



# Relatório Sprint D de ALGAV

**Grupo 3**

José Mota (1161263)

Pedro Real (1170689)

João Flores (1171409)

Patrick Timas (1171352)

## Sumário

Índice .....	<b>Erro! Indicador não definido.</b>
Introdução.....	2
Expansão do problema a várias linhas de fabrico .....	2
Balanceamento de tarefas pelas linhas.....	3
Sequenciamento de lotes de produtos em cada linha.....	3
Representação das máquinas através de agendas temporais .....	4
Situações não previstas.....	6

## Introdução

Para este trabalho os objetivos a alcançarmos era garantir a existência de múltiplas linhas de produção e representar as linhas segundo agendas temporais. Para além disso, foi-nos solicitado realizar um estudo/pesquisa relativamente a sugestões de como se deveria proceder para a partir da solução gerada para lidar com situações não previstas, aplicação de pelo menos uma técnica de Machine Learning a pelo menos uma área do Planeamento da Produção

## Expansão do problema a várias linhas de fabrico

A primeira tarefa deste sprint foi garantir a existência de funcionamento de múltiplas de produção. Para tal, uma das soluções que encontramos foi definirmos uma estrutura contendo o id da linha, o MakeSpan acumulado da linha e uma lista de tarefas, sendo que inicialmente o valor do MakeSpan é zero e a lista está vazia.

```
cria_e_preenche_Linhas(ListaTarefasOrdenadas, LinhasPreenchidas) :-  
    findall(Linhas, linhas(Linhas), Lista),  
    flatten(Lista, LinhasExistentes),  
    write(LinhasExistentes),  
    iniciaLinhas(LinhasExistentes, ListaLinhas),  
    write(ListaLinhas),  
    atribui_tarefa_linha(ListaTarefasOrdenadas, ListaLinhas, LinhasPreenchidas).  
  
iniciaLinhas([], []).  
iniciaLinhas([L|Linhas], [ 1(L, 0, []) | TarefasLinha ]):-  
    iniciaLinhas(Linhas, TarefasLinha).
```

## Balanceamento de tarefas pelas linhas

Para a realização desta tarefa temos que atribuir a cada tarefa criada anteriormente, e ordenadas pelo seu tempo de conclusão, à linha que produza o produto contido na tarefa e que o produza com o menor makespan.

```
atribui_tarefa_linha([],TarefasLinhas,TarefasLinhas).
atribui_tarefa_linha([Tarefa|Tarefas],TarefasLinha,TarefasLinhaFinal):-
    sort(2, @=<,TarefasLinha,TarefasLinhaOrd),
    get_menor_linha(Tarefa,TarefasLinhaOrd,MenorLinha),
    delete(TarefasLinha,MenorLinha,Restantes),
    MenorLinha = l(Linha,MakespanAcomulado,T),
    Tarefa=tarefa(_,Makespan,_,_),
    MK1 is MakespanAcomulado + Makespan,
    atribui_tarefa_linha(Tarefas,[l(Linha,MK1,[Tarefa|T])|Restantes],TarefasLinhaFinal).

get_menor_linha(_,[],erro).
get_menor_linha(Tarefa,[TarefaLinha|TarefasLinha],MenorLinha):-
    Tarefa=tarefa(IdTarefa,_,_,_),
    TarefaLinha=l(Linha,_,_),
    obterOperacoesdaLinhaSemRepetidos(Linha,OperacoesLinhaSemRepetidos),
    dados_produto_cliente_da_tarefa(IdTarefa,Produto,_,_,_),
    operacoes_produto(Produto,OperacoesProduto),!,
    ((allMembers(OperacoesProduto,OperacoesLinhaSemRepetidos),!,MenorLinha=TarefaLinha);get_menor_linha(Tarefa,TarefasLinha,MenorLinha)).

obterOperacoesdaLinhaSemRepetidos(Linha,OperacoesLinhaSemRepetidos):-
    tipos_maq_linha(Linha,TiposMaquinaLinha),
    findall(Operacao,(member(Maquina,TiposMaquinaLinha), operacao_maquina(Operacao,Maquina,_,_,_)),OperacoesLinha),
    sort(0, @=<,OperacoesLinha,OperacoesLinhaSemRepetidos).
```

