

# ALGAV

2019-2020

## RELATÓRIO SPRINT 3

Carlos Moreira – 1161882  
Marco Pinheiro – 1170483  
Pedro Barbosa – 1150486  
Pedro Mendes – 1161871

## Índice

Introdução.....	3
1º Expandir o problema para várias linhas de fabrico .....	4
2º Representar as máquinas através de agendas temporais .....	7
3º Sugestões de soluções para lidar com situações não previstas.....	8
4º Estudo da aplicação de uma técnica de Machine Learning a uma área do Planeamento da produção .....	9
Bibliografia .....	11
Conclusão .....	12

## Introdução

Este relatório surge no âmbito da disciplina de Algoritmia Avançada (ALGAV), do terceiro ano da Licenciatura em Engenharia Informática do Instituto Superior de Engenharia do Porto. Este sprint tem como objetivo a concretização de conhecimentos da disciplina, nomeadamente a expansão do problema para várias linhas de fabrico e representar as máquinas através de agendas temporais. Além disso, tem como objetivo a concretização de sugestões para resolver o problema de situações não previstas, bem como a realização de um estudo sobre o impacto atual de técnicas de machine learning no planeamento de produção.

## 1º Expandir o problema para várias linhas de fabrico

A resolução do problema de expansão para várias linhas de fabrico, tem como início a inclusão de novas linhas de produção, bem como máquinas que as constituem.

```
assert(machines_of_line("l1",["ma","mb","mc","md"])),
assert(machines_of_line("l2",["me","mf","mg","mh"])),
assert(machines_of_line("l3",["mi","mj","mk","ml"])),
assert(machines_of_line("l4",["mm","mn","mo","mp"])),
```

Figure 1 Atribuição de máquinas às linhas

```
assert(type_of_machine("ma","t1")),
assert(type_of_machine("mb","t2")),
assert(type_of_machine("mc","t3")),
assert(type_of_machine("md","t4")),
assert(type_of_machine("me","t2")),
assert(type_of_machine("mf","t6")),
assert(type_of_machine("mg","t8")),
assert(type_of_machine("mh","t7")),
assert(type_of_machine("mi","t5")),
assert(type_of_machine("mj","t6")),
assert(type_of_machine("mk","t2")),
assert(type_of_machine("ml","t3")),
assert(type_of_machine("mm","t1")),
assert(type_of_machine("mn","t2")),
assert(type_of_machine("mo","t6")),
assert(type_of_machine("mp","t8")),
```

Figure 3 Atribuição de tipos de máquinas a máquinas

```
assert(machine_type_operation("t1",p("opt1","fa"))),
assert(machine_type_operation("t1",p("opt2","fb"))),
assert(machine_type_operation("t1",p("opt5","fe"))),
assert(machine_type_operation("t2",p("opt1","fb"))),
assert(machine_type_operation("t2",p("opt2","fc"))),
assert(machine_type_operation("t2",p("opt6","ff"))),
assert(machine_type_operation("t3",p("opt2","fb"))),
assert(machine_type_operation("t3",p("opt2","fc"))),
assert(machine_type_operation("t3",p("opt5","fc"))),
assert(machine_type_operation("t4",p("opt3","fc"))),
assert(machine_type_operation("t4",p("opt6","fb"))),
assert(machine_type_operation("t5",p("opt1","fb"))),
assert(machine_type_operation("t5",p("opt4","fa"))),
assert(machine_type_operation("t5",p("opt7","fa"))),
assert(machine_type_operation("t6",p("opt3","fd"))),
assert(machine_type_operation("t6",p("opt1","fd"))),
assert(machine_type_operation("t6",p("opt5","fc"))),
assert(machine_type_operation("t7",p("opt1","fa"))),
assert(machine_type_operation("t7",p("opt3","fd"))),
assert(machine_type_operation("t7",p("opt8","fa"))),
assert(machine_type_operation("t8",p("opt1","fb"))),
assert(machine_type_operation("t8",p("opt3","fc"))),
assert(machine_type_operation("t8",p("opt5","fa"))),
```

