

ALGAV
2024/2025

Relatório SPRINT 3

Vasco Sousa (1221700),
João Pereira (1211503),
Mariana Correia (1211883).

Índice

Introdução.....	3
User Story 7.3.1	4
User Story 7.3.2	8
Conclusão.....	11
Estado de Arte	12
Minimamente invasiva (MIS).....	13
Peri operatório	13
Conclusão.....	15
Referências.....	16

Introdução

Este relatório documenta a implementação completa do projeto que visa otimizar a gestão de operações em salas de trabalho através de métodos automáticos e algoritmos genéticos. Através de um planeamento estruturado e alinhado com os objetivos estabelecidos, foram desenvolvidas e implementadas soluções inovadoras que atendem às exigências de eficiência e flexibilidade nos processos operacionais.

O foco do projeto inclui a atribuição automática de operações às salas de operação, a criação de agendas otimizadas utilizando algoritmos genéticos, e a integração de métodos adaptativos que garantem um balanceamento adequado entre precisão e aleatoriedade no cruzamento de indivíduos. Adicionalmente, foi assegurado que os melhores indivíduos de cada geração são preservados para populações subsequentes, sem recorrer a abordagens elitistas.

As principais etapas do desenvolvimento consistiram na parametrização de métodos de paragem, no estudo e aplicação de algoritmos genéticos ao contexto específico das salas de operações, e na realização de uma revisão do estado da arte para fundamentar as decisões metodológicas. Cada componente foi avaliado e ajustado com o objetivo de alcançar o melhor desempenho possível, respeitando as condições definidas, como o número de genes e o tempo de solução desejados.

Por fim, este relatório apresenta as conclusões obtidas, a avaliação do desempenho das soluções implementadas, e uma síntese do conhecimento adquirido durante o projeto, destacando os avanços realizados e as oportunidades para trabalhos futuros.

User Story 7.3.1

Objetivo Geral

Implementar um método automático para atribuir operações a salas de cirurgia, otimizando o agendamento com base na disponibilidade do staff, das salas, e dos requisitos das cirurgias. Este código aborda dois componentes principais:

1. Agendamento Automático de Cirurgias
2. Impressão do Cronograma Final

Componentes do Código

O método *schedule_surgeries* implementa o processo de agendamento automático das cirurgias para um dia específico.

Alguns dos passos principais são:

- **Atualizar as variáveis dinâmicas:** Remove informações antigas de agendamentos, como agendas de staff e salas.
 - **Objetivo:** Garantir que o novo agendamento começa em estado limpo.
- **Reiniciar agendas:** Recarrega as agendas do staff e das salas para o dia especificado.
 - **Função envolvidas:**
 - *agenda_staff1* para carregar agendas do staff.
 - *agenda_operation_room1* para carregar agendas das salas.
- **Preparar disponibilidade do staff:** Converte as agendas do staff em listas de intervalos de tempo livres.
 - **Objetivo:** Garantir que apenas os horários disponíveis são considerados no agendamento.
- **Obter Salas e Cirurgias:** Cria listas de salas disponíveis e cirurgias a serem agendadas.
- **Distribuição Round-Robin:** As cirurgias são distribuídas entre as salas de forma equilibrada.
 - **Funcionamento:** Cada sala recebe uma cirurgia por vez, de forma cíclica.
 - **Funções utilizadas:**
 - *distribute_surgeries_to_rooms*: Garante que cada cirurgia seja atribuída a uma sala.
 - *schedule_single_surgery*: Agenda uma cirurgia específica numa sala.
 - **Resultados esperados:** Após a execução, cada sala terá uma lista organizada de cirurgias agendadas para o dia.

```

% Schedule surgeries for a given day -- example usage: schedule_surgeries(20241028).
schedule_surgeries(Day) :-
    write('--- Starting scheduling for the day: '), write(Day), write(' ---'), nl, flush_output,

    write('Clearing existing dynamic facts...'), nl, flush_output,
    retractall(agenda_staff1(_,_)),
    retractall(agenda_operation_room1(_,_)),
    retractall(availability(_,_)),
    write('Dynamic facts cleared.'), nl, flush_output,

    write('Reinitializing staff and room agendas...'), nl, flush_output,
    findall(_, (agenda_staff(D,Day,Agenda), write(' Found staff: '), write(D), write(' - Agenda: '), write(Agenda), nl, flush_output, assertz(agenda_staff1(D,Day,
Agenda)))), _,
    findall(_, (agenda_operation_room(Room,Day,Agenda), write(' Found room: '), write(Room), write(' - Agenda: '), write(Agenda), nl, flush_output, assertz
(agenda_operation_room1(Room,Day,Agenda)))), _,
    write('Agendas reinitialized.'), nl, flush_output,

    write('Preparing staff availability...'), nl, flush_output,
    findall(_, (agenda_staff1(D,Day,L), write(' Processing staff: '), write(D), write(' - Agenda: '), write(L), nl, flush_output, free_agenda0(L,LFA), write('
Initial free agenda: '), write(LFA), nl, flush_output, adapt_timetable(D,Day,LFA,LFA2), write(' Adapted agenda: '), write(LFA2), nl, flush_output, assertz
(availability(D,Day,LFA2)))), _,
    write('Availability prepared.'), nl, flush_output,

    write('Retrieving all rooms and surgeries...'), nl, flush_output,
    findall(Room, agenda_operation_room(Room, Day, _), LRooms),
    write(' Available rooms: '), write(LRooms), nl, flush_output,
    findall(OpCode, surgery_id(OpCode, _), LSurgeries),
    write(' Available surgeries: '), write(LSurgeries), nl, flush_output,

    write('Starting round-robin distribution...'), nl, flush_output,
    distribute_surgeries_to_rooms(LSurgeries, LRooms, Day),
    write('Round-robin distribution completed.'), nl, flush_output.