## Sequenciamento de lotes de produtos em cada linha

Para o sequenciamento das tarefas em cada linha, utilizou-se o um gerador das melhores soluções, utilizando “Brute Force” visto que o nosso algoritmo genético realizado no sprint passado não estava correto e tivemos vários problemas, que nos impossibilitou de o utilizar nesta iteração. O gerador de todas as soluções é chamado para todas as linhas. A lista populações será o resultado do gerador das melhores soluções.

```
brute_force([],[]).
brute_force([TarefaLinha|TarefasLinha],[l(Linha,Pop)|Populacoes]):-
    brute_force(TarefasLinha,Populacoes),
    (retract(tarefas(_));true), (retractall(tarefa(_,_,_,_));true),
    TarefaLinha=l(Linha,_,Tarefas),
    length(Tarefas, NumTarefas),
    asserta(tarefas(NumTarefas)),
    assert_tarefas(Tarefas),
    geraListasdeTarefas(Pop,X).
```

## Representação das máquinas através de agendas temporais

Como dito anteriormente, as máquinas serão representadas por uma agenda temporal, agenda esta constituída pela linha de produção, tipo de máquina, e a sequencia temporal que contem setups e execução, sendo que os setups conterão o início e o fim da operação, a referência a Setup e a ferramenta. Em contrapartida, a execução ira conter o início e o fim da operação, uma referência a exec e uma estrutura de dados adicionais contendo a operação, a quantidade, o produto, o cliente e a tarefa.

Inicialmente começamos por apagar os factos existentes das agendas (caso existam), e criamos e guardamos as agendas temporais para máquina presente na linha. Assim a agenda fica com a linha, a respetiva máquina e uma lista vazia que irá conter a sequência de setups e execução.

```
cria_pre_agendas([Pop|Populacoes],PreAgendas,PreAgendasFinal):-
    Pop = l(Linha,Genes),
    asserta(linha_escalonamento(Linha,Genes)),
    cria_pre_agenda(Linha,Genes,PreAgenda),
    cria_pre_agendas(Populacoes,[PreAgenda|PreAgendas],PreAgendasFinal).

cria_pre_agenda(_,[],[]).
cria_pre_agenda(Linha,[Tarefa|Tarefas],[Lista|PreAgenda]):-
    cria_pre_agenda(Linha,Tarefas,PreAgenda),
    tipos_maq_linha(Linha,MaquinasLinha),
    obter_Dados_Relativos_Linha_Tarefa_Produto_Cliente_Tempos(Linha,Tarefa,MaquinasLinha,Lista).

obter_Dados_Relativos_Linha_Tarefa_Produto_Cliente_Tempos(Linha,Tarefa,MaquinasLinha,Lista):-
    dados_produto_cliente_da_tarefa(Tarefa,Produto,Client,Qt,TConc),
    findall( l(Linha,Tarefa,Op,M,F,Produto,Client,Qt,TConc,Tsetup,Texec),
        (op_prod_client(Op,_,_,Produto,Client,Qt,TConc,_,_),
         classif_operacoes(Op,Opt),
         operacao_maq(M,Op,M,F,Tsetup,Texec),
         member(M, MaquinasLinha)
        ),Lista).
```

Depois desta inicialização, temos que associar as operações da linha com a tarefa. Aqui entra as tarefas originadas anteriormente associadas a linha de produção em questão e são armazenadas em *linha\_escalonamento/2*, contendo a linha e a lista. Depois associamos a linha á tarefa e as operações que serão realizadas naquela tarefa. Por fim, obtemos também a máquina, tempo de setup e tempo de execução para a linha em questão, através do tipo de operação, realizando um findall.

Devido a erros durante o nosso trabalho não conseguimos terminar a otimização das linhas, não sendo possível realizar os deslizamentos para a esquerda das operações. Apenas nos foi possível realizar a remoção dos setups desnecessários, setups estes que correspondem ao Setup de ferramentas iguais a anterior a sua execução.