Figure 2 Atribuição de operações aos tipos de máquina

O primeiro passo para a atribuição de tarefas a linhas de produção passa pela ordenação de todas as tarefas por tempo de conclusão.

```
sortTC(L,LR):-
    sortByTC(L, LR).
sortByTC(List,Sorted):-
    tcsort(List,[],Sorted),!.
tcsort([],Acc,Acc).
tcsort([H|T],Acc,Sorted):-
    tcinsert(H,Acc,NAcc),
    tcsort(T,NAcc,Sorted).
tcinsert(q(IX,IDX,MSX,TCX,PRX,PLX,TX),[q(IY,IDY,MSY,TCY,PRY,PLY,TY)|T],[q(IY,IDY,MSY,TCY,PRY,PLY,TY)|NT]):-
    TCX>TCY,
    tcinsert(q(IX,IDX,MSX,TCX,PRX,PLX,TX),T,NT).
tcinsert(q(IX,IDX,MSX,TCX,PRX,PLX,TX),[q(IY,IDY,MSY,TCY,PRY,PLY,TY)|T],[q(IX,IDX,MSX,TCX,PRX,PLX,TX),q(IY,IDY,MSY,TCY,PRY,PLY,TY)|T]):-
    TCX<=TCY.
tcinsert(X,[],[X]).
```

Figure 4 Ordenação de tarefas por tempo de conclusão

De seguida, é necessário filtrar quais as linhas onde é possível realizar determinado produto encomendado. Essa validação é conseguida através da verificação de quais operações são possíveis fazer em determinada linha de produção, quando comparados com o plano de produção daquele produto.

```
linhas_possiveis_tarefas([],_,[]).
linhas_possiveis_tarefas([H|T],LLP,R):-
    linhas_possiveis_tarefas(T,LLP,R2),
    linhas_possiveis_tarefa(H,LLP,R1),
    append([H*R1],R2,R).

linhas_possiveis_tarefa(_,[],[]).
linhas_possiveis_tarefa(TR,[H*_|T],R):-
    linhas_possiveis_tarefa(TR,T,R1),
    ((is_possible(TR,H),append([H],R1,R));(R = R1)).

is_possible(q(_,_,_,_),PL,_):-
    operations_of_line(L,L0),
    check_plan(PL,L0),!.

check_plan([],_).
check_plan([OP|T],[m(_),L0]|RT):-
    ((member(OP,L0),check_plan(T,[m(_),L0]|RT));(check_plan([OP|T],RT))).

operations_of_line(L,L0):-
    machines_of_line(L,LM),
    operations_of_machines(LM,L0).

operations_of_machines([],[]).
operations_of_machines([H|T],L0):-
    type_of_machine(H,MT),
    findall(p(OP,T00L),machine_type_operation(MT,p(OP,T00L)),L01),
    operations_of_machines(T,L02),
    append([m(H,L01)],L02,L0).

```

Figure 5 Filtragem das linhas onde é possível executar as tarefas

Após obter as linhas possíveis começa o processo de atribuição das tarefas às linhas. Essa atribuição é decidida através de uma heurística de balanceamento cuja responsabilidade é de calcular e comparar os makespan acumulados, assim como atribuir a tarefa à linha com o menor valor acumulado.

```
desliza_tarefas([_|[]],AG,AG).
desliza_tarefas([T1,T2|T],AGM,D):-
    tempo_deslizamento(AGM,T1,T2,TMP),!,
    sort(TMP,[RTMP|_]),
    desliza(AGM,T2,RTMP,AGM1),
    desliza_tarefas([T2|T],AGM1,D).

desliza([],_,_,[]).
desliza([M*A|T],TR,TMP,[M*A1|T1]):-
    desliza1(A,TR,TMP,A1),
    desliza(T,TR,TMP,T1).

desliza1([],_,_,[]).
desliza1([t(SI,SF,setup,T00L),t(EI,EF,exec,info(OP,P,Q,0,T))|TAIL],T,TMP,
    [t(SI1,SF1,setup,T00L),t(EI1,EF1,exec,info(OP,P,Q,0,T))|TAIL]):-
    SI1 is SI-TMP,
    SF1 is SF-TMP,
    EI1 is EI-TMP,
    EF1 is EF-TMP.
desliza1([t(EI,EF,exec,info(OP,P,Q,0,T))|TAIL],T,TMP,
    [t(EI1,EF1,exec,info(OP,P,Q,0,T))|TAIL]):-
    EI1 is EI-TMP,
    EF1 is EF-TMP.
desliza1([H|TAIL],T,TMP,[H|RT]):-
    desliza1(TAIL,T,TMP,RT).

