

Relatório ALGAV
Sprint C
3DJ G059

Feito por:

André Ferreira, 1190378

David Marques, 1221276

Diogo Cunha, 1221071

João Monteiro, 1221023

Tomás Peixoto, 1221948

Professor:

Luis Manuel Silva Conceição (MSC)

30/12/2024

Índice

1. Introdução.....	3
2. Aleatoriedade no cruzamento entre indivíduos da população no AG	4
3. Seleção da nova geração da população do AG	5
4. Parametrização da condição de término do AG	6
5. Adaptação do Algoritmo Genético	7
6. Consideração de vários blocos de operação.....	10
7. Estudo do Estado da Arte	13
8. Anexo – Escolha do algoritmo a usar.....	17
9. Conclusão	19
10. Referências	20

1. Introdução

Neste *sprint*, foram pedidos os seguintes aspetos:

- Mais uma forma de escalonar cirurgias numa sala, usando **Algoritmos Genéticos** (código base fornecido, ainda para ser adaptado para a cirurgias em salas em vez de tarefas em máquinas).
- Adicionar uma forma de decidir automaticamente quais cirurgias ficam em quais salas através dum algoritmo de distribuição.
- Fazer um estudo do Estado da Arte da Visão Computacional e Robótica.
- Adicionar uma forma de, através do número de cirurgias a escalonar e o tempo disponível, decidir automaticamente qual seria o melhor algoritmo a usar.

Neste relatório, a equipa vai explicar como é que estes pedidos foram cumpridos.

2. Aleatoriedade no cruzamento entre indivíduos da população no AG

O cruzamento é tentado em indivíduos aleatórios da população devido a função *shuffle*, onde esta mistura a população para depois o cruzamento acontecer entre indivíduos 2 a 2 da população misturada.

```
% Shuffle the list randomly
shuffle(List, Shuffled) :-
    random_permutation(List, Shuffled).
```

Figure 1. Método shuffle

```
crossover([ ],[ ]).
crossover([Ind*_],[Ind]).
crossover([Ind1*_ ,Ind2*_ |Rest],[NInd1,NInd2|Rest1]):-
    generate_crossover_points(P1,P2),
    probab_crossover(Pcruz),random(0.0,1.0,Pc),
    ((Pc =< Pcruz,! ,
        cross(Ind1,Ind2,P1,P2,NInd1),
        cross(Ind2,Ind1,P1,P2,NInd2))
    ;
    (NInd1=Ind1,NInd2=Ind2)),
    crossover(Rest,Rest1).
```

Figure 2. Crossover

3. Seleção da nova geração da população do AG

Na seleção da nova geração da população estamos a garantir que o pelo menos o melhor individuo da população atual passe para a próxima geração pois antes de acontecer a geração da nova população é guardado o melhor individuo e caso este não passe para a próxima geração, é removido o pior individuo da nova geração e adicionado o melhor da antiga.

```
% If Element is in the list, simply return the original list.
change_if_not_present(Element, List, List) :-
    member(Element, List), !.

% If Element is not in the list, remove the last element and add Element to the end.
change_if_not_present(Element, List, ReturnedList) :-
    remove_last(List, RemovedList),
    add_Best(Element, RemovedList, ReturnedList).

% Remove the last element of a list.
remove_last([], []). % Base case: Single-element list.
remove_last([H|T], [H|Result]) :- % Recursive case: Process the rest.
    remove_last(T, Result).

% Add an element to the end of a list.
add_Best(Element, [], [Element]). % Base case: Adding to an empty list.
add_Best(Element, [Head|Tail], [Head|NewTail]) :-
    add_Best(Element, Tail, NewTail). % Recursive case: Add to the rest.
```

Figure 3. Verificação do melhor individuo

4. Parametrização da condição de término do AG

O programa termina quando a função principal **generate** atingir a última geração configurada pelo número de gerações. Isso é controlado pela recursão na função **generate_generation**.

```
generate_generation(G,G,Pop,_,_):-!,
    write('Generation '), write(G), write(':'), nl, write(Pop), nl.
generate_generation(N,G,Pop,Room,Day):-
    write('Generation '), write(N), write(':'), nl, write(Pop), nl,
    Pop = [Best|_], % Extract the best element
    write('Best Element: '), write(Best), nl, % Write the first element
    crossover(Pop,NPop1),
    mutation(NPop1,NPop),
    evaluate_population(NPop,NPopValue,Room,Day),
    order_population(NPopValue,NPopOrd),
    change_if_not_present(Best, NPopOrd, NPopFinal),
    order_population(NPopFinal,NPopOrd2),
    N1 is N+1,
    generate_generation(N1,G,NPopOrd2,Room,Day).
```

Figure 3. generate_generation

5. Adaptação do Algoritmo Genético

Para adaptar o código fornecido às cirurgias em salas, tivemos de considerar as seguintes coisas de domínio:

1. Cada sala tem uma agenda.
2. Cada operação ocupa tempo.
3. Cada operação tem 3 fases.
4. Cada operação tem *staff*.
5. Cada *staff* tem a sua própria agenda.

Para usabilidade (integração com o resto do projeto), também tivemos de considerar outros aspetos:

1. Remover escritas à consola, porque estas interferem no *output* do **JSON** quando a *request* é chamada pelo exterior.
2. Enviar os parâmetros (número de gerações, tamanho da população, etc.) como argumentos, e não por leitura do teclado.
3. Ter a agenda da sala, as agendas do staff e o melhor tempo final como *outputs*.