Para cada maquina será retirado o Setup desnecessário. Assim vamos a agenda e vamos verificar se a ferramenta do Setup está em utilização, se esta estiver,não será adicionado este Setup a agenda,caso contrario será adicionado este Setup a agenda.

```

agendas(InstanteInicial,[Tarefa|Tarefas],Anteriores,Instante,TExecAnterior,ExecFimAnterior,InstanteFinal):-
    Tarefa = 1(Linha1,Tarefa1,Opl1,Maquina1,Ferramental,Produto1,Clientel,Qtl,TConcl,TSetup1,TExec1),
    ( ( Instante==0,!,SetupVirtual is InstanteInicial);( SetupVirtual is Instante - TSetup1)),

    ( ( SetupVirtual<InstanteInicial,!,Deslizamento is InstanteInicial-SetupVirtual
        |,SetupInicio is InstanteInicial)
      ; ( SetupInicio is SetupVirtual ,Deslizamento is 0 )),

    SetupFim is SetupInicio + TSetup1,
    ExecInicio is SetupFim,
    ((( TExecAnterior<TExec1,!,ExecFim is Qtl * TExec1 + ExecInicio,Cadencia is TExec1))

        ;(ExecFim is ExecFimAnterior + TExec1 ,Cadencia is TExecAnterior + TExec1 )),

    % desliza(Anteriores,Deslizamento, InstanteInicial),% nao funciona a 100% pelo que nao Ã© utilizado, apenas conseguimos remover os setups.

    Setup = t(SetupInicio,SetupFim,setup,Ferramental),
    Detalhes = dadosExtraTarefa(Opl,Qtl,Produto1,Clientel,Tarefa1),
    Execucao=t(ExecInicio, ExecFim,exec,Detalhes),
    agenda_maq(Linha1,Maquina1,Sequencia),
    ((retract(agenda_maq(Linha1,Maquina1,_)));true),
    reverse(Sequencia,SequenciaReversed),

    Sequencia1=[Setup|SequenciaReversed],
    Sequencia2=[Execucao|Sequencia1],
    reverse(Sequencia2,SequenciaAtualizada),
    asserta(agenda_maq(Linha1,Maquina1,SequenciaAtualizada)),
    InstanteSeguinte is ExecInicio + TExec1,
    agendas(InstanteInicial,Tarefas,[Tarefa|Anteriores],InstanteSeguinte,Cadencia,ExecFim,InstanteFinal).

```

Tal como era explicado nas tps, verificamos se o instante é zero, isto é correspondente a primeira operação, se sim o valor da variavel auxiliar é zero, caso contrario o instante corresponde ao instante menos o setup.

Somando o inicio da duração com a duração do setup obtemos o calculo total.

De seguida são calculados os valores e utilizados para criar as estruturas, inclusive as agendas, estas agendas são adicionadas finalmente a base de conhecimento, na qual a linha tem as maquinas com as respetivas agendas.

## Situações não previstas

Nas empresas podem acontecer situações que não estavam previstas, convém prever algumas situações. Quando essas situações acontecem devemos prever e rever o plano previsto, podem depender de várias variáveis tais como prioridades, número de encomendas, máquina disponíveis.

### Avaria Máquina

Caso seja uma avaria poderá ser repara em pouco tempo, ou em muito colocando a linha de produção parada até a sua reparação.

Soluções possíveis:

- Alterar as linhas onde estava inicialmente designada.
- Introduzir a encomenda no fim de outra.
- Fazer novamente o planeamento todo.

### Encomenda de cliente com prioridade máxima

Com a entrada da nova encomenda, dada a prioridade do cliente é necessário despassar a encomenda o mais rápido possível. Conforme o estado da produção, estando ativa ou não.

- Caso não estava em produção deve ser replaneado o plano.
- Caso esteja devem ser calculados os novos atrasos e introduzir ao plano.

### Cancelamento de encomenda/s

Com a remoção da encomenda deve acontecer:

- Remover a encomenda atualizando os tempos.

### Má produção

Caso seja produzida uma encomenda com defeito devemos:

- Caso todas as linhas estejam ocupadas devemos introduzir na que ficar com menor atraso após cálculo com a repetição da encomenda.
- Caso haja linhas disponíveis devem ser introduzidas.

## Machine Learning

Machine learning é uma aplicação de inteligência artificial / estudo científico de algoritmos e modelos estatísticos que os sistemas computacionais usam para que uma determinada tarefa em específico seja realizada a partir de padrões e inferência. Utiliza observações para criar padrões nesses mesmos dados e tomar as melhores decisões no futuro.

Neste estudo, dos vários métodos de Machine Learning, escolheu-se estudar o método de Artificial Neural Network. As redes neurais artificiais são modelos inspirados na estrutura das redes neurais biológicas. Designam-se por uma classe de correspondência de padrões que são geralmente usados em problemas de classificação e regressão, mas são um grande subcampo composto por centenas de algoritmos e variações para variados tipos de problemas

O objetivo inicial da Artificial Neural Network (ANN) era resolver problemas da mesma forma que um cérebro humano resolveria. Ao longo tempo, o seu objetivo mudou, sendo que passou a ser a execução de tarefas específicas, com desvios da biologia. As ANN foram usadas em variadas tarefas, tais como: tradução automática, filtragem de redes sociais, diagnósticos médicos, visão computacional e em atividades que são consideradas apenas factíveis pelo ser humano (casos como a pintura).