```

Figura 1 - Método `schedule_surgeries(Day)` - Parte 1

```

% Assign surgeries to rooms
distribute_surgeries_to_rooms([], _, _).
distribute_surgeries_to_rooms(LSurgeries, LRooms, Day) :-
    distribute_surgeries_round_robin(LSurgeries, LRooms, Day, 0).

% Round-robin distribution of surgeries
distribute_surgeries_round_robin([], _, _, _).
distribute_surgeries_round_robin([OpCode|RestSurgeries], LRooms, Day, RoomIndex) :-
    length(LRooms, NumRooms),
    CurrentRoomIndex is RoomIndex mod NumRooms,
    nth0(CurrentRoomIndex, LRooms, CurrentRoom),
    (schedule_single_surgery(OpCode, CurrentRoom, Day) -> true ; true),
    distribute_surgeries_round_robin(RestSurgeries, LRooms, Day, RoomIndex + 1).

% Schedule a single surgery in a room
schedule_single_surgery(OpCode, Room, Day) :-
    surgery_id(OpCode, OpType),
    surgery(OpType, _, TSurgery, _),
    findall(Doctor, assignment_surgery(OpCode, Doctor), LDoctors),
    intersect_all_agendas(LDoctors, Day, LA),
    agenda_operation_room1(Room, Day, LAgenda),
    free_agenda0(LAgenda, LFAgRoom),
    intersect_2_agendas(LA, LFAgRoom, LIntAgDoctorsRoom),
    remove_unf_intervals(TSurgery, LIntAgDoctorsRoom, LPossibilities),
    LPossibilities \= [],
    schedule_first_interval(TSurgery, LPossibilities, (TinS, TfinS)),
    retract(agenda_operation_room1(Room, Day, Agenda)),
    insert_agenda((TinS, TfinS, OpCode), Agenda, Agenda1),
    assertz(agenda_operation_room1(Room, Day, Agenda1)),
    insert_agenda_doctors((TinS, TfinS, OpCode), Day, LDoctors).

```

Figura 2 - Método `schedule_surgeries(Day)` - Parte 2

O método *print_schedule* imprime o cronograma das cirurgias agendadas para cada sala num determinado dia.

Alguns dos passos principais são:

- **Recolher salas:** Identifica todas as salas com cirurgias agendadas.
- **Imprimir Cronograma:** Para cada sala, imprime as cirurgias com horários de início e fim.
 - **Função *print_schedule_list*:** Lê cada item da agenda da sala e imprime os detalhes no formato:
Surgery OpCode from Start to End
 - **Resultados esperados:** Um cronograma detalhado, mostrando para cada sala todas as cirurgias agendadas com os respectivos horários.

```
% After scheduling using the round robin, print the final schedule -- example usage: print_schedule(20241028).
print_schedule(Day) :-
    write('--- Schedule for'), write(Day), write(' ---'), nl, flush_output,
    findall(Room, agenda_operation_room(Room, Day, _), LRooms),
    print_rooms_schedule(LRooms, Day).

% Base case for rooms schedule
print_rooms_schedule([], _) :-
    write('--- End of schedule ---'), nl, flush_output.

% Recursive case for rooms schedule
print_rooms_schedule([Room|LRooms], Day) :-
    agenda_operation_room1(Room, Day, Schedule),
    write('Room '), write(Room), write(' Schedule: '), nl, flush_output,
    print_schedule_list(Schedule),
    print_rooms_schedule(LRooms, Day).

% Base case for individual schedules
print_schedule_list([]) :-
    write('There are no more surgeries scheduled.'), nl, flush_output.

% Recursive case for individual schedules
print_schedule_list([(Start, End, OpCode)|Rest]) :-
    write(' Surgery '), write(OpCode), write(' from '), write(Start), write(' to '), write(End), nl, flush_output,
    print_schedule_list(Rest).
```

Figura 3 - Método *print_schedule(Day)*

Conclusão

O desenvolvimento da **User Story 7.3.1** resultou na implementação de um sistema eficiente e automatizado para o agendamento de operações em salas de cirurgia, otimizando o uso de recursos e assegurando a organização do cronograma de forma clara e acessível.

O método *schedule_surgeries* demonstrou-se fundamental para o agendamento automático, garantindo a limpeza de dados antigos e a preparação de novos horários com base na disponibilidade do staff e das salas. A abordagem Round-Robin para distribuir cirurgias promoveu equilíbrio na utilização das salas, enquanto funções como *schedule_single_surgery* asseguraram precisão na atribuição de horários.

Por outro lado, o método *print_schedule* complementa este processo, permitindo uma visualização clara e detalhada do cronograma. Através de listas formatadas de cirurgias por sala, oferece uma ferramenta valiosa para comunicação e planeamento operacional.

Em suma, a solução implementada cumpre os objetivos traçados: otimiza o agendamento de cirurgias com base na disponibilidade e nos requisitos, proporcionando uma visão clara do cronograma e melhorando significativamente a gestão das operações nas salas de cirurgia.

User Story 7.3.2

Objetivo Geral

Este módulo visa implementar um sistema para agendar várias operações em salas de cirurgia usando algoritmos genéticos. O objetivo principal é criar um agendamento eficiente, atendendo a diferentes condições e cenários através de técnicas de evolução.

O trabalho foi dividido em quatro tarefas principais:

- **Tarefa 1:** Criação da estrutura básica do algoritmo genético para agendamento.
- **Tarefa 2:** Parametrização do algoritmo para atender diferentes condições.
- **Tarefa 3:** Introdução de aleatoriedade no cruzamento entre indivíduos.
- **Tarefa 4:** Preservação do melhor indivíduo na próxima geração, sem elitismo.

Tarefa 1: Estrutura Básica do Algoritmo Genético

A primeira tarefa consistia em criar o núcleo do algoritmo genético. Este algoritmo é baseado em geração, avaliação, cruzamento e mutação de uma população inicial de soluções.

Implementação

- **Representação dos Indivíduos:** Cada indivíduo é representado como uma lista de cirurgias agendadas, onde a posição indica a ordem de execução.
- **Geração da População Inicial:** A função *generate_population* cria uma população inicial de soluções aleatórias, garantindo que cada indivíduo é único.

```
generate_population(Pop) :-  
    population(PopSize),  
    surgeries(NumS),  
    findall(Surgery, surgery(Surgery, _, _, _), SurgeriesList),  
    generate_population(PopSize, SurgeriesList, NumS, Pop).  
  
generate_population(0, _, _, []) :- !.  
generate_population(PopSize, SurgeriesList, NumS, [Ind | Rest]) :-  
    PopSize1 is PopSize - 1,  
    generate_population(PopSize1, SurgeriesList, NumS, Rest),  
    generate_individual(SurgeriesList, NumS, Ind),  
    not(member(Ind, Rest)).  
generate_population(PopSize, SurgeriesList, NumS, L) :-  
    generate_population(PopSize, SurgeriesList, NumS, L).
```

Figura 4 - Método *generate_population(Pop)*

- **Avaliação de Indivíduos:** A função *evaluate* calcula a qualidade de cada solução com base no tempo de conclusão das cirurgias, aplicando penalidades caso ultrapassem o tempo permitido.

```
evaluate(Seq, V) :- evaluate(Seq, 0, V).

evaluate([], _, 0).
evaluate([S | Rest], Inst, V) :-
    surgery(S, Dur, Due, Pen),
    FinInst is Inst + Dur,
    evaluate(Rest, FinInst, VRest),
    ((FinInst <= Due, !, VT is 0) ; (VT is (FinInst - Due) * Pen)),
    V is VT + VRest.
```

Figura 5 - Método *evaluate(Seq, V)*.