```

Figure 6 Deslizamentos das tarefas

```

atribuir_tarefa_a_linha(LT,LLP):-
    sortTC(LT,LSorted),
    linhas_possiveis_tarefas(LSorted,LLP,R),
    atribui(LLP,R).

atribui(_,[]).
atribui(LLP,[q(_,_,_,_,_,_)*[]|T]):-
    atribui(LLP,T).
atribui(LLP,[q(I,ID,MS,TC,PR,_,TOOL)*[LPH|LPT]|T]):-
    length([LPH|LPT],N),
    (N==1,assert(tarefa(I,ID,MS,TC,PR,LPH,TOOL)),acumular_makespan(LLP,LPH,MS,NLLP,atribui(NLLP,T));
    (bsort(LLP,SLLP),heuristica_balanceamento(SLLP,q(I,ID,MS,TC,PR,_,TOOL)*[LPH|LPT],NLLP,atribui(NLLP,T))).

heuristica_balanceamento([H*A|T],q(I,ID,MS,TC,PR,_,TOOL)*L,[H*A1|RT]):-
    ((member(H,L),assert(tarefa(I,ID,MS,TC,PR,H,TOOL)),A1 is A+MS,RT=T);
    (A1=A,heuristica_balanceamento(T,q(I,ID,MS,TC,PR,_,TOOL)*L,RT))).

acumular_makespan([],_,_,[]).
acumular_makespan([H*Q|T],H,MS,[H*Q1|RT]):-
    Q1 is Q+MS,
    acumular_makespan(T,H,MS,RT).
acumular_makespan([H*Q|T],LEscolhida,MS,[H*Q|RT]):-
    acumular_makespan(T,LEscolhida,MS,RT).

```

Figure 7 Atribuição das tarefas às linhas de produção

## 2º Representar as máquinas através de agendas temporais

A resolução do problema da criação de agendas temporais para cada máquina inicializa com a aplicação do algoritmo genético, desenvolvido no sprint anterior, para cada uma das linhas de produção com tarefas. Para cada uma das tarefas é necessário ir buscar o plano de produção para aquele produto para posteriormente criar factos com o início e fim de cada operação, bem como a informação adicional. Após isso, agrupa todos os factos pertencentes a determinada máquina, para depois aplicar os deslizamentos.

```
agenda_temporal([H|T]*):-
    tarefa(H,_,_,_,LP,_),
    machines_of_line(LP,LM),
    loop_tarefas(LM,[H|T],AGS,0),!,
    merge_lists(AGS,ML),
    merge_agendas(LM,ML,AGM),!,
    desliza_tarefas([H|T],AGM,D),
    write(D),nl,nl.
```

Figure 8 Agenda temporal para cada máquina

Nos deslizamentos é subtraído o tempo final de uma operação numa máquina ao tempo inicial a operação seguinte e encontra o valor menor para depois aplicar um deslizamento igual a esse valor a todas as operações daquela tarefa, uma vez que esse valor é o número máximo de unidades de tempo que é possível deslizar.

[illegible]

Figure 9 Cálculo do menor valor de deslizamento

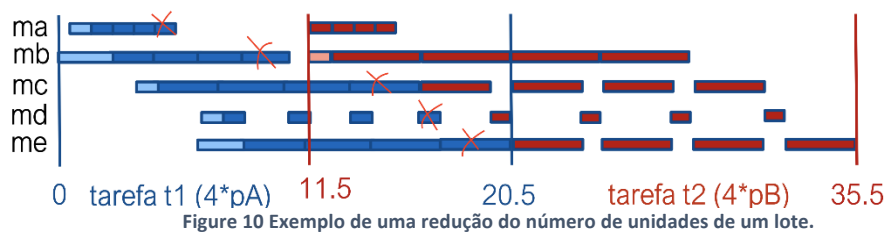
### 3º Sugestões de soluções para lidar com situações não previstas

Num contexto de uma empresa de manufatura, por vezes surgem situações não previstas no bom funcionamento da produção que têm que ser contornadas.

De seguida serão apresentadas algumas dessas situações bem como possíveis resoluções das mesmas.

No caso de avaria de uma máquina numa linha de produção: Uma vez que todo o planeamento é efetuado com a condição de uma determinada linha ter a capacidade de fazer (de acordo com o tipo de máquina das máquinas e respetivas operações) determinado produto, a melhor abordagem seria o replaneamento, pois a limitação da linha com a máquina avariada já seria contemplada.

No caso de redução do número de unidades de um lote: Uma abordagem possível, mas menos aconselhável seria o replaneamento. Porém, tem como inconveniente o tempo de geração do planeamento. Por sua vez, uma melhor abordagem seria reajustar o deslizamento para a esquerda, das tarefas daquele lote, bem como dos seguintes.