No primeiro artigo escolhido, o algoritmo da artificial Neural Network é utilizado na criação de uma “Rede Neural Feedforward” baseada num sistema inteligente para planeamento de processo assistido por programas computacionais.

Neste artigo, apresenta-se uma solução baseada em CAPP (Computer Aided Process Planning) e Neural Networks em que ajudam a escolher a sequência de operações de fabrico baseados nos atributos de recurso do componente.

No caso de estudo utilizado pelo artigo, os dados da Neural Networks são desenvolvidos tendo em conta os princípios de design para fabricação, para cada operação. Baseado em valores restritos (inputs e outputs) dos atributos do componente, o sistema de Neural Networks consegue identificar a sequência de operações necessárias para produzir o componente. O modelo de redes neurais deve ser treinado para diferenciar as várias situações.

No caso de estudo, escolheu-se um objeto formado por 3 componentes, em que os inputs e outputs do modelo de Neural Networks está representado abaixo:

Inputs	Outputs
E1/E2 : 0.300 0.160 0.187 0.330 0.0000010	1 1 0 0 0 0 0 0 0 0 (S1, S2,...)
E3/E5 : 0.120 0.220 0.100 0.340 0.200 0.340 010	0 0 1 0 0 0 1 0 0 0 (S3,...S7,...)
E4/E5 : 0.120 0.220 0.100 0.340 0.200 0.340 010	0 0 1 0 0 0 1 0 0 0 (S3,...S7,...)
E5/E7 : 0.320 0.800 0.340 0.130 0.106 0.615 001	0 0 0 0 1 1 0 0 0 0 (...S5,S6,...)
E7/E8 : 0.640 0.100 0.300 0.800 0.375 001	0 1 0 0 1 1 0 0 0 0 (...S2,...S5,S6,...)

Figura 1 Inputs e Outputs do modelo Neural Network

As conclusões retiradas deste mesmo estudo são que este novo método de planeamento de processos traz vantagens. A sequência gerada é quase ideal quando se minimiza o número de configurações se e minimiza o número de trocas de ferramentas para

realizar operações. O desenvolvimento dos sistemas CAPP vão ser bastante importantes para especialistas a trabalhar em projetos relacionados a este tema. O uso de Artificial Neural Networks foi bastante vantajoso, pois melhorou o uso da metodologia CAPP que consegue ser facilmente usada para novos componentes. Por último, esta metodologia reduziu o tempo necessário para gerar planos de fabricos e com resultados consistentes e de qualidade.

<https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=609683>

No segundo artigo, o artigo tenta arranjar uma solução para o agendamento de tarefas e seu sequenciamento de produção industrial. O artigo propõe como uma solução para esse mesmo problema uma abordagem com técnica na rede neural.

O objetivo principal deste estudo é estudar o agendamento de tarefas repetidas num sistema de plano de produção (SPP). O SPP consiste em várias linhas de produção com várias máquinas com transportadores. Uma máquina consegue fazer várias tarefas, mas apenas uma tarefa consegue ser executada de cada vez. As tarefas estão ciclicamente agendadas de acordo com os precedentes condições e disponibilidade das máquinas, como mostra a figura abaixo.

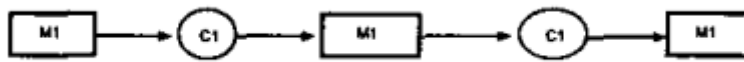


Figura 2 Linha de produção com 3 máquinas e 2 transportadoras

O modelo que o artigo pretende implementar para resolver este problema consiste em dois componentes principais: a componente de avaliação de qualidade e a componente de agendamento. O componente de avaliação da qualidade determina a qualidade das soluções de acordo com o interesse (ex: tempo de ciclo), a taxa de transferência e a média do inventário do WIP(Workin-Progress). Em relação ao componente do planeamento é sugerir novas sequências de tarefas para ser avaliado pelo componente de avaliação da qualidade até que o resultado obtido for satisfatório.

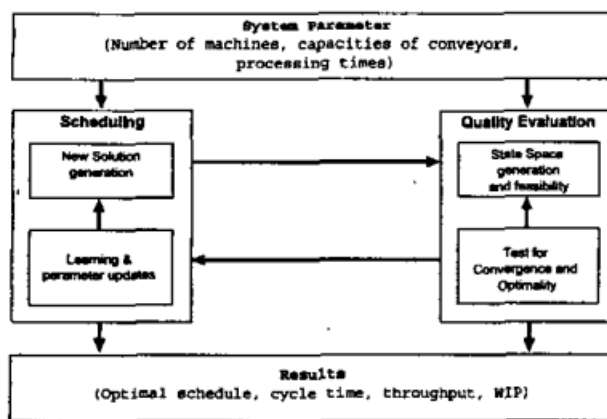


Figura 3 Agendamento e qualidade componentes

Os resultados obtidos considerando um caso com 4 operações e 2 máquinas apresenta o resultado obtido na seguinte figura.