- **Evolução da População:** O algoritmo gera várias gerações, realizando cruzamento e mutação para melhorar a qualidade das soluções ao longo do tempo.

Tarefa 2: Parametrização

Nesta etapa, o objetivo foi tornar o algoritmo mais flexível e ajustável a diferentes cenários, permitindo configurar parâmetros como:

- Tamanho da população.
- Taxa de mutação.
- Número de gerações.

Implementação

- **Função de Inicialização:** A função *initialize* permite configurar os parâmetros do algoritmo antes de sua execução.

```
initialize :-
    write('Number of new generations: '), read(NG),
    (retract(generations(_)); true), asserta(generations(NG)),
    write('Population size: '), read(PS),
    (retract(population(_)); true), asserta(population(PS)),
    write('Probability of crossover (%):'), read(P1),
    PC is P1 / 100,
    (retract(prob_crossover(_)); true), asserta(prob_crossover(PC)),
    write('Probability of mutation (%):'), read(P2),
    PM is P2 / 100,
    (retract(prob_mutation(_)); true), asserta(prob_mutation(PM)).
```

Figura 6 - Método *initialize*.

- **Adaptação para diferentes Cenários:** O algoritmo pode ser ajustado de acordo com as condições do problema, como maior ou menor taxa de mutação.

Tarefa 3: Aleatoriedade no Cruzamento

A introdução de aleatoriedade no cruzamento tem como objetivo aumentar a diversidade genética e evitar a convergência prematura.

Implementação

- **Cruzamento com Componentes Aleatórios:** A função *crossover* usa pontos de corte aleatórios para gerar novos indivíduos.

```
crossover([], []).
crossover([Ind * _], [Ind]).
crossover([Ind1 * _ | Rest], [NInd1, NInd2 | Rest1]) :-
    generate_crossover_points(P1, P2),
    prob_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).
```

Figura 7 - Método *crossover([],[])*.

- **Geração de Pontos de Cruzamento:** A função *generate_crossover_points* escolhe aleatoriamente os pontos onde os indivíduos serão divididos e recombinaos.

Tarefa 4: Preservação do Melhor Indivíduo

Nesta etapa, o desafio era garantir que o melhor indivíduo de uma geração fosse mantido na próxima, sem comprometer a diversidade.

Implementação

- **Inclusão do Melhor Indivíduo:** A função *include_best* insere o melhor indivíduo na próxima geração, removendo o pior para preservar a diversidade.

```
% Put the best individual in the first position, preserving the population's diversity
include_best(Best, Population, FinalPopulation) :-
    % Remove the worst individual from the population (preserving diversity)
    append(Front, [_Worst | Rest], Population),
    append(Front, Rest, TempPopulation),
    % Insert the best individual at the start
    append([Best], TempPopulation, FinalPopulation).
```

Figura 8 - Método *include_best(Best, Population, FinalPopulation)*.

- **Evitar Elitismo Excessivo:** Apenas o melhor indivíduo é preservado, garantindo que o resto da população continua a evoluir de maneira diversificada.

Conclusão

A **User Story 7.3.2** resultou na implementação de um sistema robusto de agendamento de operações em salas de cirurgia, utilizando algoritmos genéticos para otimização. A divisão do trabalho em quatro tarefas permitiu uma evolução estruturada do sistema, abordando desde a criação da base do algoritmo até à introdução de componentes avançados para flexibilidade e diversidade.

A estrutura básica implementada na **Tarefa 1** proporcionou um alicerce sólido, com mecanismos para geração, avaliação e evolução de uma população de soluções. As melhorias na parametrização (**Tarefa 2**) tornaram o algoritmo adaptável a diferentes cenários e exigências operacionais, aumentando a sua aplicabilidade.

A introdução de aleatoriedade no cruzamento (**Tarefa 3**) contribuiu significativamente para a diversidade genética da população, mitigando o risco de convergência prematura e promovendo a exploração de soluções alternativas. Por fim, a preservação do melhor indivíduo (**Tarefa 4**) assegurou que os progressos alcançados fossem mantidos, sem sacrificar a variabilidade necessária para a evolução contínua.