Os 3 pontos anteriormente mencionados levam ao que vemos neste bloco de código inicial:

```
initialize(NG,PS,P1,P2):-
    (retract(generations(_));true), asserta(generations(NG)),
    (retract(population(_));true), asserta(population(PS)),
    PC is P1/100,
    (retract(prob_crossover(_));true), asserta(prob_crossover(PC)),
    PM is P2/100,
    (retract(prob_mutation(_));true), asserta(prob_mutation(PM)).

generate(Room,Day,NG,PS,P1,P2,AgendaRoom,AgendaDoctors,BestTime):-
    initialize(NG,PS,P1,P2),
    generate_population(Pop,Room),
    %write('Pop= '),write(Pop),nl,
    evaluate_population(Pop,PopValue,Room,Day,inf),
    %write('PopValue= '),write(PopValue),nl,
    order_population(PopValue,PopOrd),
    generations(NG),
    generate_generation(0,NG,PopOrd,Room,Day),
    retract(better_sol(Day,Room,AgendaRoom,AgendaDoctors,BestTime)).
```

Figure 4. Inicialização do **Algoritmo Genético**

No predicado **generate_population**, a maior modificação é a recebida da sala por argumento. Assim, podemos usar o facto dinâmico **surgeries_per_room** (explicado no ponto 6.) para ter a lista de cirurgias a escalonar e o tamanho da lista.

```
generate_population(Pop,Room):-
    population(PopSize),
    surgeries_per_room(Room,SurgeryList),
    length(SurgeryList,NumT),
    retractall(availability(_,_,_)),
    retractall(num_surgeries(_)),
    assertz(num_surgeries(NumT)),
    generate_population(PopSize,SurgeryList,NumT,Pop).
```

Figure 5. Criação da população

Enquanto avaliamos a população, verificamos o tempo final é melhor que o antigo melhor. Se for, guardamos esse valor, a agenda da sala e as agendas do *staff*.

```
evaluate_population([], [], _, _, _).
evaluate_population([Ind | Rest], [Ind*V | Rest1], Room, Day, BestV) :-
    evaluate(Ind, V, Room, Day),
    ( BestV = inf ->
        findall(Doctor,assignment_surgery(_,Doctor,_),LDoctors1),
        remove_equals(LDoctors1,LDoctors),
        list_doctors_agenda(Day,LDoctors,LDAgendas),
        agenda_operation_room1(Room, Day, Agenda),
        asserta(better_sol(Day,Room,Agenda,LDAgendas,V)),
        NewBestV = V
    ;
    ( V < BestV ->
        retract(better_sol(_,_,_,_,_)),
        findall(Doctor,assignment_surgery(_,Doctor,_),LDoctors1),
        remove_equals(LDoctors1,LDoctors),
        list_doctors_agenda(Day,LDoctors,LDAgendas),
        agenda_operation_room1(Room, Day, Agenda),
        asserta(better_sol(Day,Room,Agenda,LDAgendas,V)),
        NewBestV = V
    ;
    NewBestV = BestV
    ),
    evaluate_population(Rest, Rest1, Room, Day, NewBestV).
```

Figure 6. Avaliação da população

Na avaliação individual, temos de escalonar as várias cirurgias dessa lista na sala (processo igual ao desenvolvido no *Sprint* anterior) e avaliar o tempo final.

NOTA: Por mais que se use o método de avaliação fornecido, poderia ser aplicado, em vez deste, o método de avaliação implementado pela equipa no *Sprint* anterior.

```
evaluate(Seq,V,Room,Day):-
    retractall(agenda_staff1(_,_)),
    retractall(agenda_operation_room1(_,_)),
    retractall(availability(_,_)),
    findall(.,(agenda_staff(D,Day,Agenda),assertz(agenda_staff1(D,Day,Agenda))),_),
    agenda_operation_room(Room,Day,Agenda),assert(agenda_operation_room1(Room,Day,Agenda)),
    findall(.,(agenda_staff1(D,Day,L),free_agenda0(L,LFA),adapt_timetable(D,Day,LFA,LFA2),assertz(availability(D,Day,LFA2))),_),
    schedule_operations(Seq,Room,Day),
    agenda_operation_room1(Room,Day,NewAgenda),
    reverse(NewAgenda,AgendaR),
    evaluate_final_time(AgendaR,Seq,V).

schedule_operations([],_,_).
schedule_operations([Surgery|TOP],Room,Day):-
    surgery_id(Surgery,Type),
    surgery(Type, TA,TS,TC),
    Duration is TA+TS+TC,
    findall(Staff, assignment_surgery(Surgery, Staff, _), LStaff),
    intersect_all_agendas(LStaff, Day, LA),

    agenda_operation_room1(Room, Day, Agenda),
    free_agenda0(Agenda, FAgRoom),
    intersect_2_agendas(LA, FAgRoom, LIntADoctorsRoom),
    remove_unf_intervals(Duration, LIntADoctorsRoom, LAPossibilities),
    schedule_first_interval(Duration, LAPossibilities, (TStart, _)),

    TEndAnaesthesia is TStart + TA,
    TEndSurgery is TEndAnaesthesia + TS,
    TEndCleaning is TEndSurgery + TC,
    retract(agenda_operation_room1(Room, Day, Agenda)),
    insert_agenda((TStart, TEndAnaesthesia, Surgery), Agenda, Agenda1),
    insert_agenda((TEndAnaesthesia, TEndSurgery, Surgery), Agenda1, Agenda2),
    insert_agenda((TEndSurgery, TEndCleaning, Surgery), Agenda2, Agenda3),
    assertz(agenda_operation_room1(Room, Day, Agenda3)),

    findall(Staff, assignment_surgery(Surgery, Staff, 1), LASTaff),
    insert_agenda_doctors((TStart, TEndAnaesthesia, Surgery), Day, LASTaff),
    findall(Staff, assignment_surgery(Surgery, Staff, 2), LSStaff),
    insert_agenda_doctors((TEndAnaesthesia, TEndSurgery, Surgery), Day, LSStaff),
    findall(Staff, assignment_surgery(Surgery, Staff, 3), LCStaff),
    insert_agenda_doctors((TEndSurgery, TEndCleaning, Surgery), Day, LCStaff),
    schedule_operations(TOP,Room,Day).
```