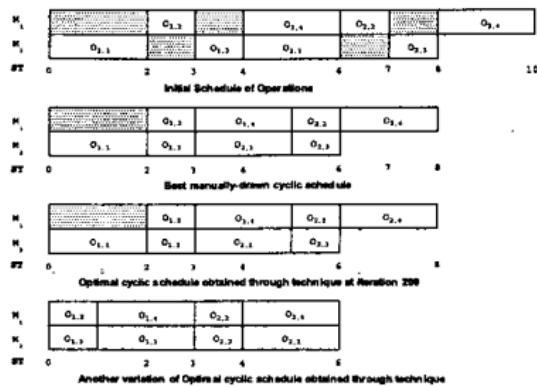


Figura 4 Comparação do agendamento obtido manualmente e usando a técnica

<https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1176333>

A figura acima representa a comparação obtida do tempo manualmente e através da técnica apresentada. A técnica foi capaz de dar a melhor opção de sequenciamento das tarefas com o menor tempo possível. Como demonstrado na figura, tanto manualmente como a partir da técnica foi possível apresentar o melhor sequenciamento das tarefas.

No 3º artigo, pretende-se encontrar soluções para o planeamento da instalação e sequenciamento de operações. Este problema de sequenciamento é feito através de Neural Network e convertido no problema do caixeiro viajante em que o seu objetivo é reduzir o custo total, usando, neste caso, um algoritmo genético.

Para tentar encontrar soluções para o problema apresentado, dividiu-se o método em vários passos: primeiro começa-se por identificar as formas geométricas para serem fabricadas com detalhes tecnológicos; Uma tabela é feita em que demonstra a abordagem das ferramentas e as instruções e ferramentas necessárias para os recursos, sendo que as instruções e ferramentas serão valores para padrões de ferramentas futuras; De seguida, o agrupamento de recursos é feito de acordo com a direção da abordagem da ferramenta e ferramentas em comum; Fazer os sequenciamos dos setups das operações em cada setup; Relação de precedência das operações das máquinas é obtida considerando as várias restrições de viabilidade; Por último, determinar a melhor sequência de operações no ambiente pretendido utilizando os algoritmos genéticos.

Para apresentar resultados, dividiu-se as operações em dois grupos baseados na sua relação.

1	F1,F2,F3,F4,F12	T1,T2,T4,T5 T7,T8
2	F5,F6,F7,F8,F9 F10,F11	T1,T2,T3,T4 T5,T6

Figura 5 Outputs finais depois de agrupamento

Após se dividir as operações em grupos, a ferramenta necessária para cada grupo é convertida numa operação porque cada ferramenta representa uma única operação. A cada

operação foi atribuído um número baseado na sua precedente relação. Este processo é iterativo até que se encontre uma solução ótima. A tabela abaixo demonstra os códigos atribuídos a cada operação no primeiro grupo e para o grupo 2. Estas tabelas indicam a sequência de operações ótimas para que o “custo” seja o menor.

Operation	Code	Operation	Code
Plane milling (A)	0	Plane milling (A)	0
Rough milling (B)	1	Rough milling (B)	1
Plane milling (C)	10	Slot cutting (C)	4
Slot cutting (D)	4	Hole (D)	5
Slot cutting (E)	14	Rough boring (E)	6
Finish milling (F)	2	Plane milling (F)	10

*Figura 6 Códigos das operações para os dois grupos*

Este artigo apresenta resultados bastante eficientes, pois, por exemplo, o algoritmo utilizado pode ser estendido e manipulado, a criação dos setups não tem apenas como base o básico da ferramenta, mas também a sua informação, materiais e outros fatores, etc...

<https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4492512>

#### Referências:

- [https://en.wikipedia.org/wiki/Artificial\\_neural\\_network](https://en.wikipedia.org/wiki/Artificial_neural_network)
- [https://pt.wikipedia.org/wiki/Aprendizado\\_de\\_m%C3%A1quina](https://pt.wikipedia.org/wiki/Aprendizado_de_m%C3%A1quina)
- <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1176443>
- <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1176333>
- <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4492512>