Em suma, a implementação deste módulo demonstra como algoritmos genéticos podem ser eficazes na resolução de problemas complexos de agendamento, combinando eficiência na execução com flexibilidade para atender diferentes condições e requisitos. O sistema desenvolvido representa uma abordagem avançada e promissora para a gestão de operações em salas de cirurgia.

Estado de Arte

A cirurgia ortopédica robótica constitui uma das inovações mais marcantes no domínio da medicina moderna, aliando a destreza dos cirurgiões à precisão das tecnologias de ponta. Estes sistemas utilizam tecnologias de imagem, como tomografias computadorizadas (TC) e ressonâncias magnéticas (RM), para criar mapas tridimensionais detalhados da anatomia do paciente. Estas representações permitem planejar e executar intervenções de forma mais precisa, personalizada e eficaz, contribuindo para a melhoria da segurança e eficiência nos procedimentos cirúrgicos. [1]

Entre os principais benefícios desta abordagem estão a elevada precisão, que reduz erros e potencia melhores resultados, e a menor invasividade dos procedimentos, traduzindo-se em tempos de recuperação mais curtos, menor dor no período pós-operatório e um internamento hospitalar reduzido, promovendo uma recuperação mais rápida. [1]

A aplicação da robótica na cirurgia ortopédica estende-se por diversas áreas clínicas. Em substituições articulares, os sistemas robóticos são utilizados para posicionar próteses com elevada precisão, especialmente em cirurgias ao joelho e à anca, melhorando a funcionalidade e a longevidade das articulações tratadas. Nas intervenções à coluna vertebral, a tecnologia assegura um alinhamento rigoroso de implantes e instrumentos, favorecendo resultados consistentes. Outras áreas em destaque incluem a cirurgia pediátrica e procedimentos oncológicos, onde os robôs são utilizados para reconstruir estruturas delicadas, reduzindo danos nos tecidos saudáveis circundantes. [1]

Com vista a aprofundar o conhecimento sobre o estado da arte no uso de robôs em contexto hospitalar, foi realizada uma revisão sistemática em quatro bases de dados reconhecidas – Elsevier, MDPI, IEEE e PubMed. A pesquisa centrou-se em três áreas principais de interesse:

- Cirurgias minimamente invasivas (MIS);
- Fases peri operatórias (pré, intra e pós-operatória);
- Treinamento cirúrgico.

Na área das cirurgias minimamente invasivas, a pesquisa resultou em 44 artigos na base MDPI, 205 na Elsevier, 48 no IEEE e 24 no PubMed. Após uma triagem cuidadosa, foram identificados 16 artigos de maior relevância. No âmbito das fases perioperatórias, foram encontrados 78 artigos no MDPI, 735 na Elsevier, 46 no PubMed e nenhum no IEEE, culminando na seleção de 15 estudos. Relativamente ao treino cirúrgico, foram identificados 177 artigos no MDPI, 338 na Elsevier e 183 no PubMed, sendo 14 considerados pertinentes para análise.

Para esta análise, foram selecionados nove artigos que representam diferentes perspetivas nos três domínios estudados.

Minimamente invasiva (MIS)

No campo das técnicas minimamente invasivas, destacam-se três estudos.

O primeiro analisa o uso de robôs na colocação de parafusos pediculares em cirurgias de fusão lombar. Os resultados demonstraram que esta tecnologia reduz a duração das cirurgias, a perda de sangue e o tempo de hospitalização quando comparada com métodos tradicionais, como fluoroscopia manual e navegação por tomografia, apresentando uma relação custo-benefício vantajosa. [2]

O segundo estudo investigou as complicações e taxas de revisão em fusões da coluna vertebral, comparando intervenções assistidas por robôs com técnicas baseadas em fluoroscopia. O grupo que utilizou os sistemas robóticos apresentou menores taxas de complicações e uma redução em 80% do tempo de exposição à radiação por fluoroscopia por parafuso para o paciente e a equipa cirúrgica. [3]

O terceiro artigo foca-se na aplicação de técnicas minimamente invasivas em crianças com fraturas traumáticas na região toracolumbar. Este estudo confirma a eficácia das técnicas robóticas, garantindo uma recuperação estável com menores complicações associadas ao uso de implantes. [4]

Peri operatório

Os artigos selecionados na área peri operatória analisam diferentes aspetos do impacto da robótica no desempenho cirúrgico e nos resultados para os pacientes.