Figure 7. Avaliação individual e agendamento

Aqui vemos uma execução do código para a sala **or1** no dia **28/10/2024**. (Nº de gerações = 4, tamanho da população = 4, chance de *crossover* = 60%, chance de mutação = 4%)

NOTA: As escritas à consola foram restauradas para esta *print*.

```
Pop=[{so100007,so100003,so100001,so100007},{so100007,so100003,so100001,so100007},{so100009,so100007,so100003,so100001},{so100009,so100003,so100007,so100001}]
PopValue=[{so100009,so100003,so100001,so100007}*1240,{so100007,so100003,so100001,so100007}*1210,{so100009,so100007,so100003,so100001}*1240,{so100009,so100003,so100007,so100001}*1240]
Generation 0:
[[{so100007,so100003,so100001,so100007}*1210,{so100009,so100003,so100001,so100007}*1240,{so100009,so100007,so100003,so100001}*1240,{so100009,so100003,so100007,so100001}*1240]
Best Element: {so100007,so100003,so100001,so100007}*1210
PopValue=[{so100009,so100003,so100001,so100007}*1240,{so100007,so100003,so100001,so100007}*1210,{so100009,so100007,so100003,so100001}*1240,{so100009,so100003,so100007,so100001}*1240]
Generation 1:
[[{so100007,so100003,so100001,so100007}*1210,{so100009,so100003,so100001,so100007}*1240,{so100009,so100007,so100003,so100001}*1240,{so100009,so100003,so100007,so100001}*1240]
Best Element: {so100007,so100003,so100001,so100007}*1210
PopValue=[{so100009,so100003,so100001,so100007}*1240,{so100007,so100003,so100001,so100007}*1210,{so100009,so100007,so100003,so100001}*1240,{so100009,so100003,so100007,so100001}*1240]
Generation 2:
[[{so100007,so100003,so100001,so100007}*1210,{so100009,so100003,so100001,so100007}*1240,{so100009,so100007,so100003,so100001}*1240,{so100009,so100003,so100007,so100001}*1240]
Best Element: {so100007,so100003,so100001,so100007}*1210
PopValue=[{so100009,so100003,so100001,so100007}*1240,{so100007,so100003,so100001,so100007}*1210,{so100009,so100007,so100003,so100001}*1240,{so100009,so100003,so100007,so100001}*1240]
Generation 3:
[[{so100007,so100003,so100001,so100007}*1210,{so100009,so100003,so100001,so100007}*1240,{so100009,so100007,so100003,so100001}*1240,{so100009,so100003,so100007,so100001}*1240]
Best Element: {so100007,so100003,so100001,so100007}*1210
PopValue=[{so100009,so100003,so100001,so100007}*1240,{so100007,so100003,so100001,so100007}*1210,{so100009,so100007,so100003,so100001}*1240,{so100009,so100003,so100007,so100001}*1240]
Generation 4:
[[{so100007,so100003,so100001,so100007}*1210,{so100009,so100003,so100001,so100007}*1240,{so100009,so100007,so100003,so100001}*1240,{so100009,so100003,so100007,so100001}*1240]
R = [[{320,365,so100003},{365,440,so100003},{440,485,so100003},{520,579,so1000000},{580,625,so100007},{625,715,so100007},{715,760,so100007},{791,...},{...,...}],...],
D = [[{d003,[[{365,440,so100003},{760,790,m01}]],{d005,[[{320,365,so100003},{720,850,m01}]]},{d004,[[{950,980,c04}]]},{n003,[[{320,365,so100003}]]},{m001,[[{440,485,so100003}]]},{896,...,...}],...],
{d001,[[{836,...,...},{...,...}],{n001,[[{...,...}],...},{d002,[[{...,...}],{...,...}],...}],...}],
R = 1210 .
```

Figure 8. Execução do algoritmo

6. Consideração de vários blocos de operação

Neste requisito, foi pedido que atribuíssemos (sem escalonar, só decidir) quais operações vão para quais salas.

Para isto, desenvolvemos um predicado ***associate_surgeries_to_rooms*** que recebe um dia como atributo e devolve uma lista de salas, cada uma com a sua lista de cirurgias.

NOTA: Assim como no ponto 5., as escritas à consola estão comentadas porque estas interferem com o *output* do **JSON** quando este é chamado através duma *request*.

