e algoritmos inteligentes. Respetivamente, permitem ao sistema fazer a leitura e interpretar padrões e informações para uma aprendizagem automática.

É importante destacar que, para que a aprendizagem aconteça, o sistema de AI tem de ser constantemente alimentado com novos dados.

Assim, o funcionamento da IA baseia-se em:

* Modelo de dados: estruturas utilizadas para processar, categorizar e analisar dados;
* Big Data: disponibilidade de grandes volumes de dados;
* Poder de processamento: capacidade operacional do sistema em processar as informações.

Para que esse processo aconteça é necessário combinar diferentes tecnologias que, juntas, possam conferir à máquina a capacidade de imitar o raciocínio lógico humano.

São elas:

# Machine Learning

Machine Learning trata-se de uma tecnologia que torna os sistemas capazes de aprender sozinhos e evoluir.

Tal situação é possível pelo processamento de dados e identificação de padrões, que possibilitam, a máquina, a tomada de decisões sem a necessidade de que o sistema seja programado para chegar a uma determinada conclusão.

Portanto, sem Machine Learning, a AI não seria possivel.

Um exemplo da utilização de Machine Learning é em sistemas recomendação personalizadas de serviços de Streaming.

As máquinas analisam os dados de consumo dos usuários, identifica padrões e sugere conteúdos ou produtos mais alinhados ao gosto de cada pessoa.

# Deep Learning

Outro alicerce da AI é o Deep Learning. O Deep Learning é um aprofundamento do Machine Learning. Porém, com capacidade de tornar a aprendizagem do sistema mais complexo e inteligente para que forneça resultados mais acertados.

O Deep Learning aplica redes neurais complexas, que seguem a mesma lógica da ligação entre os neurónios do cérebro humano. Assim, o sistema tem a capacidade de aprender padrões complexos e interpretar grandes quantidades de dados.

Uma das aplicações mais comuns do Deep Learning é no reconhecimento de imagens e fala, sendo utilizado para o desenvolvimento de veículos autónomos.

# Processamento de Linguagem Natural (PLN)

Por último, uma das vertentes das AI é a capacidade de as máquinas interagirem e comunicarem com os seres humanos de forma natural; essa habilidade só existe graças ao Processamento da Linguagem Natural.

O PLN consiste na utilização da tecnologia de Machine Learning para que o sistema analise, entenda, procure e encontre padrões e gere linguagem humana de modo natural, inclusive na forma de fala.

O Processamento de Linguagem Natural é utilizado em áreas como análise de sentimentos, algoritmos que identificam o teor de um texto e também nos chatbots de atendimento ao consumidor.

Um exemplo da aplicação de uma ferramenta de AI, neste caso o chatGPT, num trabalho académico é o seguinte:

Uma imagem com texto, captura de ecrã, Tipo de letra

Descrição gerada automaticamente

Uma imagem com texto, captura de ecrã, Tipo de letra, documento

Descrição gerada automaticamente

Uma imagem com texto, captura de ecrã, Tipo de letra, design

Descrição gerada automaticamente

# Pros e Contra

## Vantagens

* Automatização de tarefas
* Tomada de decisão mais precisa
* Personalização e customização
* Eficiência operacional
* Inovação
* Melhoria da qualidade de vida
* Maior segurança
* Análise de sentimentos e opiniões
* Economia de recursos
* Acesso a informações

## Desvantagem

* Dificuldade na compreensão de processos
* Dependência de dados
* Viés e preconceito
* Complexidade e custo
* Falta de controle humano
* Desafios regulatórios
* Falta de humanização

# Bibliografia

<https://www.europarl.europa.eu/news/pt/headlines/society/20200827STO85804/o-que-e-a-inteligencia-artificial-e-como-funciona>

<https://blog.neoway.com.br/inteligencia-artificial/>

<https://tecnoblog.net/responde/machine-learning-ia-o-que-e/>

https://blog-parceiros.ifood.com.br/vantagens-desvantagens-inteligencia-artificial/

https://www.iberdrola.com/inovacao/o-que-e-inteligencia-artificial