O primeiro estudo avaliou os efeitos clínicos da utilização da robótica em fusões lombares, destacando melhorias significativas na estabilidade da coluna e na funcionalidade reportada pelos pacientes ao longo de um ano. Apesar de existirem complicações associadas, observou-se que estas tendem a diminuir à medida que os profissionais ganhavam mais experiência no uso da tecnologia em procedimentos posteriores. [5]

Um segundo estudo centrou-se no impacto da robótica em artroplastias totais bilaterais do joelho, evidenciando a capacidade desta tecnologia em reduzir a necessidade de transfusões de sangue após a segunda cirurgia, mesmo quando realizada num intervalo curto de tempo. Estes resultados foram atribuídos à precisão dos cortes ósseos, que preservaram melhor os tecidos circundantes, reduzindo o trauma durante o procedimento. [6]

O terceiro artigo comparou os resultados técnicos e clínicos em artroplastias unicongilares do joelho realizadas com e sem auxílio de robôs. Verificou-se que a robótica proporcionou maior precisão no alinhamento dos implantes, garantindo uma instalação mais consistente. Contudo, essa melhoria técnica não foi traduzida, pelo menos a curto prazo, em diferenças significativas no alívio da dor ou na funcionalidade clínica quando comparada às abordagens convencionais. [7]

Treinamento

Na área do treino, os estudos evidenciaram a importância do uso de robôs para aprimorar a experiência cirúrgica.

O primeiro estudo avaliou a aplicação de um sistema robótico para artroplastias totais do joelho utilizando cadáveres, permitindo medir a precisão dos cortes ósseos realizados. Os resultados evidenciaram que o sistema alcançou um elevado grau de exatidão, com desvios mínimos em relação ao plano cirúrgico traçado. Embora a pesquisa tenha sido conduzida em modelos não vivos, os dados sugerem um forte potencial de utilização em cenários clínicos reais. [8]

O segundo trabalho explorou a curva de aprendizagem associada a uma nova plataforma robótica para artroplastias da anca. Verificou-se que a eficiência e eficácia da técnica foram substancialmente melhoradas após 13 intervenções, com reduções significativas no tempo operatório e nas perdas sanguíneas. Este estudo sublinhou a importância do treino contínuo dos profissionais de saúde para maximizar os benefícios proporcionados pela adoção destas tecnologias inovadoras. [9]

Por último, um estudo experimental examinou a capacidade de um robô realizar a colheita do nervo frénico num modelo animal. Os resultados demonstraram uma precisão notável, reforçando as vantagens inerentes ao uso da robótica, nomeadamente maior controlo e fiabilidade. Estas características evidenciam o potencial desta tecnologia para ser aplicada em procedimentos clínicos exigentes, como reconstruções do plexo braquial. [10]

Os nove artigos selecionados ilustram, de forma clara, os progressos registados em cada uma das áreas chave da cirurgia ortopédica robótica. Sob diferentes prismas, estes estudos evidenciam o impacto desta tecnologia nas práticas cirúrgicas, desde o aumento da precisão técnica em intervenções minimamente invasivas, passando pela otimização do peri operatório, até ao desenvolvimento e capacitação de equipas médicas para operar sistemas robóticos avançados.

Conclusão

Este projeto abordou com sucesso a implementação de um sistema automatizado para a gestão de operações em salas de trabalho, utilizando algoritmos genéticos como núcleo metodológico. As soluções desenvolvidas demonstraram ser eficazes na resolução dos desafios apresentados, nomeadamente a atribuição automática de operações às salas, o agendamento eficiente de múltiplas tarefas e a parametrização de métodos de paragem para garantir a flexibilidade do sistema.

Um dos pontos de destaque foi a adaptação dos algoritmos genéticos ao contexto específico das salas de operações. A introdução de aleatoriedade no cruzamento entre indivíduos e a preservação dos melhores indivíduos sem recorrer ao elitismo permitiram um equilíbrio entre inovação e consistência nos resultados. Além disso, o foco na parametrização garantiu que o sistema fosse ajustável às necessidades específicas do utilizador, proporcionando soluções otimizadas dentro do tempo e dos recursos disponíveis.