Por mais que este método não faça o agendamento, é importante calculá-lo duma forma simples para saber se seria sequer possível colocar as cirurgias numa dada sala.

```
associate_surgeries_to_rooms(Day,SurgeriesPerRoom):-
    retractall(total_surgery_time(_,_)),
    retractall(agenda_operation_room1(_,_,_)),
    retractall(surgeries_per_room(_,_)),
    findall(, (agenda_operation_room(R,D,Agenda), assertz(agenda_operation_room1(R,D,Agenda))), _),
    findall(SurgeryID, surgery_id(SurgeryID, _), LSurgeries),
    getTotalTimes(LSurgeries),
    sort_surgeries_by_time(SortedSurgeries),
    findall(Room, agenda_operation_room1(Room, Day, _), LRooms),
    distribute_surgeries(SortedSurgeries, LRooms, Day),
    findall((R,S), surgeries_per_room(R, S), SurgeriesPerRoom).

getTotalTimes([]):- !.
getTotalTimes([Surgery|TSurgeries]):-
    surgery_id(Surgery, Type),
    surgery(Type, TimeA, TimeS, TimeC),
    Time is TimeA + TimeS + TimeC,
    assertz(total_surgery_time(Surgery, Time)),
    %write("Surgery "), write(Surgery), write(" has time="), write(Time), nl,
    getTotalTimes(TSurgeries).
```

Figure 9. Predicado principal

Para distribuir as cirurgias por salas adotamos a seguinte estratégia (Assumindo que temos 3 salas):

1. A 1ª cirurgia vai para a 1ª sala.
2. A 2ª cirurgia vai para a 2ª sala.
3. A 3ª cirurgia vai para a 3ª sala.
4. A 4ª cirurgia vai para a 1ª sala.
5. A 5ª cirurgia vai para a 2ª sala.
6. ...

Como dito anteriormente, verificamos se é possível uma dada lista de cirurgias ficar numa sala verificando a sua agenda.

Também verificamos que a sua taxa de ocupação é menor que 80%. Caso seja, o processo continua.

```
distribute_surgeries([], _, _) :- !.
distribute_surgeries([Surgery|RemainingSurgeries], Rooms, Day):-
    nth1(_, Rooms, Room),
    schedule_surgery(Surgery, Room, Day),
    rotate_list(Rooms, RotatedRooms),
    distribute_surgeries(RemainingSurgeries, RotatedRooms, Day).

schedule_surgery(Surgery, Room, Day):-
    total_surgery_time(Surgery, Duration),
    agenda_operation_room1(Room, Day, Agenda),
    free_agenda0(Agenda, FAgRoom),
    remove_unf_intervals(Duration, FAgRoom, LAPossibilities),
    schedule_first_interval(Duration, LAPossibilities, (TStart, _)),

    TEnd is TStart + Duration,
    retract(agenda_operation_room1(Room, Day, OldAgenda)),
    insert_agenda((TStart, TEnd, Surgery), OldAgenda, NewAgenda),
    total_room_occupation(NewAgenda, Occupation),
    RoomOccupationRate is Occupation / 1440,
    %write("Occupation Rate for room "), write(Room), write(" = "), write(RoomOccupationRate), nl,
    RoomOccupationRate < 0.8,
    assertz(agenda_operation_room1(Room, Day, NewAgenda)),

    ( surgeries_per_room(Room, Operations) ->
        retract(surgeries_per_room(Room, Operations))
    ;
        Operations = []
    ),
    assertz(surgeries_per_room(Room, [Surgery|Operations])).
    %write("List of surgeries for room "), write(Room), write(" = "), write([Surgery|Operations]), nl,
    %write("Scheduled "), write(Surgery), write(" (with duration=", write(Duration), write(") in room "), write(Room), nl,
    %write("Updated room "), write(Room), write("'s agenda: "), write(NewAgenda), nl.

total_room_occupation([], 0).
total_room_occupation([(Start, End, _)|T], Occupation):-
    total_room_occupation(T, TOccupation),
    Occupation is TOccupation + End - Start.
```

Figure 10. Distribuição das cirurgias e verificação da quota

Aqui vemos uma execução do programa (com as escritas ao ecrã restauradas).