No caso de cancelamento de uma encomenda: Novamente, uma abordagem possível, mas menos aconselhável seria o replaneamento. Por sua vez, uma melhor abordagem seria reajustar o deslizamento para a esquerda das tarefas das encomendas seguintes.

No caso de uma encomenda extra de um cliente muito importante: Mais uma vez, uma abordagem possível, mas menos aconselhável seria o replaneamento. A melhor abordagem neste caso seria inserir as tarefas desta encomenda no início, efetuando o deslizamento para a direita das restantes tarefas das encomendas.



## 4º Estudo da aplicação de uma técnica de Machine Learning a uma área do Planeamento da produção

O planeamento de produção é um processo complexo com um conjunto vasto de variantes que têm que ser tidas em conta quando for feito o escalonamento das operações a realizar para satisfazer um conjunto de encomendas.

Essas variantes vão desde a operação que um tipo de máquina consegue fazer até a distribuição de todas as operações necessárias para a criação de um produto, numa determinada linha que contenha máquinas capazes para realizar tais operações.

Para além destas variantes, existem também situações inesperadas que poderão surgir, como a avaria de uma máquina ou a entrada de uma encomenda importante para o plano de produção.

Surge então o conceito de *Industry 4.0* que se trata de um novo paradigma na indústria de manufactura que visa adotar uma abordagem mais proactiva usando como apoio a constante evolução da interconectividade dos sistemas e também a recolha avançada de informação **(Mota et al., 2019)**.

A interconectividade dos sistemas permite a monitorização constante das máquinas que constituem as diversas linhas de produção conseguindo assim prever alguma avaria que possa surgir, mas também acompanhar o processo de produção obtendo assim, por exemplo métricas e previsões de conclusão de determinada encomenda. **(Ferreira et al., 2017)**.

A restrições e limitações de recursos, como as máquinas, são consideradas como o maior problema relativamente ao planeamento de produção (Abdolshah, 2014), exige-se por isso a monitorização das mesmas e recolha constante de informação através de sensores de laser e câmaras. **(Ferreira et al., 2017)**.

Após iniciar o processo de recolha dessa informação é possível prever problemas e iniciar processos de recuperação ou de reformulação de planeamento de produção.

No artigo “MASPI: A Multi Agent System for Prediction in Industry 4.0 Environment”

**(Candanedo, González, De la Prieta, & Arrieta, 2019)** é proposto um sistema multiagente que desenvolverá uma inteligência artificial modelo que será aplicada aos agentes, nomeadamente nas ações de processamento, testes, avaliação e previsão de informação e criação de relatórios para que a manutenção preditiva seja possível. Para isso são instalados diversos sensores nos equipamentos que medem a temperatura, pressão e vibração e através de conversores convertem o sinal analógico em sinal digital. São criados então diversos agentes de sistema (MAS) que têm responsabilidades

distintas, mas com uma relação estrutural que permite a interação entre os diversos agentes.

A um grupo destes agentes dão o nome de VO (Virtual Organizations) e todos eles têm que coordenar, supervisionar e sincronizar as tarefas que lhe estão alocadas, mas também tratar do pré processamento dos agentes. Esse pré-processamento inclui um conjunto de técnicas que podem ser agrupadas em 2 áreas principais: a preparação da informação e a redução da informação, ambas com o objetivo de servir de input para os algoritmos usados: *k-nearest neighbors* e *Naïve Bayes*.