A revisão do estado da arte foi essencial para enquadrar as escolhas metodológicas e validar os avanços alcançados. Os resultados obtidos refletem não apenas a viabilidade técnica das abordagens implementadas, mas também o seu potencial para futuras aplicações em contextos semelhantes.

Em suma, este trabalho contribui para a melhoria das práticas de gestão operacional em ambientes complexos, apresentando um sistema robusto e adaptável. No entanto, há espaço para expandir as soluções propostas, como a integração de novas técnicas de otimização ou a aplicação do sistema noutros domínios. Estes pontos abrem caminho para investigações futuras que poderão consolidar e ampliar os benefícios observados neste estudo.

Referências

- [1] ORTESP. (n.d.). *Cirurgia ortopédica robótica: Benefícios e aplicações em procedimentos cirúrgicos*. Disponível em: <https://www.ortesp.com.br/index.php/especialidades/ortopedia-geral/cirurgia-ortopedica-robotica-beneficios-e-aplicacoes-em-procedimentos-cirurgicos> (Acedido em: 03/01/2025).
- [2] Sturgill, D., How, J., Blajda, T., Davis, Z., Ali, M., O'Malley, G., Patel, N. V., Khan, M. F., & Goldstein, I. (2024). *Are the Clinical Outcomes and Cost-Effectiveness of Robot-Assisted Pedicle Screw Placement in Lumbar Fusion Surgery Superior to Computed Tomography Navigation and Freehand Fluoroscopy-Guided Techniques? A Systematic Review and Network Meta-Analysis*. *World Neurosurgery*, vol.191, pp. 81-90.
- [3] Good, C. R., Orosz, L., Schroerlucke, S. R., Cannestra, A., Lim, J. Y., Hsu, V. W., Zahrawi, F., Villalobos, H. J., Ramirez, P. M., Sweeney, T., & Wang, M. Y. (2021). *Complications and Revision Rates in Minimally Invasive Robotic-Guided Versus Fluoroscopic-Guided Spinal Fusions: The MIS ReFRESH Prospective Comparative Study*. *Spine*, vol. 46, nº 23, pp. 1661-1668.
- [4] Castillo, J., Soufi, K., Zhou, J., Kulubya, E., Javidan, Y., & Ebinu, J. O. (2024). *Minimally Invasive Techniques in the Surgical Management of Traumatic Pediatric Thoracolumbar Fractures*. *World Neurosurgery*, vol. 182, pp. 292-300.
- [5] Lee, N. J., Lombardi, J. M., Boddapati, V., Mathew, J., Leung, E., & Lehman, R. A. (2021). *Clinical and patient-reported outcomes after robot-assisted short-segment lumbar fusion with a minimum 1-year follow-up*. *Interdisciplinary Neurosurgery: Advanced Techniques and Case Management*, vol. 25.
- [6] Lee, J. H., Jung, H. J., Choi, B. S., Ro, D. H., & Kim, J. I. (2023). *Effectiveness of Robotic Arm-Assisted Total Knee Arthroplasty on Transfusion Rate in Staged Bilateral Surgery*. *Journal of Clinical Medicine*, vol. 12.
- [7] Wu, C., Fukui, N., Lin, Y.-K., Lee, C.-Y., Chou, S.-H., Huang, T.-J., Chen, J.-Y., & Wu, M.-H. (2022). *Comparison of Robotic and Conventional Unicompartmental Knee Arthroplasty Outcomes in Patients with Osteoarthritis: A Retrospective Cohort Study*. *Journal of Clinical Medicine*, vol. 11.
- [8] Yi, J., Gao, Z., Huang, Y., Liu, Y., Zhang, Y., & Chai, W. (2024). *Evaluating the accuracy of a new robotically assisted system in cadaveric total knee arthroplasty procedures*. *Journal of Orthopaedic Surgery and Research*, vol. 19, nº 354.
- [9] Sun, H., Lu, H., Xiao, Q., Ding, Z., Luo, Z., & Zhou, Z. (2024). *The learning curve of a novel seven-axis robot-assisted total hip arthroplasty system: a randomized controlled trial*. *BMC Musculoskeletal Disorders*, vol. 25, nº 342.
- [10] Porto de Melo, P., Miyamoto, H., Serradori, T., Mantovani, G. R., Selber, J., Facca, S., Xu, W.-D., Santelmo, N., & Liverneaux, P. (2014). *Robotic phrenic nerve harvest: A feasibility study in a pig model*. *Chirurgie de la main*, vol. 33, nº 4, pp. 356-360.