Vemos que, sempre que uma cirurgia é colocada numa sala, é verificada a sua agenda e a sua taxa de ocupação. Desta forma, temos a certeza de tudo está dentro dos limites.

```
9 ?- associate_surgeries_to_rooms(20241028,L).
Occupation Rate for room or1 = 0.18611111111111112
List of surgeries for room or1 = [so100001]
Scheduled so100001 (with duration=150) in room or1
Updated room or1's agenda: [(0,150,so100001),(520,579,so100000),(1000,1059,so099999)]
Occupation Rate for room or2 = 0.17361111111111111
List of surgeries for room or2 = [so100004]
Scheduled so100004 (with duration=150) in room or2
Updated room or2's agenda: [(0,150,so100004),(700,800,so200000)]
Occupation Rate for room or3 = 0.13888888888888889
List of surgeries for room or3 = [so100006]
Scheduled so100006 (with duration=150) in room or3
Updated room or3's agenda: [(0,150,so100006),(750,800,so300000)]
Occupation Rate for room or4 = 0.24305555555555555
List of surgeries for room or4 = [so100008]
Scheduled so100008 (with duration=150) in room or4
Updated room or4's agenda: [(0,150,so100008),(800,900,so300000),(950,1050,so088888)]
Occupation Rate for room or1 = 0.29027777777777778
List of surgeries for room or1 = [so100009,so100001]
Scheduled so100009 (with duration=150) in room or1
Updated room or1's agenda: [(0,150,so100001),(151,301,so100009),(520,579,so100000),(1000,1059,so099999)]
Occupation Rate for room or2 = 0.27777777777777778
List of surgeries for room or2 = [so100010,so100004]
Scheduled so100010 (with duration=150) in room or2
Updated room or2's agenda: [(0,150,so100004),(151,301,so100010),(700,800,so200000)]
Occupation Rate for room or3 = 0.24305555555555555
List of surgeries for room or3 = [so100012,so100006]
Scheduled so100012 (with duration=150) in room or3
Updated room or3's agenda: [(0,150,so100006),(151,301,so100012),(750,800,so300000)]
Occupation Rate for room or4 = 0.34722222222222222
List of surgeries for room or4 = [so100013,so100008]
Scheduled so100013 (with duration=150) in room or4
Updated room or4's agenda: [(0,150,so100008),(151,301,so100013),(800,900,so300000),(950,1050,so088888)]
Occupation Rate for room or1 = 0.40406111111111111
List of surgeries for room or1 = [so100003,so100009,so100001]
Scheduled so100003 (with duration=165) in room or1
Updated room or1's agenda: [(0,150,so100001),(151,301,so100009),(302,467,so100003),(520,579,so100000),(1000,1059,so099999)]
Occupation Rate for room or2 = 0.39236111111111111
List of surgeries for room or2 = [so100005,so100010,so100004]
Scheduled so100005 (with duration=165) in room or2
Updated room or2's agenda: [(0,150,so100004),(151,301,so100010),(302,467,so100005),(700,800,so200000)]
Occupation Rate for room or3 = 0.35763888888888889
List of surgeries for room or3 = [so100011,so100012,so100006]
Scheduled so100011 (with duration=165) in room or3
Updated room or3's agenda: [(0,150,so100006),(151,301,so100012),(302,467,so100011),(750,800,so300000)]
Occupation Rate for room or4 = 0.47222222222222222
List of surgeries for room or4 = [so100002,so100013,so100008]
Scheduled so100002 (with duration=180) in room or4
Updated room or4's agenda: [(0,150,so100008),(151,301,so100013),(302,467,so100002),(800,900,so300000),(950,1050,so088888)]
Occupation Rate for room or1 = 0.52906111111111111
List of surgeries for room or1 = [so100007,so100003,so100009,so100001]
Scheduled so100007 (with duration=180) in room or1
Updated room or1's agenda: [(0,150,so100001),(151,301,so100009),(302,467,so100003),(520,579,so100000),(580,760,so100007),(1000,1059,so099999)]
L = [(or2,[so100005,so100010,so100004]),(or3,[so100011,so100012,so100006]),(or4,[so100002,so100013,so100008]),(or1,[so100007,so100003,so100009,so100001])] .
```

Figure 11. Execução do algoritmo

7. Estudo do Estado da Arte

Introdução

O avanço rápido das tecnologias trouxe marcou de forma significativa o ramo da saúde. A Robótica e Automação são a frente de uma revolução que tem alterado os princípios e os métodos da prática da medicina e cuidados dos pacientes. Uso de ferramentas como *Robots* e inteligência artificial (IA) permitem reduzir os custos operacionais, aumentar a precisão da cirurgia e aumenta o potencial para melhores resultados para o paciente uma vez que também é possível reduzir o número de procedimentos.

Visão Computacional (VC) é um tipo de inteligência artificial que usa vídeos ou sequencias de imagens para reconhecer conteúdo, sendo adotada em diversas indústrias nos últimos anos. Devido a implicações de privacidade, segurança e princípios éticos as suas aplicações estão severamente limitadas em aplicações relacionadas com a indústria dos Saúde. No entanto, as aplicações da VC têm potencial de melhorar vários fatores nos cuidados de saúde como a monitorização dos pacientes, eficiência de sistemas e a redução da carga de trabalho do staff.

Contextualização histórica

A introdução da robótica nos cuidados médicos é datada para os finais do século XX. A convergência do avanço dos *robots* e da tecnologia usados no setor industrial com o conhecimento medicinal forneceram a base para avanços significativos. Apesar da base destes avanços ter sido o setor industrial, a Medicina foi o que mais beneficiou. Com o avanço das tecnologias passou-se de *robots* rudimentares, usados para tarefas simples, para mecanismos com capacidade para realizar operações cirúrgicas remotamente.

Um dos principais pontos da história da introdução da robótica e automação nos cuidados médicos foi em 1954, quando o Dr. Harry Sherman desenvolveu uma mão mecânica que podia operar remotamente.

Na década de 60, com os avanços na telerobótica, associados ao setor industrial e à exploração espacial, deram a possibilidade de se começar a desenvolver *robots* para casos onde a presença humana fosse impossível. Apesar da Medicina não ser destes casos não serem do interesse da Medicina, foi este período que mais contribuiu para o avanço dos sistemas remotamente controlados.

A primeira geração de *robots* cirúrgicos foi um dos pontos de viragem no crescimento da robótica nos cuidados médicos. Os **da Vinci Surgical System** representam um destes avanços que definiu o standard de precisão dentro da sala de operações. Cirurgias podiam agora ser feitas com precisão sem precedentes, requerendo menos incisões e reduzindo os efeitos negativos nos pacientes e diminuindo o tempo de recuperação, ao controlar os braços robóticos. Estes sistemas foram os pioneiros de muitos dos projetos de robótica aplicada à Medicina atuais, e revolucionaram completamente o negócio dos cuidados médicos.