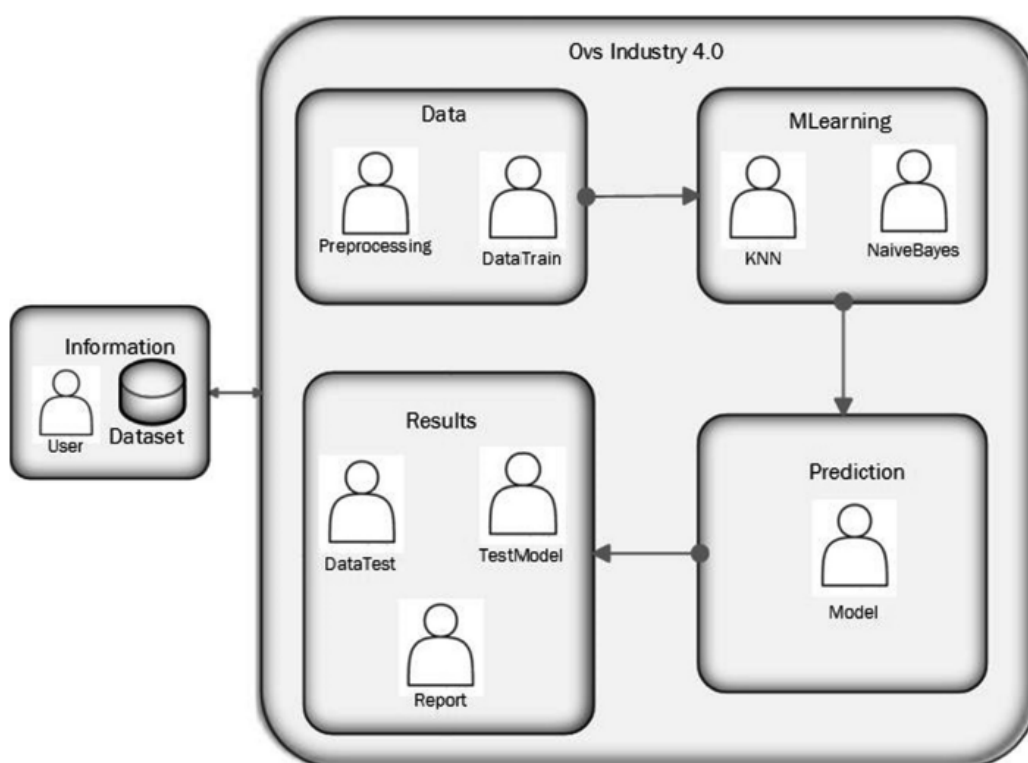


Figure 11 Retirado de artigo (Candanedo et al., 2019)

Os algoritmos *k-nearest neighbors* e *Naïve Bayes* são usados para obter a previsão modelo que será analisada pelo agente DataTest, depois pelo agente TestModel e finalmente pelo agente Report que criará, se for caso disso, informação pertinente e alertas.

## Bibliografia

- Abdolshah, M. (2014). A Review of Resource-Constrained Project Scheduling Problems (RCPSP) Approaches and Solutions. *Int. Trans. Journal of Engineering, Management, & Applied Sciences & Technologies*, 5(4), 253–286.
- Candanedo, I. S., González, S. R., De la Prieta, F., & Arrieta, A. G. (2019). MASPI: A Multi Agent System for Prediction in Industry 4.0 Environment. *Advances in Intelligent Systems and Computing*. [https://doi.org/10.1007/978-3-319-94120-2\\_19](https://doi.org/10.1007/978-3-319-94120-2_19)
- Ferreira, L. L., Albano, M., Silva, J., Martinho, D., Marreiros, G., Di Orio, G., ... Ferreira, H. (2017). A pilot for proactive maintenance in industry 4.0. *IEEE International Workshop on Factory Communication Systems - Proceedings, WFCS*. <https://doi.org/10.1109/WFCS.2017.7991952>
- Mota, D., Martins, C., Carneiro, J., Martinho, D., Conceição, L., Almeida, A., ... Marreiros, G. (2019). A MAS Architecture for a Project Scheduling Problem with Operation Dependant Setup Times. *Advances in Intelligent Systems and Computing*, 771, 177–186. [https://doi.org/10.1007/978-3-319-94120-2\\_17](https://doi.org/10.1007/978-3-319-94120-2_17)

## Conclusão

Este trabalho foi importante na consolidação de conhecimentos relativos ao Algoritmo Genético em que se fez uma simulação mais próxima da realidade da indústria manufatura, onde existem diversas máquinas em diversas linhas. Essa diversidade torna o problema de escalonamento ainda mais complexo, o que fez com que houvesse necessidade de perceber a que linha seria atribuído o lote de produtos a realizar. O somatório do *makespan* acumulado foi o fator de decisão e a linha que tivesse o menor ficaria com a atribuição do lote referido.

A fim de otimizar o escalonamento foram ainda criadas agendas, para cada uma das máquinas, que contêm todas as tarefas e respetivos períodos temporais que estas terão que realizar. Posteriormente foram efetuados deslizamentos de blocos de operações que reduzem o período de produção de uma encomenda significativamente.

Houve também a oportunidade de estudar diversas publicações e artigos científicos para a possibilidade de uma técnica de Machine Learning no planeamento de produção. Este estudo incidiu sobre a previsão de avaria das máquinas, uma vez que um dos maiores problemas da indústria manufatura são precisamente os fatores inesperados que acontecem com os recursos.