A Visão Computacional tem uma vasta história, e apesar dos seus primórdios serem datados nos anos 50 e 60, as fundações de VC atuais foram estabelecidas na

década de 80, marcados principalmente por algoritmos de fluxo ótico e detecção de arestas.

Nos anos 90, com os avanços tecnológicos de *machine learning* (ML) e técnicas de calculo estatístico, começaram a aparecer as primeiras aplicações computacionais capazes de reconhecer e interpretar imagens com padrões mais complexos. Com a chegada do novo milénio, as aplicações de CV mostram-se mais em domínios práticos, como a análise de imagens médicas ou o reconhecimento de faces.

Em 2012, o uso de *Convolutional Neural Networks* (CNN) mostrou avanços significativos na área de CV, ao demonstrar uma excelente performance no ImageNet Large Scale Visual Recognition Competition, afirmando-se como o método mais eficaz de aprendizagem para CV. Estas CNN's mostram excelente capacidade na classificação de imagens, detecção de objetos e separação semântica transversal a várias áreas, como medicina, vigilância ou condução autónoma.

Presente

O ramo da Saúde tem experienciado em primeira mão o desenvolvimento de novas tecnologias, como AI, IoT e robótica. Existe uma grande variedade de dispositivos com diversas funcionalidades, desde apanhar e analisar dados, a motores e sensores em próteses. Para se manterem atualizados os centros hospitalares e de cuidados médicos cada vez mais adotam a robotização e automação dos processos.

A robótica, em específico, ajuda os estabelecimentos de saúde a construir uma especialização, melhorando aspetos com a precisão, a velocidade dos procedimentos, e a redução dos tempos de recuperação. O grande objetivo da robótica acaba por ser o desenvolvimento dos profissionais de saúde, uma vez que para serem capazes de extrair o máximo destes sistemas, têm de ser proficientes em tecnologias, aprendizes rápidos, flexíveis e abertos a novos problemas e desafios. Atualmente, a maioria dos casos de uso da robótica na medicina são aplicações de cirurgia ou então de monitorização remota dos pacientes.

As aplicações atuais de CV focam-se principalmente em monitorização e, em grande maioria, estão ou em fases de desenvolvimento ou de testes. Os desenvolvimentos recentes nas áreas de ML e CNN contribuíram para um grande avanço em CV e no reconhecimento e interpretação de imagens, ao ponto de tarefas complexas, como a classificação e segmentação de imagens médicas e o reconhecimento de objetos de interesse, são tarefas de cada vez menos desafiantes.

A classificação de imagens por CV é usada amplamente para identificar doenças através de imagens médicas. Os métodos baseados em *Deep Learning* (DL) com CNN's têm sido a abordagem de escolha, uma vez que superam os métodos tradicionais com relativa facilidade. Para além de ser possível analisar vários tipos de imagens médicas, existem vários projetos com o objetivo de detetar problemas de saúde como febre em espaços públicos.

A detecção de objetos por CV é usada para localizar e identificar anomalias com lesões, nódulos ou tumores. Métodos *one-step*, como YOLO e SSD, oferecem eficiência computacional, enquanto métodos *two-step*, como Mask-RCNN, apresentam maior precisão ao combinar propostas de regiões com classificação detalhada. Recentemente, com o aparecimento de métodos *anchor-free*, houve uma redução na complexidade das

deteções, no entanto, necessidade de grandes volumes de imagens médicas totalmente anotadas ainda representa um desafio significativo.

A segmentação de imagens é fundamental para identificar regiões anatómicas específicas em imagens médicas. O uso de redes adversárias generativas (GAN) têm sido um grande contributo para a segmentação de órgãos como o coração, os pulmões ou o cérebro. Por exemplo, para segmentação de camaras cardíacas, combinações de modelos baseados em aprendizagem por transferência e otimização de enxame de partículas têm proporcionado resultados mais precisos, ajudando no diagnóstico de doenças como hipertensão.

Desafios e limitações

Embora a integração de robótica e de VC na Saúde tem impulsionado avanços significativos, têm sido enfrentados desafios que dificultam a sua expansão pelo setor e o seu desenvolvimento.

O desenvolvimento da capacidade e funções dos *robots* requer uma avaliação cuidadosa da sustentabilidade dos resultados esperados durante as fases de design e desenvolvimento, incluindo preocupações relacionadas com segurança e privacidade devido ao acréscimo de processamento de documentação médica necessária. No campo dos serviços médicos e terapêuticos, os padrões regulamentares do setor de saúde, como privacidade, interação médico-paciente e uso eficiente de recursos, precisam ser expandidos para abranger os requerimentos tecnológicos e a análise preditiva proporcionada pela integração de robótica e inteligência artificial (IA).

A convergência de tecnologias robóticas enfrenta problemas sérios, como o aumento da população idosa, falta de profissionais de saúde, tratamentos diferentes para cada pessoa, uso errado de medicação, aumento de doenças comportamentais e atrasos nos diagnósticos. Além disso, há um aumento significativo de doenças crônicas e não modificáveis, particularmente em grupos minoritários. Tecnologias relacionadas com *Big Data*, nano-materiais, IoT e ferramentas de IA têm sido um auxílio a lidar com estes problemas, potenciando maior eficácia e eficiência nos cuidados de saúde. Para maximizar estes benefícios, é essencial que a robótica e a IA sejam integradas de forma contínua às infraestruturas existentes, garantindo maior precisão, proteção em tempo real e rastreamento da relação custo-benefício ao longo do tempo vida útil das tecnologias médicas.

Futuro e Objetivos

A investigação atual destes tópicos centra-se em expandir a sua potência e funcionalidade, manter a confidencialidade ao fornecer serviços médicos terapêuticos, e convergir tecnologias emergentes com as funcionalidades e aplicações já existentes.

A área da robótica tem continuado a evoluir assumindo papéis fundamentais dentro dos estabelecimentos médicos. A possibilidade de acelerar otimizar processos tem impacto significativo em termos operacionais e administrativos nestes espaços, sendo um dos principais fatores que motivam o desenvolvimento. Muito do trabalho realizado procura resolver questões como a qualidade dos tratamentos, redução da carga de trabalho dos profissionais de saúde, custos de funcionamento e aquisição e fornecimento de plataformas de treino e melhoria de habilidades dos profissionais.

Devido às tecnologias emergentes, o futuro da VC na área da saúde aparente ser bastante promissor, por exemplo, novos métodos DL capazes de aprender com *data sets*, ou GAN's para gerar dados diversos e reduzir o desequilíbrio de classes. A automação inteligente no processo de anotação e rotulagem poderá acelerar o desenvolvimento de modelos robustos, enquanto a integração de novos algoritmos compreensíveis e transparentes ajudará a construir confiança entre os profissionais de saúde, os pacientes e as tecnologias em causa. A convergência do conhecimento dos especialistas com as tecnologias de IA será crucial para criar soluções práticas, precisas e explicáveis, permitindo uma amplificação do alcance do uso de CV em diagnósticos, cirurgias e tratamentos. A VC tem o potencial de democratizar o acesso a cuidados médicos de alta qualidade, promovendo tratamentos mais eficazes, diagnósticos mais antecipados, melhorando globalmente o resultado da operação dos profissionais. A combinação da inovação tecnológica e da colaboração interdisciplinar será o principal agente na construção de um futuro em que a VC é um auxiliar indispensável das equipas médicas.

Resumidamente, procura-se construir um sistema de saúde centrado no paciente, eficiente, construído com base em novas tecnologias através de pesquisa inovadora e colaborativa e uma abordagem multidisciplinar. Enquanto se trabalham os problemas atuais é necessário preparar a implementação de tecnologias emergentes e uso ético é transparente das mesmas, como IA, aprendizagem federada, computação quântica e robótica avançada.

Conclusão

Apesar de ainda haver alguma preocupação sobre o declínio da participação humana nos cuidados dos pacientes, com o avanço da tecnologia, é cada vez mais obvio que não é possível substituir a interação humana com a robotização e inteligência artificial.

A implementação de novas tecnologias tem um grande potencial com benefícios para ambos os lados, como melhoria dos resultados de cirurgias, melhor gestão dos pacientes, maior segurança da medicação, no entanto ainda existe um longo caminho a ser percorrido até que possa resolver todos os problemas, sendo que a pesquisa da interação Humano-Robot tem sido cada vez mais crucial. Embora os desenvolvimentos recentes nesta área mostrem um futuro promissor nos ambientes variados dos cuidados médicos, é preciso notar que os desafios e problemas morais que vem de arrasto, sendo necessário garantir a infraestrutura legal, a segurança dos dados e as elevadas despesas de entrada nesta área.

É preciso alcançar um equilíbrio entre o potencial transformativo destas tecnologias e ferramentas e assegurar a segurança dos envolvidos á medida que se ultrapassam barreiras de desenvolvimento.

8. Anexo – Escolha do algoritmo a usar

Neste requisito, é pedido para o sistema conseguir escolher o método de agendamento de cirurgias (gerar todas e seleccionar a melhor, heurística ou algoritmo genético) mais apropriado analisando a dimensão do problema.

Para isto, foi criado o seguinte predicado:

```
calculate_f(X, F) :-
    F is (2.29 * 10**(-7) * exp(2 * (X+3))).
    % Function that represents the obtain best equation. (X = nº of surgeries, Y = nº of seconds needed)

what_method_to_use(Room, Time, Method) :-
    surgeries_per_room(Room, Surgeries),
    length(Surgeries, X),
    calculate_f(X, F),
    (Time > F ->
        Method = obtain_better_sol % Use the "obtain_better_sol" predicate
    ; Time < 0.2 * X ->
        Method = heuristic % Must use heuristic, since user has little time
    ;
        P is ceil(Time * 10) // X,
        G is P // 2,
        format(atom(Method), 'generate (Population = ~w, Generations = ~w)', [P, G])
    ).
```

Figure 12. Predicado principal

Primeiro, o predicado vê quantas cirurgias é que a sala em específico vai ter de escalonar (usando o facto dinâmico criado no requisito 7.3.1.

A partir daí, vemos qual opção será a melhor:

- Se o tempo (y) for maior que $f(x) = 2.29 * 10^{-7} * e^{2*(x+3)}$, (x sendo o nº de cirurgias), então podemos usar o “**obtain_better_sol**”.
 - Esta equação foi elaborada através dos resultados do estudo do requisito 6.3.2. Os valores de y mais à frente são maiores que os estudados, mas isto acrescente um nível de segurança na sua execução (não há o risco de sugerir o **obtain_better_sol** mas a execução do mesmo demorar mais que o tempo fornecido pelo utilizador)
- Se o tempo for menor que $0.2 * (\text{nº de cirurgias})$, é recomendada a utilização da heurística.

- Caso contrário, é recomendada a utilização do algoritmo genético em que o tamanho da população e o número de gerações são maiores conforme o **tempo**.

NOTA: Os seguintes *prints* usam uma versão do método que, em vez de receber o nome da sala, recebe diretamente o nº de cirurgias a escalonar, para questão de facilitar a demonstração.

```
61 ?- what_method_to_use(4,0.5,M).  
M = obtain_better_sol.
```

```
69 ?- what_method_to_use(6,6,M).  
M = 'generate (Population = 10, Generations = 5)'.  
  
70 ?- what_method_to_use(4,0.26,M).  
M = heuristic.  
  
71 ?- what_method_to_use(5,1,M).  
M = 'generate (Population = 2, Generations = 1)'.
```

9. Conclusão

Este relatório examina os aspetos cruciais do projeto, como, mas não limitado por:

- Adaptação do código base para cumprir com o que foi pedido.
- Adaptação do código base para cirurgias em salas.

Este estudo forçou-nos a estudar também os métodos de agendamento criados no *sprint* anterior para podermos efetivamente decidir qual método deve ser escolhido de acordo com o número de cirurgias a escalonar e o tempo disponível.

Desta forma, acreditamos que temos um sistema com várias formas de fazer o que um utilizador quer, contendo também sistemas para sugerir ao utilizador qual método usar de acordo com a situação atual.

10. Referências

- [1] Elyan E, Vuttipittayamongkol P, Johnston P, Martin K, McPherson K, Moreno-García CF, Jayne C, Sarker MMK. Computer vision and machine learning for medical image analysis: recent advances, challenges, and way forward. *Art Int Surg* 2022;2:xx.<http://dx.doi.org/10.20517/ais.2021.15>
- [2] Lindroth, H.; Nalaie, K.; Raghu, R.; Ayala, I.N.; Busch, C.; Bhattacharyya, A.; Moreno Franco, P.; Diedrich, D.A.; Pickering, B.W.; Herasevich, V. Applied Artificial Intelligence in Healthcare: A Review of Computer Vision Technology Application in Hospital Settings. *J. Imaging* 2024, 10, 81. <https://doi.org/10.3390/jimaging10040081>
- [3] Y. Fu, G. Niu, B. Pan, and K. Wang, "Design and Optimization of Remote Center Motion Mechanism of Minimally Invasive Surgical Robotics," presented at the Wang, IEEE, Dec. 2013.
- [4] A. Ingole and S. Sonwane, "Robotics And Automation in Healthcare," 2024 International Conference on Healthcare Innovations, Software and Engineering Technologies (HISSET), Karad, India, 2024, pp. 78-81, doi: 10.1109/HISSET61796.2024.00036.
- [5] Q. T. Do and J. Chaudri, "Creating Computer Vision Models for Respiratory Status Detection," 2022 44th Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC), Glasgow, Scotland, United Kingdom, 2022, pp. 1350-1353, doi: 10.1109/EMBC48229.2022.9871978.
- [6] R. Saini, "Research on the Recent Trends and Challenges on the Adoption of Robotics in the Healthcare Sector," 2023 2nd International Conference on Edge Computing and Applications (ICECAA), Namakkal, India, 2023, pp. 1140-1144, doi: 10.1109/ICECAA58104.2023.10212123.
- [7] M. Alseddiqi, B. AlMannaei, O. Najam and A. Al-Mofleh, "The Importance of Medical Robots in Improving Healthcare Services," 2022 International Conference on Automation, Computing and Renewable Systems (ICACRS), Pudukkottai, India, 2022, pp. 1397-1401, doi: 10.1109/ICACRS55517.2022.10029105
- [8] L. R. Kennedy-Metz et al., "Computer Vision in the Operating Room: Opportunities and Caveats," in *IEEE Transactions on Medical Robotics and Bionics*, vol. 3, no. 1, pp. 2-10, Feb. 2021, doi: 10.1109/TMRB.2020.3040002.
- [9] Charleen, C. Angelica, H. Purnama and F. Purnomo, "Impact of Computer Vision With Deep Learning Approach in Medical Imaging Diagnosis," 2021 1st International Conference on Computer Science and Artificial Intelligence (ICCSAI), Jakarta, Indonesia, 2021, pp. 37-41, doi: 10.1109/ICCSAI53272.2021.9609708.
- [10] B. M and M. R, "Urban Object Detection in UAV Imagery for Healthcare Applications: Leveraging the Internet of Things (IoT) in Smart Cities for Enhanced Urban Healthcare Monitoring and Management," 2023 International Conference on Sustainable Emerging Innovations in Engineering and Technology (ICSEIET), Ghaziabad, India, 2023 [6] R. Saini, "Research on the Recent Tre, pp. 607-612, doi: 10.1109/ICSEIET58677.2023.10303381.
- [11] N. K. Rajpoot, P. D. Singh, B. Pant and V.2. Tripathi, "The Future of Healthcare: A Machine Learning Revolution," 2023 International Conference on Artificial Intelligence for Innovations in Healthcare Industries (ICAIIHI), Raipur, India, 2023, pp. 1-6, doi: 10.1109/ICAIIHI57871.2023.10489320.
- [12] F. Mehmood, S. Ahmad and T. K. Whangbo, "Object detection based on deep learning techniques in resource-constrained environment for healthcare industry," 2022 International Conference on Electronics, Information, and Communication (ICEIC), Jeju, Korea, Republic of, 2022, pp. 1-5, doi: 10.1109/ICEIC54506.2022.974882